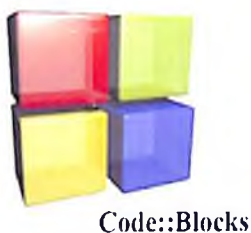




**B.B.Mo'minov**

# DASTURLASH II



**O‘ZBEKISTON RESPUBLIKASI**  
**OLIY VA O‘RTA MAXSUS TA‘LIM VAZIRLIGI**  
**MUHAMMAD AL-XORAZMIY NOMIDAGI TOSHKENT**  
**AXBOROT TEXNOLOGIYALARI UNIVERSITETI**

**B.B.MO‘MINOV**

# **DASTURLASH II**

*O‘zbekiston Respublikasi Oliy va o‘rta maxsus ta‘lim vazirligi tomonidan oliy  
o‘quv yurtlarining talabalari uchun darslik sifatida tavsiya etilgan  
(28.12.2020 yil 676 sonli buyrug‘i)*

TOSHKENT 2020

UO‘K: 004.4(075)  
KBK 32.973.-01

Mo‘minov B.B., Dasturlash II. (Darslik). – T.: «Nihol print»  
OK, 2021. – 604 b.

ISBN 978–9943–7028–1–3

Darslikda noma‘lum tiplar va noma‘lum nomlar fazosi, STL kutubxonalari, oddiy, assosiativ va tartiblanmagan assosiativ konteynerlar hamda konteynerlar adapterlari tushunchalari, standart algoritmlar, iteratorlar va sonli sinflar bilan ishlash, sintaktik tahlil qilishni tashkil etishning samarali usullari va murakkab saralash algoritmlaridan unumli foydalanish, Visual C++ muhitida menyular va uskunalar paneli, komponenta tushunchasi va xususiyatlari, muloqot oynalari, Visual C++ ning grafik imkoniyatlari, OLE va MFC texnologiyalari hamda kichik loyihalarni yaratishning nazariy va amaliy asoslari keltirilgan.

UO‘K: 004.4(075)  
KBK 32.973.-01

*Taqrizchilar:*

- Nuraliyev F.M.** – Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti, Televizion texnologiyalari fakulteti dekani, t.f.d., dotsent;
- Raximov N.O.** – O‘zbekiston Milliy universiteti Jizzax filiali, o‘quv ishlari bo‘yicha direktor o‘rinbosari, t.f.d.

ISBN 987–9943– 7028–1–3

© B.B.Mo‘minov, 2021.  
© «Nihol print» OK, 2021.

## KIRISH

O‘zbekiston Respublikasi Prezidentining bir qator Qaror va Farmonlari Axborot-kommunikatsiya texnologiyalarini yanada jadal ruvojlantirish maqsadlarida qabul qilinmoqda. Ushbu qonunlardan “Raqamli iqtisodiyot va elektron hukumatni keng joriy etish chora-tadbirlari to‘g‘risi”dagi PQ-4699-son 28.04.2020 va “«Raqamli O‘zbekiston — 2030» strategiyasini tasdiqlash va uni samarali amalga oshirish chora-tadbirlari to‘g‘risi”dagi PF-6079-son 05.10.2020 yilgi qaror va farmonlarida takidlanganidek «Bir million dasturchi» loyihasi doirasida 500 ming nafar yoshlarni qamrab olish orqali kompyuter dasturlash asoslariga o‘qitish tashkillashtirish maqsadida Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universitetining barcha yo‘nalish talabalariga mo‘ljallangan “Dasturlash II” fani uchun darslikni yaratish dolzarb masala hisoblanadi.

Darslikning birinchi bobida noma’lum tiplar va noma’lum nomlar fazosi, STL kutubxonalari, oddiy konteynerlar, assosiativ va tartiblanmagan assosiativ konteynerlar hamda konteynerlar adapterlari tushunchlari va ulardan dasturlashda unumli foydalanish o‘rgatiladi.

Darslikning ikkinchi bobida standart algoritmlar, iteratorlar va sonli sinflar bilan ishlash hamda sintaktik tahlil qilishni tashkil etishning samarali usullari va murakkab saralash algoritmlaridan unumli foydalanish o‘rgatiladi.

Darslikning uchinchi bobida Visual C++ muhitida menyular va uskunalar paneli, komponenta tushunchasi va xususiyatlari, muloqot oynalari, Visual C++ning grafik imkoniyatlari, OLE va MFC texnologiyalari hamda kichik loyihalarni yaratish o‘rgatiladi.

Qo‘llanma bo‘yicha takliflar, undagi aniqlangan kamchiliklar bo‘yicha fikr-mulohazalaringizni mbbahodir@gmail.com elektron pochtasiga yuborishingizni so‘raymiz. Sizning bu bildirgan fikr-mulohazalaringiz qo‘llanmaning keyingi nashrlarida uni yanada mazmunliroq va kamchiliklardan xoli tarzda chop etishimizga albatta asqotadi.

**Muallif**

## 1-BOB. KONTEYNERLAR

### 1.1. Noma'lum tiplarni yaratish va nomlar fazosi.

📖 Tiplarni dinamik tarzda aniqlashning ba'zi usullari, typeid operatori, typeid kutubxonasi, typeid sinfi'zolaridan foydalanish, dastur bajarilish davomida ko'rsatkichli va o'zgaruvchilarning tiplarini almashtrish operatorlari dynamic\_cast, const\_cast, static\_cast, reinterpret\_cast operatorlaridan foydalanish, noma'lum va foydalanuvchining nomlar fazolarini yaratish va dasturda ularda foydalanish usullari, buferlashtirilgan kirish va chiqishni yaratish, IO amallarida formatlashning identifikatorlari va monupulyatorlarini o'rnatish va o'chirish usullari keltirilgan bo'lib, nazariy bilimlarni asoslash uchun dasturlar tuzib ko'rsatilgan.

Bilimlarni mustahkamlash uchun 25  
tanazariy savol va amaliy ko'nikma va malakalarni rivojlantirish uchun 4  
ta assistent topshiriq va bahar assistentda 7 ta topshiriq, jami  
28 ta topshiriq berilgan.

Butopshiriqlarni bajarish mavzu mustahkamlash uchun xizmat qiladi.

*✍️ Kalitso'zlar.* Tipni aniqlash, tipni almashtrish, nomlar fazosi, buferlashtirilgan kirish va chiqish, typeid, typeid sinfi, dynamic\_cast, const\_cast, static\_cast, reinterpret\_cast, namespace, ifstream, oqim, formatlash identifikatori, monupulyatorli formatlash.

*📝 Bilish shart bo'lgan tushunchalar.* Tip tushunchasi, sinf va sinf ob'ekti, polimorf sinf, kasting amali, dastur fragmenti va soxasi, kutubxona, sinfi'zolari, funksiyalar va ko'rsatkich, identifikator, noma'lum otlarni kirish va chiqishi, oqim, dasturlashga oid dastlabki tushunchalar va C++ tilini o'qitish uchun qimmatli va zarur bo'lgan hitda ishlatishni bilish lozim.

#### *📖 Bilib olasiz.*

Ixtiyoriy ko'rsatkichli o'zgaruvchining tipni aniqlash va tipni almashtrish usullari, typeid, typeid sinfi, dynamic\_cast, const\_cast, static\_cast, reinterpret\_cast yangi operatorlar, kalitso'z namespace va nomlar fazosini yaratish, ulardan dasturda foydalanish usullari va texnologiyalari, buferlashtirilgan kirish va chiqish orqali noma'lum otlarni massivga joylatirish va a ifstream, oqim, formatlash identifikatori, monupulyatorli formatlash usullari orqali lokalizatsiyalashgan IO operatorlarini yaratish va foydalanish usullari o'rganishning mumkin.

## REJA

1. Tiplarnidynamiktarzdaaniqlash.
2. Tiplarnialmashtirishoperatorlari.
3. Yanginomlarfazosiyaratish.
4. Buferlashtirilgankiritishvachiqarish.

## KIRISH

Noma'lum tiplar vanoma'lum fazolar uchun dinamik tipni aniqlash va tiplarni almashtirish operatorlarizamonaviy ob'ektga yo'naltirilgan dasturlashni iqlab-quvvatlovchi C++ dasturlash tilining kiritilgan vositalari: dinamik tipni aniqlash (runtime type identification - RTTI) va qo'shimchatip quyish operatorlar bilan bog'liq. Bu vositalarning biri C++ spesifikatsiyasi qismi emasdi, lekin har ikki runtime polimorfizm uchun qo'llab-quvvatlash oshirish uchun qo'shilgan. RTTI dastur bajarish davomida ob'ekt turini aniqlash imkoniyati mavjudligini bildiradi.

Bundamuhokama qilina yotgan tiptur itashlash operatorlari bu amalni bajarish uchun dasturchigaxavfsiz yo'llarini taklif qiladi. Ulardan biri, `dynamic_cast`, bevosita RTTI aniqlash bilan bog'liq.

***Tiplarnidynamiktarzdaaniqlash.*** Ushbu vosita polimorf bo'lmagan til larda (masalan, C) mavjud emasligi sababli dinamik turdagi identifikatsiyalash bilan tanish bo'lmaganligi mumkin, chunki har bir ob'ektning tipi kompilyatsiya vaqtida (ya'ni, dastur bajarishda) ma'lum bo'lgani uchun dastur bajarilishida ma'lum otlashning hojati yo'q.

Biroq, C++ kabipolimorf tillarda ob'ekt tipi kompilyatsiya vaqtida ma'lum bo'lgan holatlar bo'lishi mumkin,

chunki dastur bajarilaboshlagunga qadar bu ob'ektning aniqlik tipi (tiplari) aniqlanmaydi. Ma'lumki, C++ sinfi erarxiyasi, virtual funksiyalar va ko'rsatkichlarni (pointer)

asossin flarga ishlatib polimorfizmni amalga oshiradi.

Tayanchsinfga ko'rsatkich butayanchsin fa'zolariga yo'ki undan olingan harq anday ob'ekta'zolariga murojaat qilish uchun foydalanish mumkin.

Shuning uchun asossinfga ko'rsatkichning istalgan vaqtda qanday turdagi ob'ektga murojaat qilishihardoim ham oldindan ma'lum emas.

Bu faqat dastur ishlayotganida

dinamik turdagi identifikatsiya vositalaridan biri yordamida aniqlanadi.

Dastur bajarilishida davomida ob'ekt turini olish uchun `typeid` operatoridan foydalaniladi.

Dastur bajarilayotgan vaqtda ob'ekt turini olish uchun `<typeid>` kutubxonasi kiritish va `typeid` operatoridan foydalanish mumkin. `Typeid`

operatoridan foydalanishning eng keng tarqalgan yozilish sintaktis quyidagi cha:

```
typeid(object)
```

Bunda, object elementini olish uchun ob'ektanglatadi. Shuningdek, nafaqat ichki tipini, balki dasturchi tomonidan yaratilgan sinftipini ham aniqlash mumkin. Buning uchun typeid operatori ob'ekt tipini tasvirlaydigan type\_info ob'ektiga murojaat qiladi.

type\_info sinfda public modifikator bilan quyidagi statik operatorlar aniqlangan bo'ldi.

```
bool operator == (const type_info &ob);
bool operator != (const type_info &ob);

bool before(const type_info &ob);

const char *name();
```

Bunda quyidagi aniqlangan (overloaded) operatorlari [==] va [!=] tiplarini taqqoslash uchun ishlatiladi.

Chaqiruvchi ob'ekt parametrlar sifatida ishlatiladigan ob'ekt (element ob) oldidagi ketma-ketlik tartibidagi bo'lsa before() funksiyasi true qaytaradi. Bu funksiya asosan ichki foydalanish uchun mo'ljallangan.

Uning natijasi qiymatimero sxo'ryokisini fiyerarxi bilan hech qanday aloqasi yo'q. name() funksiyasi tip nomiga ko'rsatki qaytaradi.

typeid operatoridan foydalanishga birmisol keltiramiz.

## 1.1-dastur. Tiplarni taqqoslash<sup>1</sup>

```
#include "stdafx.h"
#include <iostream>
#include <typeinfo>
using namespace std;
class myclass {
// . . .
};
int main(){
int i, j;
float f;
myclass ob;
```

<sup>1</sup>Дастур Visual C++ muhitida tuzilgan.

```

cout << "i o'zgaruvchi tipi: " <<
typeid(i).name()<<endl;
cout << "f o'zgaruvchi tipi: " <<
typeid(f).name()<<endl;
cout << "ob o'zgaruvchi tipi: " <<
typeid(ob).name()<<endl<<endl;
if(typeid(i) == typeid(j))
    cout << " i va j tiplar bir xil.\n";
if(typeid(i) != typeid(f))
    cout << " i va f tiplar bir xil emas.\n";
system("PAUSE");
return 0;
}

```

#### 1.1 – dasturnatijasi. Output

```

i o'zgaruvchi tipi: int
f o'zgaruvchi tipi: float
ob o'zgaruvchi tipi: class myclass
i va j tiplar bir xil.
i va f tiplar bir xil emas

```

**1.1-dasturtahlili.**Dasturda int tipigaoid i va j, floattipidaf, myclasstipidaobo'zgaruvchilarianiqlangan.

typeid()metodibilanularningtiplarianiqlangan.

Taqqsolashoperatorlaribilano'zraguvchilarningtiplari tekshirilganvanatijalarekrangachiqarilgan.

Buesatiplarbo'yichadastlabkiishlovberishuchunlozim.

#### 1.1-jadval. Type\_info sinfa'zolarivavazifalari

№	Funksiyano mi	vazifasi
1	(constructor)	Sinfda standart vanusxakonstruktorimavjudemas.
2	(destructor) [virtual]	xavfsiztayanchsinfuchun pointer orqaliob'ektnio'chirishmumkin (virtual funksiyasi)
3	operator=	O'zlashtirishamalidanfoydalanibko'chiribbo'lmaydi
4	operator== operator!=	Ob'ektlarbirxiltipgaliginitaqqoslashydi (public funksiya)



5	<u>before</u>	Joriytipdagiboshqa ob'ekttomonidannusxasitipiniboryokiyo'qliginitekshira di.Ob'ektlar, amalgaoshirilishibelgilangantartibdatashkiletildi. (public funksiya)	type_index aniqrog'itiplari,
6	<u>hash_code</u>	Joriytipgamoskeladigantiplarniqaytaradi (public funksiya)	
7	<u>name</u>	Joriytipninganiqlangannominiqaytaradi. (public funksiya)	

Typeid operatoripolimorftayanchsinfuchunko'rsatkichsifatidaqo'llaniladiganbo'lsa (eslabtibo'tamiz, polimorfsinfkamidabitta virtual funksiyanio'zichigaolgansinfgaaytiladi), uavtomatikravishdarealob'ekttipiniqaytaradivauasossidanolingansinf tayanchsinfob'ektyokiob'ektemasliginihamko'rsatadi.

Shuninguchun typeid operatoridasossinfgako'rsatgichbilanmurojaatqilinganob'ekttipinidina mikravishdaaniqlashuchunfoydalanishmumkin. Buxususiyatquyidagidasturdako'rsatilgan.

1.2-dastur. Typeid operatorinipolimorfsinflarierarxiyasigaqo'llanilishi.

```
#include "stdafx.h"
#include <iostream>
#include <typeinfo>
using namespace std;
class Base {
virtual void f() {}; // Asos sinf uchun polimorf
// . . .
};
class myClass_one: public Base {
// . . .
};
class myClass_two: public Base {
// ...
};
int main(){
Base *p, baseob;
```

```

myClass_one ob1;
myClass_two ob2;
p = &baseob;
cout << " O'zgaruvchi p tipdagi obyektga ishora:
";
cout << typeid(*p).name() << endl;
p = &ob1;
cout << " O'zgaruvchi p tipdagi obyektga ishora:
";
cout << typeid(*p).name() << endl;
p = &ob2;
cout << " O'zgaruvchi p tipdagi obyektga ishora:
";
cout << typeid(*p).name() << endl;
system("pause");
return 0;}

```

## 1.2 – dasturnatijasi. Output

```

O'zgaruvchi p tipdagi obyektga ishora: class Base
O'zgaruvchi p tipdagi obyektga ishora: class
myClass_one
O'zgaruvchi p tipdagi obyektga ishora: class
myClass_two

```

**1.2-dasturtahlili.** Dasturdadinamikko‘rsatkichliBasetipli p o‘zgaruvchisiyaratilganva 3 ta, base tipiga baseob, myClass\_one tipiga ob1, myClass\_two tipiga ob2 moso‘zgaruvchilaryaratilgan. Dasturdavomidadinamiko‘zgaruvchigaoddatiyo‘zgaruvchilarqiymatqilib berilgan.

Natijashuma‘lumbo‘ldikidininamiko‘zgaruvchiningtipidanamikko‘rinishda o‘zgardir.

Typeid operatoripolimorfturdagitayanchsinfgako‘rsatgichsifatidaqo‘llanilsa, haqiqiymanzillanganob’ektningtipinidasturbajarilayotganvaqtdaaniqlaydi.

Typeid operatoripolimorfb o‘lmaganiyerarxiksinflaruchunko‘rsatkichqo‘llaniladigan barchahollarda, tayanchtipiuchunko‘rsatikichbo‘ladi. Boshqaso‘zlarbilanaytganda,

buko'rsatkichaslidatipnianiqlashmumkinemas. Tajribasifatida 1.2-dasturdagiasossinfidagi f() virtual funksiyaniizohgaolingvanatijanitekshiribko'ring.

Dasturdabuo'zgarishniamalgaoshirgandanso'ngharbirturlitipliob'ektning tipi base debaniqlanishiniko'rasiz, chunkibu p ko'rsatgichgaegabo'lgantipdir.

Typeid operatoriodatdakengqirraliko'rsatkich (dereferenced pointer ya'ni, [\*] operatorigaegabo'lganko'rsatkich) uchunqo'llanilganligisababli, bubuko'rsatkich null bo'libchiqqandavaziyatniboshqarishuchunmaxsusistisnoyaratiladi.

Buholda, typeid operatori bad\_typeid istisnogao'tadi.

Polimorfsinflarierarxiyasidagiob'ektlargahavolalarko'rsatgichlarka biishlaydi. Typeid operatoripolimorfsinfningmosyozuvlaruchunqo'llaniladiganbo'lsa, uaslidaob'ekttipiniqaytaradivauasosiytipio'rnigaolingantipdagiob'ektbo'lishimumkin.

Buusulo'zgaruvchilarnifunksiyalargamurojaatqilibo'tishdaengko'pqo'llaniladi. Masalan, quyidagi (1.3-dastur) dasturda WhatType() funksiyasiasosiytipdagiob'ektlargamosyozuvlarparametrinie'lonqiladi.

Demak, WhatType() funksiyasibazaviyturdagiob'ektlargamurojaatlarniyokiasossinfdanolinga nharqandaysinfob'ektlarigamurojaatlarnikeltiribchiqaradi.

Bundayparametrlargaqo'llaniladigan typeid operatorifunksiyagao'tganob'ektninghaqiqiyturiniqaytaradi.

1.3-dastur. Typeid operatorinihavolaliparametrigaqa'llash.

```
#include "stdafx.h"
#include <iostream>
#include <typeinfo>
using namespace std;
class Base {
virtual void f() {}; // Asos sinf uchun polimorf
// . . .
};

class myClass_one: public Base {
// . . .
};
class myClass_two: public Base {
// ...
```

```

};
void WhatType(Base &ob){
cout << " ob parametri ob'ekt tipiga havolasi: ";
cout << typeid(ob).name() << endl;
}
int main(){
Base baseob;
myClass_one ob1;
myClass_two ob2;
WhatType(baseob);
WhatType(ob1);
WhatType(ob2);
    system("pause");
return 0;
}

```

### 1.3 – dasturnatijasi. Output

```

ob parametri ob'ekt tipiga havolasi: class Base
  ob parametri ob'ekt tipiga havolasi: class
myClass_one
  ob parametri ob'ekt tipiga havolasi: class
myClass_two

```

**1.3-dasturtahlili.** Dasturda `WhatType` funksiyasi yaratilib, unga asossinftipidp gihavolalip parametrbelgilandi. Asosiy funksiyada 3 ta, base tipiga baseob, myClass\_one tipiga ob1, myClass\_two tipiga ob2 moso‘zgaruvchilaryaratilgan.

Dasturdavomida `WhatType` funksiyasininghavolalip parametr ga odati yo‘zgaruvchilarqiymatqilibberilgan. Natijashuma’lum bo‘ldiki, dinamik o‘zgaruvchiningtipidanamikko‘rinishdao‘zgardi.

Dasturlash jarayonlarida o‘zgaruvchilarni tiplarini real tip bilan taqqoslashgato‘g‘rikeladi. Buning uchun `typeid` operatori argument sifatida tipini olish yordamida amalga oshiriladi. Uning yozilishi quyidagicha:

```
typeid(tipnomi)
```

Masalan, quyidagilistingamalqiladi.

```
Cout<<typeid(int).name();
```

## Typeid

operatoriniushbuvariantdaishlatilishiningmaqsaditipnitaqqoslashuchunfoydalanishdir, (belgilanganma'lumotlartipinita'riflaydi) ammo, typeid operatoridanfoydalanishuchunemas.

Dinamikiptipnianiqlashni (RTTI) qo'llash. Dasturdadinamikiptipnianiqlash (identifikatsiya) vositasi (RTTI) imkoniyatifoysdalibo'lishimumkinlinginiko'rsatibo'tamiz. Masalan, geometrikshakllaruchuniyerarxiksinfberilganbo'lsin, undadoira, uchburchakvato'rtburchakmaydonihisoblashusullarimavjud.

Ushbudasturdadoira, uchburchakyokito'rtburchakniyaratishuchunmo'ljallanganfunksiyasi factory() bo'lsin. Ushbufunksiyayaratilganob'ektgabir nusxaniqaytadi. Ob'ektlarnihosilqiluvchifunksiyaba'zanob'ektfabrikasidebataladi.

Yaratilayotganob'ektningo'zigaxostipi C++ tasodifiysonlargeneratorining Rand() funksiyasigakirishorqalianiqlanadi. Demak, qandayturdagiob'ekthosilbo'lishinioldindanbilolmaymiz. 1.4-dasturdao'ntaob'ektyaratadivaharbirturdagiyaratilganob'ektlarsoninihiso blaydi.

factory() funksiyasinichaqirishdaharqandayshaklturinihosilqilishmumkinligisabab li, dasturaslidayaratilganob'ektipnianiqlashuchun typeid operatoridanfoydalanadi.

1.4-dastur. Dinamikiptidagiidentifikatsiyavositadanfoydalanish.

```
#include "stdafx.h"
#include <iostream>
#include <typeinfo>
#include <cstdlib>
using namespace std;
class figure {
protected:
    double x, y;
public:
    figure(double i, double j) {
        x = i;
        y = j;
    }
    virtual double area() = 0;
};
class triangle : public figure {
```

```

public:
    triangle(double i, double j) : figure(i, j) {}
    double area() {
        return x * 0.5 * y;
    }
};
class rectangle : public figure {
public:
    rectangle(double i, double j) : figure (i, j)
    {}
    double area() { return x * y;}
};
class circle : public figure {
public:
    circle(double i, double j=0) : figure(i, j) {}
    double area() {return 3.14 * x * x;}
};
// figure sinf ob'ektlarini yaratish.
figure *factory(){
    switch(rand() % 3 ) {
        case 0: return new circle (rand()%15);
        case 1: return new triangle
(rand()%15,rand()%15);
        case 2: return new rectangle (rand()%15,
rand()%15);
    }
return 0;
};
int main(){
    figure *p; // asos sinfga ko'rsatkich
    int i;
    int t=0, r=0, c=0;
        // 5 ta ob'ektlarni hosil qilsh va sanash
    for(i=0; i<15; i++) {
        p = factory(); // ob'ektni hosil qilish
        cout << "Ob'ektning tipi: " <<
typeid(*p).name();
        cout << ". ";

```

```

        // ob'ektni tekshirish orqali sanasj
        if(typeid(*p) == typeid(triangle)) t++;
        if(typeid(*p) == typeid(rectangle)) r++;
        if(typeid(*p) == typeid(circle)) c++;
        // Yuzasini hisoblash
        cout << " S= " << p->area() << endl;
    }
cout << endl;
cout << "Quyidagi ob'ektlar hosil qilindi:\n";
cout << " Uchburchaklar: " << t << endl;
cout << " To'rtburchaklar: " << r << endl;
cout << " Doiralalar: " << c << endl;
system("pause");
return 0;
}

```

#### 1.4 – dasturnatijasi. Output

```

Ob'ektning tipi: class rectangle. S= 8
Ob'ektning tipi: class triangle. S= 28
Ob'ektning tipi: class circle. S= 28.26
Ob'ektning tipi: class triangle. S= 35
Ob'ektning tipi: class rectangle. S= 12
Ob'ektning tipi: class triangle. S= 55
Ob'ektning tipi: class rectangle. S= 72
Ob'ektning tipi: class circle. S= 254.34
Ob'ektning tipi: class rectangle. S= 21
Ob'ektning tipi: class triangle. S= 33
Ob'ektning tipi: class rectangle. S= 10
Ob'ektning tipi: class circle. S= 379.94
Ob'ektning tipi: class circle. S= 254.34
Ob'ektning tipi: class triangle. S= 4
Ob'ektning tipi: class triangle. S= 36

```

```

Quyidagi ob'ektlar hosil qilindi:
Uchburchaklar: 6
To'rtburchaklar: 5
Doiralalar: 4

```

**Typeid operatorini shablonsinflariga qo'llash.** Typeid operatorini shablonsinflariga ham qo'llash mumkin albatta. Shablonsinfi sosisida hosil bo'lgan ob'ektning tipini ob'ektning amalga oshirishida uning umumlashgan ma'lumotlari uchun qanday ma'lumotlardan foydalanilganligini sosisida qisman aniqlanadi. Shuning uchun, turli ma'lumotlari yordamida yaratilgan bir xil shablonsinfi gategishli bo'lgan ob'ektlar tipini aniqlash mumkin.

1.5-dastur. shablonsinflar bilan typeid operatoridan foydalanish.

```
#include "stdafx.h"
#include <iostream>
#include <typeinfo>
using namespace std;
template <class T>
class myclass {
    T a;
public:
    myclass(T i) { a = i; }
    // . . .
};
int main(){
    myclass<int> o1(28), o2(1);
    myclass<double> o3(19.80);
    cout << " o1 ob'ekt tipi: ";
    cout << typeid(o1).name() << endl;
    cout << " o2 ob'ekt tipi: ";
    cout << typeid(o2).name() << endl;
    cout << " o3 ob'ekt tipi: ";
    cout << typeid(o3).name() << endl;
    cout << endl;

    if(typeid(o1) == typeid(o2))
        cout << " o1 va o2 ob'ektlar bir xil
tip.\n";
    if(typeid(o1) == typeid(o3)) cout <<
"Xatolik\n";
    else cout << "o1 va o3 ob'ektlar bir xil tip
emas.\n";
    system("pause");
}
```



```
return 0;
}
```

#### 1.5 – dasturnatijasi. Output

```
o1 ob'ekt tipi: class myclass<int>
o2 ob'ekt tipi: class myclass<int>
o3 ob'ekt tipi: class myclass<double>

o1 va o2 ob'ektlar bir xil tip.
o1 va o3 ob'ektlar bir xil tip emas.
```

1.5-dasturdanko‘rinibturibdiki, ikkiob‘ektlarbirxilshablonsinfob‘ektlaridabo‘lsa-da, ularningparametrlima‘lumotlarmosbo‘lmasa, ularturitengemas. Budasturda o1 ob‘ekti myclass <int >tipida, o3 ob‘ektiesamyclass <double>tipidabo‘ladi. Shundayqilib, buob‘ektlarturlixilliginiko‘rsatadi.

Tipeid operatorinishablonsinflarigaqo‘llashningyanabirmisolini, ya‘nil.4-

dasturgeometrikshaklnianiqlashdasturiningo‘zgartirilganvariantiniko‘rib chiqamiz. Busafar figure sinfinishablonsinfigaaylantiriladi.

1.6-dastur. Figure ierarxiyasiningshablonvarianti.

```
#include "stdafx.h"
#include <iostream>
#include <typeinfo>
#include <cstdlib>
using namespace std;
template <class T>
class figure{
protected:
    T x, y;
public:
    figure(T i, T j) {
        x = i;
        y = j;
    }
    virtual T area() = 0;
};
template <class T>
```

```

class triangle : public figure<T>{
    public:
        triangle(T i, T j) : figure<T>(i, j) {}
        T area() {
            return x * 0.5 * y;
        }
};
template <class T>
class rectangle : public figure<T>{
    public:
        rectangle(T i, T j) : figure<T>(i, j) {}
        T area() {
            return x * y;
        }
};
template <class T>
class circle : public figure<T>{
    public:
        circle(T i, T j=0) : figure<T>(i, j) {}
        T area() {
            return 3.14 * x * x;
        }
};
figure<double> *generator(){
    switch(rand() % 3 ) {
        case 0: return new circle<double> (rand() %
28);
        case 1: return new triangle<double>(rand()
% 28, rand() % 28);
        case 2: return new rectangle<double>
(rand() % 28, rand() % 28);
    }
    return 0;
}
int main(){
    figure<double> *p;
    int i;
    int t=0, c=0, r=0;

```

```

    for(i=0; i<15; i++) {
        p = generator();
        cout << "Ob'ekt tipi: " <<
typeid(*p).name();
        cout << ". ";
        if(typeid(*p) == typeid(triangle<double>))
t++;
        if(typeid(*p) == typeid(rectangle<double>))
r++;
        if(typeid(*p) == typeid(circle<double>))
c++;
        cout << " S= " << p->area() << endl;
    }
    cout << endl;
    cout << "Ob'ektlarni sanash:\n";
    cout << " uchburshaklar: " << t << endl;
    cout << " o'rtburchaklar: " << r << endl;
    cout << " doiralar: " << c << endl;
    system("pause");
    return 0;
}

```

#### 1.6 – dasturnatijasi. Output

```

Ob'ekt tipi: class rectangle<double>. S= 90
Ob'ekt tipi: class triangle<double>. S= 136
Ob'ekt tipi: class circle<double>. S= 615.44
Ob'ekt tipi: class triangle<double>. S= 210
Ob'ekt tipi: class rectangle<double>. S= 351
Ob'ekt tipi: class triangle<double>. S= 202.5
Ob'ekt tipi: class rectangle<double>. S= 44
Ob'ekt tipi: class circle<double>. S= 803.84
Ob'ekt tipi: class rectangle<double>. S= 156
Ob'ekt tipi: class triangle<double>. S= 30
Ob'ekt tipi: class rectangle<double>. S= 225
Ob'ekt tipi: class circle<double>. S= 706.5
Ob'ekt tipi: class circle<double>. S= 1384.74
Ob'ekt tipi: class triangle<double>. S= 66.5
Ob'ekt tipi: class triangle<double>. S= 15

```

Ob'ektlarni sanash:  
uchburshaklar: 6  
o'rtburchaklar: 5  
doiralar: 4

Dinamik tipni aniqlash har bir dasturda ishlatilmaydi. Ammo, dastur bajarilish vaqtida, polimorfturlar bilan ishlashda, undagi ob'ektlarning har qanday tipini aniqlash gaimkoniniberadi.

***Tiplarni almashtirish operatori.*** C++  
tilida beshta tipni almashtirish operatori mavjud.

Ulardan birinchi sibuodatdagi (an'anaviy) uslubda ishlatiladigan [()](cast operatori) azaldan C++ ga qurilgan. U quyidagicha ishlatiladi:

```
float f = (float)5;  
bool b = (bool)5;
```

Tipni almashtirish to'rtta operatori (dynamic\_cast, const\_cast, reinterpret\_cast va static\_cast) 10 yil oldin C++ tilga qo'shildi. Bu operatorlarni tiplar bilan amallarni bajarishga dastur xilimkoniyatlarini yaratib beradi. Ularning har birini alohida ko'rib chiqaylik.

***Dynamic\_cast operatori.*** Dynamic\_cast operatori dastur bajarilishida vomida polimorfturda quyish amalini bajaradi. Yangi operatorlardan eng muhimi dynamic\_cast tipidagi tipni almashtirish (kasting, casting) operatori dir. Dastur bajarish davomida talif qilinayotgan amalning bajarilishini tekshiradi. Agar belgilangan amal chaqirilgan dasturda bo'lsa, hech qanday turdagi kasting amalga oshirilmaydi. Dynamic\_cast operatorini qo'llashning umumiy formati quyidagicha:

```
dynamic_cast<type> (expr)
```

Bunda type elementibu amalning maqsadibo'lganyangitipni, xprelementiesabuyangitipga almashadiganifodanibildiradi.

Tipturiko'rsatkichyokimosyozuvlar bilan almashish kerak va expr ifodako'rsatkichyokimosyozuvlar uchun tashlanish kerak. Shu tarzda dynamic\_cast operatoridan birturda giko'rsatgichni boshqaturdagiko'rsatgichga yoki birturdagimurojaatni boshqaturdagimurojaatga aylantirish uchun foydalanish mumkin.

Bu operator asosan polimorftiplar orasidagi dinamik tipni quyish amallari uchun ishlatiladi. Masalan, agar polimorfti sinflar B va D bo'lsin. Sinf D va sinf V olingach, dynamic\_cast operatori yordamida,

hardoimko'rsatkichni Dgaaylantirishmumkin. Chunki \*in pointer\* tayanchsinf gako'rsatkichhardoimbazasidanolingansinfob'ektiniko'rsatis huchunfoydalanishmumkin. Biroq, dynamic\_cast operatorifaqatko'rsatkichaylantirishimumkin. D ob'ektko'rsatkichuchunmurojaatqilinganda, albatta, sinf D ob'ektibo'ladi. Umumanolganda, dynamic\_cast operatorifaqatpolimorftipdagikastinggaruxsatberilsa, muvaffaqiyatlibajariladi, ya'niyangitipdagiko'rsatikchushbuyangitipdagiob'ektgayokiundanolinga nob'ektgahavolaqilishimumkin. Aksholda, agarbelgilangantipdagialmashishamalinibajaribbo'lmasa, buamaldako'rsatgichlarishtirokesa dynamic\_cast operatoriningnatijasi null qiymatgaaylanadi. Agarushbuamalnibajarilishimuvaffaqitsizbo'lsa, uzilishlarishtiroketganbo'lsa, bad\_cast istisnosiavtomatikbajariladi.

Oddiy misol niko'ribchiqamiz. Base asossinf polimorfsinfbo'lsinva myClass sinfesa Base sinfidanolingandebfarazqilaylik.

```
Base *bp, b_ob;
myClass *dp, d_ob;
bp = &d_ob; // myClass sinfob'ektga Base sinfuchunko'rsatkich.
dp = dynamic_cast< myClass *> (bp); // myClass
sinfko'rsatgichgaalmashtirish
if(dp) cout << "Tipnialmashtirishmuvaffaqiyatlibo'ldi!";
```

Buyerdabpko'rsatkichini (asossinf) dpko'rsatkichiga (myClass sinf) almashtirishmuvaffaqiyatlibo'ladi, chunkibpaslidasinfob'ektigaishoraqiladi.

myClass shuninguchunushbumisolnibajarishdatipnialmashtirishmuvaffaqiyatlichiqdi. bp aslidasinf bazasiniob'ektgaishoraqilgandi. Quyidagimisol datipnialmashtirishda myClass sinfob'ektbo'lsa, ko'rsatgichgatayanchsinfgaob'ektnitashlashuchunmuvaffaqiyatsizvanoto'g'riolinganisoblanadi.

```
Base *bp, b_ob;
myClass *dp, d_ob;
bp = &b_ob; // Base sinfgako'rsatkichsifatidaBase sinfob'ektberilgan.
dp = dynamic_cast< myClass *> (bp); // xatolik
if(!dp) cout << "Tipnialmashtirishmuvaffaqiyatsiz ";
```

Ushbudasturfragmentidatipnialmashtirishamalimuvaaffaqiyatsizbo‘l  
divatipnialmashtirishuchunmuvaaffaqiyatsizbo‘lganliginiekrangachiqarad  
i.

dynamic\_cast operatoridanfoydalanishdaquyidagi 1.7-  
dasturdatipnialmashtirishholatlarianiqlaymiz.

1.7-dastur. dynamic\_cast operatoridanfoydalanish.

```
#include "stdafx.h"
#include <iostream>
#include <typeinfo>
using namespace std;
class Base {
public:
    virtual void f() { cout << " Base sinfi
orqali.\n"; }
// . . .
};
class myClass : public Base {
public:
    void f() { cout << "myClass sinfi
orqali.\n"; }
};
int main(){
    Base *bp, b_ob;
    myClass *dp, d_ob;
    dp = dynamic_cast<myClass *> (&d_ob);

    if(dp) {
        cout << " tipni almashtirish " <<" (myClass*
=> myClass *) bajarildi. ";
        dp->f();
    } else cout <<"Xatolik\n";
    cout << endl;
    bp = dynamic_cast<Base *> (&d_ob);

    if(bp) {
        cout << " tipni almashtirish " <<"(myClass*
=> Base *) bajarildi. ";
        bp->f();
    }
}
```

```

} else cout << "Xatolik\n";
cout << endl;

bp = dynamic_cast<Base *> (&b_ob);
if(bp) {
    cout << " tipni almashtirish " <<"(Base* =>
Base *) bajarildi. ";
    bp->f();
} else cout << "Xatolik\n";
cout << endl;
dp = dynamic_cast<myClass *> (&b_ob);
if(dp) cout <<"Xatolik\n";
else
    cout <<" tipni almashtirish " <<"(Base*=>
myClass*) bajarilmadi.\n";
cout << endl;

bp = &d_ob; // bp myClass ob'eyktiga
ko'rsatkich
dp = dynamic_cast<myClass *> (bp);
if(dp) {
    cout << " bp ni myClass tipiga almashtirish
bajarildi. ";
    dp->f();
} else cout << "Xatolik\n";
cout << endl;

bp = &b_ob; // br Base ob'eyktiga ko'rsatkich
dp = dynamic_cast<myClass *> (bp);
if(dp) cout << "Xatolik";
else cout <<" bp myClass tipiga almashtirish
bajarilmadi.\n";
cout << endl;
dp = &d_ob; // dp myClass ob'eyktiga
ko'rsatkich
bp = dynamic_cast<Base *> (dp);

if(bp) {

```

```

        cout <<" dp ni myClass tipiga almashtirish
bajarildi. ";
        bp->f();
    }
    else cout <<"Xatolik\n";
    system("pause");
return 0;
}

```

#### 1.7 – dasturnatijasi. Output

```

tipni almashtirish (myClass* => myClass *)
bajarildi. myClass sinfi orqali.
tipni almashtirish (myClass* => Base *) bajarildi.
myClass sinfi orqali.
tipni almashtirish (Base* => Base *) bajarildi.
Base sinfi orqali.
tipni almashtirish (Base*=> myClass*) bajarilmadi.
bp ni myClass tipiga almashtirish bajarildi.
myClass sinfi orqali.
bp myClass tipiga almashtirish bajarilmadi.
dp ni myClass tipiga almashtirish bajarildi.
myClass sinfi orqali.

```

Dynamic\_cast operatoridan typeid operatoriga nisbatan ishlatilish mumkin. Masalan, Base sinfi polimorf va myClass sinfi uchun asos sinfi, deb faraz qilaylik. Quyidagi dastur fragmentini bajarishda, ob'ekt aslida myClass sinfi ob'ekt bo'lsada dp ko'rsatkich bp ko'rsatkich tomonidan ob'ekt manziliga beriladi.

```

Base *bp;
Derived *dp;
// . . .
if(typeid(*bp) == typeid(Derived))
dp = (Derived *) bp;

```

Bu holda, odatiy tipni almashtirish ishlatildi. Bujudaxavfsiz, chunki if operatori typeid operatoriyordamida typeid amalining haqiqiy ligini bajarilishidan oldin tekshiradi. Typeid operatorini va if operatorini bilan almashtirish orqali yanada samaraliroq qilish mumkin.

dynamic\_cast:



```
dp = dynamic_cast<myClass *> (bp);
```

Dynamic\_cast

operatorifaqatob'ektningbelgilangantipiyokibelgilangantipialmashtirilgan tipibilanbirob'ektbo'lsamuvaaffaqiyatlinatijaberadi,

bualmashtirishtugagandanso'ng, dp

ko'rsatkicholingano'ektganullqiymatiniyokiko'rsatkichnihamo'zichiga oladi. Belgilangantipnitashlashamalito'g'ribajarilganbo'lsa

dynamic\_cast operatorifaqatmuvaaffaqiyatlibajariladi,

uningalmashtirishmantig'inimuayyanvaziyatlardasoddalashtirilganmum

kin. Quyidagi 1.8-dastur typeid operatorini dynamic\_cast operatoribilanalmashtirishmumkinliginiko'rsatilgan.

Buyerdabirxilamallarto'plamiikki marta bajariladi, masalan, avval typeid operatori, so'ngra dynamic\_cast operatoriyordamida.

1.8-dastur. Typeid operatorio'rniga dynamic\_cast operatoridanfoydalanish.

```
#include "stdafx.h"
#include <iostream>
#include <typeinfo>
using namespace std;
class Base {
public:
    virtual void f() {}
};

class MyClass : public Base {
public:
    void derivedOnly() {
        cout << "Bu MyClass ob'ekti.\n";
    }
};

int main(){
    Base *bp, b_ob;
    MyClass *dp, d_ob;
    // typeid operatoridan foydalanish
    bp = &b_ob;
    if(typeid(*bp) == typeid(MyClass)) {
        dp = (MyClass *) bp;
        dp->derivedOnly();
    }
}
```

```

    } else
        cout <<" Base tipidagi ob'ekt MyClass
almashtirilmadi. \n";
        bp = &d_ob;
        if(typeid(*bp) == typeid(MyClass)) {
            dp = (MyClass *) bp;
            dp->derivedOnly();
        } else
            cout <<"Xatolik, tip almashtirish yozilmagan!
\n";
        // dynamic_castoperatoridan foydalanish
        bp = &b_ob;
        dp = dynamic_cast<MyClass *> (bp);
        if(dp) dp->derivedOnly();
            else cout << " Base tipidagi ob'ekt MyClass
almashtirilmadi. \n";
                bp = &d_ob;
                dp = dynamic_cast<MyClass *> (bp);
                if(dp) dp->derivedOnly();
                    else cout << "Xatolik, tip almashtirish
yozilmagan! \n";
                        system("pause");
return 0;
}

```

#### 1.8 – dasturnatijasi. Output

```

Base tipidagi ob'ekt MyClass almashtirilmadi.
Bu MyClass ob'ekti.
Base tipidagi ob'ekt MyClass almashtirilmadi.
Bu MyClass ob'ekti.

```

1.8-dasturdanko‘rinadiki, asossinfningko‘rasatkichini MyClass ko‘rsatkichigaalmashtirishda dynamic\_cast operatoriniqo‘llashdasturmantig‘iniqulayroqqiladi. Yuqoridagidasturnatijasigaqarang. Shuningdek, dynamic\_cast operatorinishablonsinflargahamqo‘llashmumkin.

**const\_cast operatori.** Const\_cast operatori const va/yoki volatile o‘zgartirgichlarniqaytaaniqlashuchunxizmatqiladi.

Const\_cast operatori const va / yoki volatile o'zgartirgichlarni ochiq qayta aniqlash/joriy qilish uchun ishlatiladi. const yoki volatile ning atributlariga istisnolar sifatida, yangi tip joriy tip bilan mos bo'lishi kerak. Ko'pincha const\_cast operatori const atributini olib tashlash uchun ishlatiladi. Uning umumiy formati quyidagicha:

```
const_cast<type> (expr)
```

Buyerda type elementi yangi tipga almatirish tipini o'rnatadi, expr elementi yangi tipga almatirilishi kerak bo'lgan o'zgaruvchi (ifoda) ni bildiradi.

1.9-dastur. Const\_cast operatoridan foydalanish.

```
#include "stdafx.h"

#include <iostream>
#include <typeinfo>

using namespace std;

void f (const int *p){
    int *v;
    v = const_cast<int *> (p); // const-
o'zgaruvchini qayta almashtirish.
    *v = 1980; // yangi qiymat berish,
modifikatsiyalash
}

int main(){
    int x = 2004;
    cout <<" f() funksiya chaqirilmasdan oldin x ni
qiymati: " << x<< endl;
    f (&x);
    cout <<" f() funksiya chaqirilgandan so'ng x ni
qiymati: " << x<< endl;
    system("pause");
    return 0;
}
```

1.9 – dastur natijasi. Output

```
f() funksiya chaqirilmasdan oldin x ni qiymati:
2004
f() funksiya chaqirilgandan so'ng x ni qiymati:
1980
```

Ko'ribturganingizdek, xo'zgaruvchi qabul qilgan parametr `const` ko'rsatgich sifatida o'rnatilgan `vaf()` funksiyasitomonidan o'zgartirilgan.

`Const` atributini o'chirish uchun `const_cast` operatoridan foydalanish xavflidir. Ajabovosita ekanligini alohida ta'kidlash hlozim. Shuning uchun, juda diqqat bilan foydalanishni tavsiya qilamiz.

Faqat `const_cast` operatori `const` atributini o'chirish imkonligini bilib olingiz.

Boshqasozlar bilan aytganda, na `dynamic_cast`, `static_cast`, yoki `reinterpret_cast` ob'ekt `const` atributini o'zgartirish uchun foydalanish mumkin emas.

**`static_cast` operatori.** `Static_cast` operatori polimorf bo'lmagan tipdagi ko'rsatkichlarni almashtirish uchun foydalaniladi. Har qanday tipni almashtirish uchun foydalanish mumkin. Biroq, dastur bajarilgan vaqtda hech qanday tekshirish amalga oshirilmaydi.

`Static_cast` operatori quyidagi umumiy yozuv formatiga ega:

```
static_cast<type> (expr)
```

Buyerda `type` elementi yangi tipdagi almashtirish tipini belgilaydi va `expr`

elementi yangi tipga tipni almashtirilish kerak bo'lganifodan bildiradi.

`Static_cast` operatori tipni almashtirishning original operatoridir. U faqat polimorf bo'lmagan tiplarni amalga oshiradi. Masalan, quyidagi 1.10-dastur ni ishlaganda `float` tipidagi xo'zgaruvchi `int` tipiga aylantiradi.

1.10-dastur. `Static_cast` operatoridan foydalanish.

```
#include "stdafx.h"
#include <iostream>

using namespace std;

int main(){
    int i;
    float f;
    f = 199.22F;
    i = static_cast<int> (f);
    cout << i;
    return 0;
```

```
system("pause");  
}
```

reinterpret\_cast operatori. Reinterpret\_cast operatorifundamental tipni almashtirishni amalga oshiradi.

Reinterpret\_cast operatori tipni tubdan farqqiladigan tipga aylantiradi. Masalan, bir ko'rsatkichni int tipiga hamda bir ko'rsatkichni bir int qiymatiga aylantirish uchun foydalanishingiz mumkin.

Bundan tashqari, tabiiy mos kelmaydigan ko'rsatkich tiplarni tashlash uchun foydalanish mumkin. Ushbu operator quyidagi umumiy formatga ega:

```
reinterpret_cast<type> (expr)
```

Buyerda type elementi yangi tipdagi tipni almashtirishni belgilaydi va expr elementi yangi tipga almashtiriladigan ifodani bildiradi.

Reinterpret\_cast operatoridan foydalanish quyidagi 1.11-dastur ko'rib chiqamiz.

1.11-dastur. Reinterpret\_cast operatoridan foydalanish.

```
#include "stdafx.h"  
#include <iostream>  
  
using namespace std;  
  
int main(){  
    int i;  
    char *p = "Salom Buxoro";  
    i = reinterpret_cast<int> (p); // ko'rsatkichni  
    int tipiga almashtirish.  
    cout << i << endl;  
    system("pause");  
    return 0;  
}
```

Buyerda reinterpret\_cast operatori p ko'rsatkichni butun son int qiymatiga aylantiradi. Buzgartirish asosiy tipni o'zgarishifodalaydi. Bu tipni almashtirishni oddiy kasting bilan ham amalga oshirish mumkin.

Yangi to'rtta cast operatorlar bilan an'anaviy tipni almashtirish amali (kasting) qanday farq mavjud. Ba'zi dasturchilar to'rt cast operatorlari yuqoridagi virlangan butunlay an'anaviy tipni almashtirish amali ining o'rniga de bo'ylashlarim mumkin.

Ular noma'na qat'ib o'qilgan? — degan savol qiziq tirish mumkin. Aslida barcha dasturchilar uchun umumiy qoida yo'q.

Yangi operatorlar birma'lum o'ttira boshqasiga almashtirishni anchaxavfli va xavfsizligini yaxshilash uchun yaratilganligi sababli, ko'pgina C++ dasturchilar faqat shu maqsadda ishlatilish kerakligiga shonch hosil qilgan. Bunga e'tiroz qilib bo'lmaydi.

Boshqa dasturchilar anaviy tipialmashtirish amaliko'pyillarda vomi daula rga "sadoqat" bilan xizmat qilib bergan, undan shunday vozkeshish sonemas. Masalan, oddiy nisbatan xavfsiz tipdagikasting amallarini bajarish uchun read() va write() funksiyalarini chaqirganda talab qilinadi.

Ammo shunday dasturlashdajarayonlar borki, ularda cast operatorlaridan foydalanish maqsad gamuvofiqdir, masalan, polimorftiplar uchun cast amallarini bajarishda albatta dynamic\_cast operatoridan foydalanish gato'g'rikeladi.

### ***Yangi nomlar fazosi yaratish.***

Nomlar fazosi haqida qisqachama'lumotga ega bo'lsangiz kerak. Endi butun hunchanibatafsil roqko'ribchi qamiz va foydalanuvchining nomlar fazosi qanday yaratilishini ko'ribchi qamiz. Namespaces tushunchasi C++ dasturlashtirilgan nisbatan yangi qanday paydobo'lgan tushunchalardabiridir. Ularnoma'larningo'zarobirxilligini oldini olish (kolliziyasqaqlanish) uchun identifikatorlarning nomlarini lokalizatsiya qilish uchun mo'ljallangan. C++ dasturlash muhitida, o'zgaruvchani nomlari aqlbovarqilmaydigan bir qator operatorlari, funksiyalari, metodlarini va sinflar mavjud. Nomlar fazosi tushunchasi kiritilishidan oldin, bunoma'larning barchasi birxil global nomlar bo'lib, ko'plab nizolarni (xatoliklarni, o'zarobirxilliklarni) keltiribchi qargan. Agarda dasturda dasturchi o'zining toupper() funksiyasini yaratgan bo'lsa, masalan, agar, u (uning parametrlarini ro'yxatiga qarab) standart kutubxona toupper(), funksiyasini almashtirish mumkin. Ammo, ikkalafunksiyalarning nomlarini birxil global namespaceda saqlangan bo'ladi. Birxildasturlarni turli ishlabchi qaruvchilarning funksiyalarini va sinflar, kutubxonalar foydalangan nomlarini birxil bo'lgan funksiyalardastur bajarilishi da ziddiyatlargaduchkeladi. Bunday holda,

birkutubxonadaaniqlangannomlarboshqakutubxonadaaniqlangannomlarbilanto‘qnashishimumkin.

Barchamuammolarnomlarfazositushunchasiva namespace kalitso‘z tomonidan hal qilindi.

Bu kalitso‘z yangi nomlar fazosie‘lon qilish orqalibarcha funksiya, sinf va kutubxonalarni mahalliy lashtirishimkonini beradi.

Bu nomlar fazosi turlik kontekstlarda bir xil nomdagi funksiyavi sinflar ziddiya tgaolib kelmasdan ishlatishimkonini beradi. std standart kutubxonasi (std - Standart Library) judako‘p foydalanilgan bo‘lishimumkin. Tilning dastlabki variantlarida C++ kutubxonasi global nomlar fazosidabelgilangan.

Sezilarlinomlar ziddiyatlariehtimolinikamaytirish uchun std namespace nomlar fazosi yaratilgan. Bundantashqari,

ziddiyatlarni oldini olish uchun nomlar fazosiko‘lamin mahalliy lashtirish uchun dasturda o‘z nom fazosini yaratishimumkin.

Bu shaxsiy sinfyoki funksiyakutubxonalar yaratishda, ayniqsa, muhim ahamiyatga ega.

Namespace kalitso‘z nomlar fazosidaso halarni qo‘shib global namespace dan ajratishimkonini beradi. Aslida,

bir nomlar fazosidao‘irasibelgilaydi. Namespace kalitso‘zidan foydalanishning asosiy shakli quyidako‘rsatilgan:

```
namespace Space{
    //....
}
```

Namespace kontentida aniqlangan ixtiyoriy har bir element, bu namespace doirasida bajariladi.

1.12-dastur. myNameSpace nomli namespace e‘lon namuna.

```
namespace MyNameSpace {
    int i,k;
    void myfunc (int j) { cout <<j; }
    class myclass {
        public:
            void set(int x) { i = x; }
            int get() { return i; }
    };
};
```

MynameSpace nomli namespace da i va k o‘zgaruvchilarning nomlari, myfunc() funksiyasi va myclass sinfi yaratilgan.

Bu namespace ichida o'zgaruvchilarga, funksiyalarni deklaratsiya qilish va o'zgaruvchilarni deklaratsiya qilish uchun kerak bo'lgan kodni bir joyga to'plash imkonini beradi. Biroq, namespace kalit so'z ma'lum bir dastur sohasini belgilaydi, chunki, namespace tashqaridan namespace ichida o'zgaruvchilarning foydalanish imkoniyatini faqat ziddiyatlarni oldini olish uchun zarur bo'ladi. Masalan, MyNamespace nomli namespace tarkibiga kirmagandasturning bir qismida i o'zgaruvchiga 28 qiymat berish uchun quyidagicha ifoda yozish lozim:

```
MyNameSpace::i = 28;
```

MyNamespace nomli namespace sohasiga kiradigan MyClass tipidagi ob'ektning yaratilish imkoniyati, amma kirmagan MyClass tipidagi ob'ektning yaratilish imkoniyati:

```
MyNameSpace::myclass ob;
```

Shunday qilib, busohat tashqaridan nomlar fazosining nomi bilan kengaytirish operatori (::) bilan murojaat qilish mumkin.

Agarda dasturda namespace a'zolariga tez murojaat bo'ladigan bo'lsa, hardimuning nomi vakengaytirish operatori yozish dasturining ko'rinishiga va dastur chining ko'pyozishiga sabab bo'lishi mumkin. Ushbu muammoni hal qilish uchun foydalanuvchining nomlar fazosidan foydalanish fragmenti ishlab chiqilgan. Ushbu fragmenti kitta asosiy shaklga ega:

```
using namespace MyNameSpace;
```

```
using MyNameSpace::myclass;
```

Fragmentning birinchi qatorida keltirilgan shaklda a'zolar kirmoqchib o'lgan nomlar fazosining nomi bildiradi. Foydalanish fragmentida boshqacha yo'l bilan yaratilgan nomlar fazosidagi barha a'zolar gajori nom mavjud deb olinadi va o'zgaruvchilarning deklaratsiya qilish imkoniyati, fazo nomi vakengaytirish operatori harsafari yozish kerak bo'lmaydi.

Fragmentning ikkinchi qatorida using fragmentining ikkinchi shakli bilan foydalanilganda faqat fragmentdako'rsatilgan nom dosha'zos ko'rinadi. Masalan, yuqoridagi ifodada riqlangan mynamespace nomli namespace bo'lsa, unda quyidagi fragmentlar va operatorlaridan foydalanish to'g'ri:

```
using MyNameSpace::k;
```



```
int main(){
    k = 10;
    return 0;
}
```

Fragmentasida faqatko'zgaruvchiko'rinadi.  
ko'rsatmato'g'ri.

Demak,

```
using namespace MyNameSpace;

int main(){
    k = 10;
    i = 10;
    myclass ob;
    myfunc(25);
    return 0;
}
```

Bu fragmentda MyNameSpace  
fazoning barcha a'zolariko'rinadivaDemak, ko'rsatmato'g'ri.

Shuningdek,  
bir xil nom bilan bir nechta fazoni e'lon qilish imkoniyati mavjud.  
Bu bir nechta fayllar ichiga fazolarni ajratish imkonini beradi,  
yoki hatto bitta fayl ichida fazolarni ajratish imkonini.  
Bu nida stur chining dasturga ishlatayotgan funksiyavi va sinflarini ajratish uchun  
nqo'llash mumkin. Quyidagi misol niko'ribchiqamiz:

```
namespace NS { int a;}
namespace NS { int b;}

int main(){
    NS::a = 10;
    NS::b = 10;
    return 0;
}
```

Bu yerda NS namespace ikki qismga bo'lingan. Shunga qaramay,  
har bir qismi a'zolariga ham bir xil nom bilan murojaat qilingan, ya'ni NS  
fazo.

Bir dastur fayli da bir nomli fazolarni e'lon qilish maqsadga muvofiq.  
Ammo, dasturda 2  
va undan ortiq dastur fayllari ishlatilsa va fazolarni birlashtirish kerak bo'lsa,  
shunda fazoni ikki ga ajratish usulidan foydalanish mumkin.  
Boshqaso'zlar bilan aytganda, bir namespace faqat boshqa namespace

bo‘lishimumkinemas.

Bubirnomlifazoe‘lonqilinishimumkinemasdeganma‘nonianglatadi, masalan, birfunksiyaichida.

Nomsizfazo - fazoningmaxsusturidir.

Muayyanfaylichidanoyobidentifikatorlar, a‘zolari ni yaratishimkoniniberadi.

Nomsizfazoningasosi yshakli quyidagicha bo‘ladi:

```
namespace {  
    const float f = 3.1415;  
    int myIf(int a, int b){return a>b?1:0;}  
}
```

Fazolarninomsiznomlashlarfaqatbittafayldoirasidama‘lumbo‘lgannoyobidentifikatorlarvafunksiya, sinflarnio‘rnatishimkoniniberadi. Shutarzda, noma‘lumnomlifazonio‘zichigaolganfaylichida, ushbufazoninga‘zolarigato‘g‘ridan-to‘g‘ri, hechqandaytushuntirishgamuhtojbo‘lmasdanfoydalanishmumkin.

Ammobufayldantashqaridabundayidentifikatorlarnoma‘lumbo‘libqoladi.

Sizkichikvao‘rtadasturlaruchuno‘zfazolaringizniyaratishingizshartemas. Ammo, agarqaytafoydalanishuchunmo‘ljallanganfnuksiyalar, ssinflarva/yokisinflarkutubxonalaryaratishuchunba‘zi namespace ichidasizningkodlaringizjoylashtirishyozgandasturingiznitushunarliroqqiladi.

1.13-

dasturdanomlarfazosiniyaratishvafoydalanishgamisolkeltirilgan.

1.13-dastur. Nomlarfazosiniyaratishvafoydalanish.

```
#include "stdafx.h"  
#include <iostream>  
using namespace std;  
// birinchi nomlar fazosi  
namespace NS1{  
    class demo{  
        int i;  
        public:  
            demo(int x) { i=x; }  
            void set(int x) { i=x; }  
            int get() { return i; }  
    };  
    char str[] = "Nomlar fazolarini o‘rnatish\n";  
    int counter;
```

```

}
// ikkinchi nomlar fazosi
namespace NS2 { int x, u; }

int main(){
    NS1::demo ob(10);
    cout <<" ob = " << ob.get() << endl;
    ob.set(99);
    cout << "Endi ob = " << ob.get() << endl;

    using NS1::str; cout << str;

    using namespace NS1;
    for(counter = 1; counter<=10; counter++)
        cout << counter << ends;
    cout << endl;

    NS2::x = 10; NS2::u = 20;
    cout << "o'zgaruvchilar x, y = " << NS2::x << ", "
    << NS2::u << endl;

    using namespace NS2;
    demo x_ob(x), y_ob(u);
    cout << "x_ob, y_ob ob'ektlarning qiymati:
" << x_ob.get(); cout << " " << y_ob.get() << endl;
    system("pause");
    return 0;
}

```

### 1.13 – dasturnatijasi. Output

```

ob = 10
Endi ob = 99
Nomlar fazolarini o'rnatish
1 2 3 4 5 6 7 8 9 10
o'zgaruvchilar x, y = 10, 20
x_ob, y_ob ob'ektlarning qiymati: 10 20

```

Dasturda birmuhimnuqtako'rsatilgan:  
birnechtanomlarnibirgalikda ishlatganda,

birnomiboshqasialmashtirmaydi.

Hozirgifragmentdaba'zinomlarnikiritganingizda, uningnomlario'shapaytdaboshqanomdagingnomlarningo'zidabo'lishidanqat'inazar, bufragmentdaqo'shiladi. Shundayqilib, dasturbajarilayotganvaqtda, std, NS1 va NS2namespaces global namespace bo'libqo'shildi.

Yuqoridaaytibotilganidek, fayllarorasidayokibittafaylichidanomlarfazosiniajratishmumkin, keyinesabunomlarfazosininga'zolarinimazmunanbirlashtiriladi. Birgalikdanomlarfazosinibirlashtirishmisoliniko'ribchiqamiz.

1.14-dastur. Namespace larnibirlashtirish.

```
#include "stdafx.h"
#include <iostream>
using namespace std;
namespace Demo {
    int a; // Demo fazoda a o'zgaruvchi
}
int x; // Global fazoda x o'zgaruvchi
namespace Demo {
    int b; // Demo fazoda b o'zgaruvchi
}
using namespace Demo;
int main(){
    a = b = x = 101;
    cout << a << " " << a << " " << x << endl;
    system("pause");
    return 0;
}
```

1.14 – dasturnatijasi. Output

101 101 101

Bumisolda a va b o'zgaruvchilarihambirxilfazoda - Demo fazodava x o'zgaruvchisi global fazodae'lonqilingan.Ammo, using namespace Demo; fragmentibilan, a va b o'zgaruvchilarinivaglobalo'zgaruvchiniishlatishmumkin.

Yuqoridaaytibotilganidek, C++ standart o'z std fazosigaegavabutunkutubxonalariniqamraboladi.

Shuninguchunushbukitobdagibarchadasturlarquyidagiko'rsatmagaega:

```
using namespace std;
```

Fragment `std namespace` joriy qiladivabevosita C++ standart kutubxonasi deb belgilangan funksiyalar va sinflar nomlarini kirishim ko'rinib eradi. `std namespace` kengaytirish operatori yordamida har doim ishlatish maqsadga muvofiq emas.

Biroq, agar xohlasangiz, har bir identifikator dan oldin `std namespace` nomi ni kengaytirish operatori ni qo'yishingiz mumkin, xatobo'lmaydi. Masalan, quyidagi 1.15-dasturda standart `std` kutubxonasi global doirada kiritilmagan.

1.15-dastur. Ochiq foydalanish uchun nom ko'rsatilgan.

```
#include "stdafx.h"
#include <iostream>
int main(){
    double val;
    std::cout << "Son kiriting:";
    std::cin >> val;
    std::cout << "Son:";
    std::cout << val << std::endl;
    system("pause");
    return 0;
}
```

Ushbu dasturda ko'rsatilganidek, standart kirish va chiqish oqimlari `cin` va `cout` operatorlaridan foydalanish uchun nomlaridan oldin ularning qaysi fazodanligini belgilash kerak.

Agar dasturingiz C++ standart kutubxonasi dan keng foydalanish zarur bo'lsa, `std namespace` ni global doirada kiritish shart emas. Ammo, agar sizning dasturingiz standart kutubxonaa'zolari uchun minglab murojaatlarni o'z ichiga olsa, hara'z uchun `std` fazosini chaqirish juda ham oson kechmaydi. Shuning uchun uning global kiritish nisbatan dasturchi son va `std` barcha standart identifikatorlarni o'z ichiga oladi.

Agar dasturingizda standart kutubxonadan faqat bir nechta a'zodan foydalansangiz, bu bir nechta a'zolarini alohida belgilash uchun `using` fragmentidan foydalanish qulayroq bo'lishi mumkin. Ushbu yondashuvning afzalligishundaki, siz dasturda ko'rsatilgan a'zolarini fazosini ko'rsatmasdan kiritishingiz mumkin va ayni paytda butun standart kutubxonani global nomda kiritishingiz shart emas. Masalan:

```

#include "stdafx.h"
#include <iostream>

using std::cin;
using std::cout;
int main(){
    double val;
    cout << "Son kiriting:";
    cin >> val;
    cout << "Son:";
    cout << val << std::endl;
    system("pause");
    return 0;
}

```

Buyerda standart kirishvachiqishoqimlari cin va cout operatorlarito‘g‘ridan-to‘g‘ri ishlatilishim mumkin, ammo ayni paytda std fazoning boshqa a‘zolar joriy soxadantashqarida qoldiriladi. Maslan, endl;

Avvalroq, C++ tilining kutubxonasi global nomdabelgilanganligihaqida aytib o‘tilgan edi. Agareski C++ dasturini yangilashingiz kerak bo‘lsa, siz ham unda foydalanish uchun namespace std foydalanish kerak bo‘ladi, yokikengaytirish operatoriva std namespace nomi kiritish lozim. Agaryangi uslub sarlavhalaribilaneski header fayllarnio‘rniga, ayniqsa, muhim ahamiyatga ega (\*.h). Eski header fayllar global namespace o‘z mazmunini joylashtiradivayangi uslub sarlavhalari uchun std namespace o‘z mazmunini joylashtiradi.

Agar, nomlar fazosini global sifatidabitta faylga o‘lonqilib, cheklamoqchibo‘lsangiz, stilidastatikdebe‘lonqilish kerak edi, ya‘ni, statik identifikator bilan. Masalan, quyidagi ikki fayl bir xildasturning bir qismideb faraz qilaylik:

Birinchi fayl	Ikkinchi fayl
<pre> static int counter; void f1(){     counter == 99; // to‘g‘ri } </pre>	<pre> extern int counter; void f2() {     counter = 10; // Xato } </pre>

counter o‘zgaruvchisi birinchi faylda aniqlanganligi uchun hamunibirinchi faylda ishlatilishingiz mumkin. Ikkinchi faylda counter

o'zgaruvchis bilan fragmentdagi extern identifikatoriniko'rsatishiga qaramay, bu o'zgaruvchini ishlatishga urinish xatolikka olib keladi. Birinchi fayldagi counter o'zgaruvchisini statik deb e'lon qilib, uning ko'lamini shu faylga cheklangan.

Statik identifikator bilan global o'zgaruvchilarning deklaratsiyalari C++ halihamma mavjud bo'lsa-da, bu qochish maqsadida eng yaxshi yo'l nomlar fazosidan foydalanish hisoblanadi, quyidagimisol dako'rsatilgandek.

Birinchi fayl	Ikkinchi fayl
<pre>namespace { int counter; } void f1(){     counter == 99; // to'g'ri }</pre>	<pre>extern int counter; void f2() {     counter = 10; // to'g'ri }</pre>

**Bufershtirilgan kiritish va chiqarish.** C++ tilida ekran va faylga IO dan tashqari kiritish va chiqarish vazifalarini lokalizatsiyalash sifatida massivlardan foydalanadigan qator funksiyalarni qo'llab-quvvatlaydi.

Massivlarga asoslangan kiritish / chiqarish konseptual (array-based I/O) echim gao'xshasada, C tilining IO funksiyalaridan analogdir (ayniqsa, sscanf() va sprintf() funksiyalari).

Massivlarning yordamida kiritish/chiqarish C++ tilida juda moslashuvchan va foydalidir,

chunki unda foydalanuvchi tomonidan yaratilgan o'zgaruvchilarning kiritish va chiqarishda ham foydalaniladi.

Massivlar bilan kiritish/chiqarishning barcha jihatlari niqamrabolish mumkin emas, ammo, eng muhim va tez ishlatiladigan xususiyatlari amalga oshirish mumkin.

Massivlarning yordamida

IO ni amalga oshirish uchun ham oqimlarni talab qilishini tushunish muhimdir. Oldingiko'nikmalarimizdan C++ da IO amaliyotini bilishimiz mumkin.

Biroq,

kiritish/chiqarishda parametrlar sifatida ob'ektlardan foydalanishning barcha afzalliklari haqida tashvishlarning ko'nikmalarimiz muhimdir. Shuningdek,

ma'lumot olish uchun,

siz bir nechta xususiyatlarni boshqaradigan funksiyalar bilan tanishishin

gizkerak.

Bufunksiyalaristalganoqimnima'lumxotiramaydonigabog'lashuchunmo'ljallangan. Buamallartugagandanso'ng, barcha IO xusisiyatlariallaqachondasturchibiladigan IO vazifalariyordamidaamalgaoshiriladideganko'nikmagaolibkeladi.

Kiritish/chiqarishob'ektlarisifatidamassivdanfoydalanishniboshlashdanoldin, <sstream>kutubxonasinidasturingizgakiritilganbo'lishikerak. Bukutubxonadagi istringstream, ostream, va stringstream sinflardanfoydalanishmumkin. Businflar, o'znavbatida, kiritish, chiqishvakirish/chiqishuchunoqimlardanfoydalanishasoslangan. Ios sinfibusinflaruchuntayanchsindir, shuninguchun istringstream, ostream va ifstream sinflariningbarchafunksiyalarivamanipulyatorlari istringstream, ostream va stringstream sinflaridahammavjud. Satrmassivinichiqishuchun ostream sinfkonstruktordanquyidagiasosiyshakldafoydalaniladi:

```
ostream chiqish_oqimi(char *bufer, streamsize size, openmode mode = ios::out);
```

Buyerda, chiqish\_oqimimassivlisatrbilanbog'laydiganoqimbo'lib, buffer pointer (buferlanganko'rsatkich) orqaliberiladi. size parametrimassivningo'lchaminibelgilaydi. Mode - bustandartrejimparametri,odatdanormalchiqishholatigao'rnatiladi, lekinsiz ios sinfdabelgilanganharqandaychiqishrejimiindentifikatorio'rnatishingizmumkin.

Chiqishuchunmassivochilganda, simvolarmassivgasig'gunchajoylashtiriladi.To'lganmassivbo'lishimumkininemas.

Ixtiyoriyto'lganmassivkiritish/chiqarishnixatolikkaolibkelishimumkin. Massivdagbelgilarnisonninianiqlashuchun pcount() funksiya-a'zosiishlatiladi.

```
int pcount()
```

Funksiyafaqatbiroqimbilanaloqadabo'lib, shumunosabatibilanchaqiriladivaumassivgayozilganbelgilarsoniniqaytaradi. Shujumladan null tugatishnibildiradi.

Undanmassivgakiritishuchun istringstream sinfkonstruktoriningquyidagishaklidanfoydalaniladi.

```
istringstream kirish_oqimi (const char * bufer) ;
```



Buyerda, bufer belgilari kiritiladigan massivga ko'rsatgich hisoblanadi va kirish\_oqimi mi kirish oqimi bilan ko'rsatiladi. Biror massivdankirish ma'lumotlarini o'qishda eof() funksiyasidan foydalaniladi va massivning oxiriga yetganida true qaytaradi.

Kirish/chiqish uchun massiv oqish uchun ostream sinfi konstruktor quyidagicha shakldan foydalaniladi:

```
ostream kch_oqimi(char * bufer, streamsize size,
openmode mode = ios_base::in | ios_base::out) ;
```

Buyerda, kch\_oqimi – bu kirish oqimi, belgilar o'qilgan uzunligidagi massivga bufer ko'rsatgich orqali kirish/chiqish ob'ekt sifatida ishlatiladi.

Yuqoridagi keltirilgan barcha IO vazifalari, shuningdek, oqim bilan ishlash, ikkilik, shu jumladan tasodifiy kirish uchun muhimligini tushinish lozim.

Massivlar uchun oqim sinflaridan foydalanish C++ standart tomonidan kam ko'lab qo'llaniladi. Kelajakda C++ versiyalarida oqim qo'llanilishi bo'lmasligi ham mumkin. Belgilarni massiv oqimlari uchun mo'ljallangan bir xil vazifalar C++da sinf-konteyner asosida taqomillashtirilgan.

1.

Massivga ma'lumotlarni yozish va chiqish uchun oddiy misol keltiramiz.

1.16-dastur. Massivga chiqarish.

```
#include "stdafx.h"

#include <iostream>
#include <ostream>
using namespace std;
int main(){
char buf[255]; // chiqarish uchun bufer
ostream ostr(buf, sizeof buf); // chiqarish
uchun massiv
ostr << "kirish/chiqish massiv oqimi bilan
ishlaydi.\n";
ostr << "oddiy kirish/chiqish kabi: \n " << 100;
ostr << ' ' << 123.23 << '\n';
// manipulyator dan ham foydalanish mumkin
ostr << hex << 100 << ' ' ;
```

```

// identifikator dam ham
ostr.setf(ios::scientific);
ostr << 123.45 << ' ';
ostr << ends;
// oxirgi natijani chiqarish
cout << buf;
system("pause");
return 0;
}

```

#### 1.16 – dasturnatijasi. Output

```

kirish/chiqish massiv oqimi bilan ishlaydi.
oddiy kirish/chiqish kabi:
100 123.2364 1.234500e+002

```

Dasturda IO operatorlari qayta aniqlangan, hamda kirish/chiqish uchun ajralmas manipulatorlar, a'zo funksiyalari va massiv dvn foydalanish dato'liq formati identifikatorlari oqimlar uchun foydalanilgan.

Bu barcha manipulyatorlar uchun amal qiladigan lokal sinflar uchun IO operatorlari yaratish imkonini beradi.

Yuqoridagi dasturda ends manipulyator massivni tugatish null belgisini qo'shish uchun ishlatiladi. Null belgi avtomatik ravishda massivni kiritilgan yoki yo'qligini amalga oshirishda asturchining o'ziga bog'liq. Shuning uchun, bu dastur chi uchun muhim, chunki massiv qatorining oxiriga null belgi yozish yaxshidir.

#### 1.17-dastur. Massivga ma'lumotlarni kiritish.

```

#include "stdafx.h"

#include <iostream>
#include <sstream>
using namespace std;
int main(){
char buf[] = "Salom Buxoro 1980 28.01 B";
istringstream istr(buf);
char str[80]; float f; char c; int i;
istr >> str >> i >> f >> c;
cout << str << ' ' << i << ' ' << f << ' ' << c <<
'\n';

```

```
system("pause");  
return 0;  
}
```

1.17 – dasturnatijasi. Output

SalomBuxoro 1980 28.01 B

Budasturbuferko‘rsatkichidamassivdamavjudma‘lumotlarnio‘qiydi vaqaytaishlabekrangachiqaradi.

Agarmassivoqimbilanbog‘langanbo‘lsa, faylkabiishlatiladi. Masalan, massivma‘lumotlarni buferko‘rsatkichibilano‘qishvadasturda eof() va get() foydalanishniifodalaydi

1.18-dastur. Massivlarasosida eof() va get() funksiyalarning IO dafoydalanish.

```
#include "stdafx.h"  
#include <iostream>  
#include <sstream>  
using namespace std;  
int main(){  
    char buf[] = "SalomBuxoro 1980 28.01 B";  
    istringstream istr(buf);  
    char c;  
    while(!istr.eof()) {  
        istr.get(c);  
        if (!istr.eof()) cout << c;  
    }  
    cout<<endl;  
    system("pause");  
    return 0;  
}
```

1.18 – dasturnatijasi. Output

SalomBuxoro 1980 28.01 B

Quyidagidasturmassivma‘lumotlarnikiritishvachiqarishgaoid.

1.19-dastur. Massivlaryordamidakirish/chiqish.

```
#include "stdafx.h"  
#include <iostream>  
#include <sstream>  
using namespace std;
```

```

int main(){
    char iobuf[255];
    stringstream iostr (iobuf, sizeof iobuf) ;
    iostr << "tekshrish \n";
    iostr << 100 << hex << ' ' << 100 << ends;
    char str[80]; int i ;
    iostr.getline(str, 79);
    iostr >> dec >> i;
    cout << str << ' ' << i << ' ';
    iostr >> hex >> i;
    cout << hex << i << endl;
    system("pause");
    return 0;
}

```

1.19 – dasturnatijasi. Output

tekshrish 100 64

Buferashtirilgan kiritish va chiqarishda C++ da IO niboshqarish vazifalari uchun a'zofunksiyalarini, identifikatorlarni va manipulyatorlarni formatlash orqalita' minlanadi. Identifikatorlar, funksiyalar va manipulyatorlar bir xil vazifa bajaradi - ular axborotni kiritish/chiqarish uchun maxsus formatni o'rnatadi. C++ da ekrandan/gakiritish/chiqarish mos ravishda cin va cout operatorlari yordamida amalga oshiriladi, ya'ni formatlash manipulyatorlari ushbu IO operatorlari bilan birgalikda ishlatiladi.

Identifikator vazifalari va formatlash manipulyatorlari o'rtasidagi farqlarga e'lonlaniladigan usuldir.

Endi formatlash obektlaridan foydalanish yo'llarini ko'rib chiqamiz.

Asosiy formatlash a'zofunksiyalari:

cout.fill('symbol'); to'ldiruvchi belgini o'rnatadi. Bunda symbol to'ldiruvchi belgi va tirnoq ichida beriladi.

cout.width(width\_field); maydon uzunligini berish. width\_field - pozitsiyalar soni, har bir pozitsiyada bitta simvol sig'adi.

cout.precision(number);

haqiqiy sonlardan nuqtadan keyingi belgilar sonini berish. number - haqiqiy sonlardan nuqtadan keyingi belgilar soni.

Funksiyalarga nuqta amaliy orqalilik kirishiladi va qavslar ichida kerakli argument yoziladi. Fill()

funksiyasiga argumentni bitta qo'shtirnoq bilan belgi sifatida yoki son(belgi

odi) sifatidayozishmumkin.  
 kirish/chiqishoqimlarniformatlashuchunbittafunksiyayetarliemas.  
 C++daindentifikatorlarbilanformatlashusullariniko‘ramiz.

Indentifikatorlarorqaliformatlashkirish/chiqishparametrlaridanbiri niyoqishyokio‘chirishimkoniniberadi. IO  
 nindentifikatornio‘rnatishuchun setf()  
 funksiyasiniishlatamizvaagarchiqishindentifikatornio‘chirmoqchibo‘lsan  
 giz, unsetf() funksiyasidanfoydalaniladi.  
 Quyidagiindentifikatornio‘rnatishvao‘chirishgadoirinstruksiyako‘rsatilga  
 n.

#### Chiqishindentifikatornio‘rnatish

```
cout.setf( ios::name_flag );
```

name\_flag - indentifikatornomi.  
 Chiqarishoperatorlariganuqtaorqaliruxsatberiladi. setf()  
 funksiyasibittaargumentqabulqiladivaindentifikatornomidir.

Indentifikatorlar ios  
 sinfigategishlibo‘lganligiuchunoldinshusinfnomiyoziladivakengaytirisha  
 maliorqalikerakliindentifikatortanlanadi.

#### Chiqishindentifikatornio‘chirish

```
cout.unsetf( ios::name_flag );
```

name\_flag - indentifikatornomi.

Agarkirish/chiqishdabirnechtaidentifikatorlarniishlatishkerakbo‘lsa  
 , Yoki [] mantiqiyamalidanfoydalanishmumkin.

#### Birnechtaidentifikatorlarnio‘rnatish

```
cout.setf( ios::name_flag1 | ios:: name_flag2 | ios:: name_flag_n );
```

#### Birnechtaidentifikatorlarnio‘chirish

```
cout.unsetf( ios::name_flag1 | ios:: name_flag2 | ios:: name_flag_n );
```

### 1.1-

jadvalasosiyformatlashidentifikatorlarinikeltirilganvaulardanfoydalanish  
 gadoirmisollarko‘rsatilgan.

#### 1.1-jadval. C++daformatlashidentifikatorlari.

Identifi- kator	Vazifasi	misol	Misolna tija-si
boolalpha	Mantko‘rinishidamanti qiykattaliknikiritish (true, false)	cout.setf(ios::boolalpha); bool log_false = 0, log_true = 1; cout << log_false << endl	false true

		<< log_true << endl;	
oct	Sakkizlik sanoq sistema sidak kattaliklarni kiritish (avvaldec identifikatorni o'chiramiz va oct identifikatorni natamiz)	cout.unsetf(ios::dec); cout.setf(ios::oct); int value; cin >> value; cout << value << endl;	kirish: 9910 chiqish: 1438
dec	O'nlik sanoq sistemasidak kattaliklarni kiritish/chiqarish (identifikator joriy o'ratilgan)	cout.setf(ios::dec); int value = 148; cout << value << endl;	148
hex	O'noltilik sanoq sistemasidak kattaliklarni kiritish (avvaldec identifikatorni o'chiramiz va oct identifikatorni natamiz)	cout.unsetf(ios::dec); cout.setf(ios::hex); int value; cin >> value; cout << value << endl;	kirish: 9910 chiqish: 6316
showbase	Sanoq sistemaga asoslangan indikatorni chiqarish	cout.unsetf(ios::dec); cout.setf(ios::oct   ios::showbase); int value; cin >> value; cout << value << endl;	kirish: 9910 chiqish: 01438
uppercase	16lik SSdakat taharflarni chiqarish (joriy holda daki chikharflar o'ratilgan)	cout.unsetf(ios::dec); cout.setf(ios::hex   ios::uppercase); int value; cin >> value; cout << value << endl;	kirish: 25510 chiqish: FF16
showpos	Musbat sonlar uchun + ishorasini chiqarish	cout.setf(ios::showpos); int value = 15; cout << value << endl;	+15
scientific	Eksponensial sonlarni chiqarish	cout.setf(ios::scientific); double value = 1024.165; cout << value << endl;	1.02416 5e+003
fixed	Fiksirlangan holda haqiqiy sonlarni chiqarish	double value = 1024.165;	1024.16 5

	(joriyholatda)	cout << value << endl;	
right	O'ngtomondantekslash	cout.width(40); cout << «cppstudio.com» << endl;	__cppst udio.co m
left	Chaptomondantekislas h	cout.setf(ios::left); cout.width(40); cout << «cppstudio.com» << endl;	cppstudi o.com__

Shuningdek, manipulyatorlar bilan formatlashning usullari ham bor. manipulyator — maxsus tip bo'lib, oqimlarga o'tgan ma'lumotlarni formatlash uchun kiritish/chiqish oqimlarini nazorat qiladi. Lekin, ko'pchilik manipulyatorlarining funksiyalari identifikatorlar bilan bir xil formatlash amallarini bajaradi. Ba'zida, identifikatorli yoki formatlash funksiyalaridan foydalanish osonroq bo'lsada, ba'zan formatlashda manipulyatorlaridan foydalanish qulayroq. Shuning uchun C++ da kiritish / chiqishni formatlashning bir nechta vositalari bilan o'tirilgan. 1.2-jadvalda formatlashning manipulyatorlari va misollar keltirilgan.

1.2-jadval. Formatlashning manipulyatorlari.

Manipulyator	vazifasi	misob	natijasi
endl	Chiqarishdayangi qatorga o'tish	cout << «Salom:» << endl << «mbbGroup »;	website: cppstudio.c om
boolalpha	Mantiqiy kattalikni matn ko'rinishidagi chiqarish (true, false)	bool log_true = 1; cout << boolalpha << log_true << endl;	true
noboolalpha	Mantiqiy kattalikni son ko'rinishidagi chiqarish (true, false)	bool log_true =	1

		<pre>true; cout &lt;&lt; noboolalpha &lt;&lt; log_true &lt;&lt; endl;</pre>	
oct	8 SSdasonlarnichiqarish	<pre>int value = 64; cout &lt;&lt; oct &lt;&lt; value &lt;&lt; endl;</pre>	1008
dec	10 SSdasonlarnichiqarish	<pre>int value = 64; cout &lt;&lt; dec &lt;&lt; value &lt;&lt; endl;</pre>	6410
hex	16 SSdasonlarnichiqarish	<pre>int value = 64; cout &lt;&lt; hex &lt;&lt; value &lt;&lt; endl;</pre>	408
showbase	SSuchunasosiyindikatornichiqarish	<pre>int value = 64; cout &lt;&lt; showbase &lt;&lt;hex &lt;&lt; value &lt;&lt; endl;</pre>	0x40
noshowbase	SSuchunasosiyindikatornichiqarish maslik (joriyholat).	<pre>int value = 64; cout &lt;&lt; noshowbase &lt;&lt; hex &lt;&lt; value &lt;&lt; endl;</pre>	40
uppercase	16 SSdakattaharflarnichiqarish (joriyholatdakichikharflar)	<pre>int value = 255; cout &lt;&lt; uppercase &lt;&lt; hex &lt;&lt;</pre>	FF16



		value << endl;	
nouppercase	16 SSdakilichikharflarnichiqarish (joriyholat)	int value = 255; cout << noupper case << hex << value << endl;	ff16
showpos	Sonlarni + belgisibilanchiqarish	int value = 255; cout << showpos<< value << endl;	+255
noshowpos	Sonlarni + belgisizchiqarish (joriyholat)	int value = 255; cout <<noshowpos<< value << endl;	255
scientific	Sonlarniekspontensialko‘rinishdachi qarish	double value = 1024.165; cout << scientific << value << endl;	1.024165e+003
fixed	Fiksirlashholdasonlarnichiqarish (joriyholat).	double value = 1024.165; cout << fixed << value << endl;	1024.165
setw(int number)	Maydonuzunliginio‘rnatish, number — simvollarningpozitsiyalarisoni, parameterlimonupulyator	cout << setw(40) << “mbbGroup” << endl;	— mbbGroup

right	O'ngdantekslash	cout << setw(40) << right << « mbbGroup » << endl;	— mbbGroup
left	Chapdantekislash	cout << setw(40) << left << « mbbGroup » << endl;	mbbGroup —
setprecision(int count)	Verguldankeyikalitbelgiberish, count — verguldankeyingibelgilarsoni	cout << fixed << setprecision (3) << (13.5 / 2) << endl;	6.750
setfill(int symbol)	Qo'shimchabelgilarnio'rnatish	cout << setfill('0') << setw(4) << 15 << ends << endl;	0015

C++

dabuferlashtirilganformatlangankirish/chiqishdasturlashdagiengoddiyma vzulardanbirihisoblanadi.

Ma'lumformatlashvositalaridanqandayfoydalanishyuqoridagijadvallarda ko'rsatilgan,

shuninguchunbumavzudahechqandayqiyinchilikbo'lmasligikerak.

## NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1. Polimorfsinfqandayxususiyatgaegabo'lishilozim?



2. Dasturbajarilishidavomidaob'ektturiniolishuchunqandayoperatorda nfoydalaniladi?
3. `type_info` sinfda `public` modifikatoribilanqaysistatikoperatorlaraniqlangan?
4. `Type_info` sinfa'zolaridagiqandayoperatordanfoydalanishtaqiqlangan?
5. `Typeid` operatorinihavolaliparametriga qo'llashmumkinmi?
6. C++ tilidanechatipnialmatirishoperatorlarimavjudvaularninomlari, ishlatishuslublariniayting.
7. `Dynamic_cast` operatoridasturbajarilishidavomidaqaysiturdagiquyishamalinibajaradi?
8. `Dynamic_cast` va `const_cast` tipalmashtirishoperatorlariningvazifalari, kamchiliklarivayutuqlarieslang.
9. O'zgarmase'lonqilinganidentifikatorningtipinialmashtirishmumkin mi. Mumkinbo'lsa qanday amalga oshiriladi. Mumkinbo'lmasa, nimasababdanizohlayubering.
10. Polimorfbo'lmaganvabo'lgansinfob'ektiningtipinialmashtirishuchunqaysioperatorlardanfoydalaniladi, harbiriniizog'layubering.
11. To'rtta `cast` operatorlari `dynamic_cast`, `const_cast`, `static_cast`, `reinterpret_cast` bilan an'anaviy tipnialmashtirishamali (kasting) qanday farq mavjud.
12. Yanginomlar fazosiyaratishuchun kerak, misollar bilan eslab tushuntirib bering.
13. Jamoaviy dasturlash vaqtidasturdaganomlarning ziddiyatlarni oldin iolishuchun nimavazifani amalga oshirish kerak vaziddiyatlarni bekor qilishning eng oddiy va professional usulini tushuntiring.
14. Nomlar fazosini 2 ga ajratish mumkinmi, mumkinbo'lsa, qanday va nima uchun nomlar fazosini 2 ga ajratish kerak bo'ladi.
15. `Using` kalit so'zni ishlatmasdan turib, nomlar fazosi gamurojaat qilish mumkinmi? Mumkinbo'lsa, misol bilan eslab bering.
16. C++ tilida ekran vafaylga IO dantashqari kiritish va chiqarish vazifalarini lokalizatsiyalash sifatida qanday vositadan foydalanadigan qator funksiyalarni qo'llab-quvvatlanadi.
17. Kiritish/chiqarish ob'ektlari sifatida massivdan foydalanish uchun qanday header fayldan foydalanish lozim.
18. Massiv gama'lumotlarni yozish va chiqish uchun qanday tipdan foydalaniladi.

19. Massivlarasosida eof() va get() funksiyalarning IO dafoydalanishnimauchunzarur.
20. Kirish/Chiqishidentifikatornio‘rnatishvao‘chirishusullariaytibberingvamisollarorqaliasoslang.
21. Formatlashidentifikatorlarinimauchunkerakvauqandayishlatiladi.
22. Sanoqsystemagaasoslanganindikatornichiqarishuchunqaysiformatlashidentifikatoridanfoydalanamiz.
23. Chaptomondantekislabformatlashdaqaysiformatlashidentifikatoridanfoydalanamiz.
24. Monipulyatorlarbilanformatlashningusullarinimauchunkerakvauqandayishlatiladi.
25. Sanoqsystemagaasoslanganindikatornichiqarishuchunqaysimonipulyatorlarbilanformatlashningusuldanfoydalanamiz.



## AMALIY KO‘NIKMA VA MALAKALARNI ANIQLASH HAMDA RIVOJLANTIRISH UCHUN ASSISMENT TOPSHIRIQLARI.

<b>BIRINCHI ASSISMENT TOPSHIRIG‘I</b>	
	<p>Tipnianiqlashgaoidberilganquyidagidasturbo‘yichaberilgantopshiriqlarkeraklifragmentlariasosidabajaring.</p> <p> Bundadasturdagiba‘zio‘zgartirishlarnitopishorqalitopshiriqlarbosqichma – bosqichamalganoshiriladi.</p>
dastur	topshiriqlar
<pre>#include &lt;iostream&gt; #include &lt;typeinfo&gt; #include &lt;cstdlib&gt;  using namespace std;</pre>	<p>1. Figure <span style="float: right;">sindagixatoliknianiqlang.</span></p> <p>Dasturdatuzatishkiriting.</p> <hr style="border: 0; border-top: 1px solid black; margin: 5px 0;"/> <hr style="border: 0; border-top: 1px solid black; margin: 5px 0;"/> <hr style="border: 0; border-top: 1px solid black; margin: 5px 0;"/> <hr style="border: 0; border-top: 1px solid black; margin: 5px 0;"/> <hr style="border: 0; border-top: 1px solid black; margin: 5px 0;"/> <hr style="border: 0; border-top: 1px solid black; margin: 5px 0;"/> <hr style="border: 0; border-top: 1px solid black; margin: 5px 0;"/> <hr style="border: 0; border-top: 1px solid black; margin: 5px 0;"/>



```

    }
};

class
rectangle :
public
figure {
public:
    rectang
le(double
i, double
j) : figure
(i, j) {}
    double
area() {
return x *
y;}
};

class
circle :
public
figure {
public:
    circle(
double i,
double j=0)
: figure(i,
j) {}
    double
area()
{return
3.14 * x *
x;}
};

// figure
sinf

```

5. dasturdatipnialmashtirishishlatilganmiyokitiplarn itaqqoslashmi?

Vabuamaldasturningqaysifragmentlaridao‘zaksinitopgan.

---



---



---



---

ob'ektlarin  
i yaratish.

```
figure
*factory(){
    switch(
rand() % 3
) {
    case 0:
return new
circle
(rand()%15)
;
    case 1:
return new
triangle
(rand()%15,
rand()%15);
    case 2:
return new
rectangle
(rand()%15,
rand()%15);
    }
return 0;
};
```

```
int main(){
    figure
*p; // asos
sinfga
ko'rsatkich
    int i;
    int
t=0, r=0,
c=0;
// 15 ta
ob'ektlarni
```

```

hosil qilsh
va sanash
    for(i=0
; i<15;
i++) {
        p =
factory();
// ob'ektni
hosil
qilish

        cout <<
"Ob'ektning
tipi: " <<
typeid(*p).
name();

        cout <<
". ";
        //
ob'ektni
tekshirish
orqali
sanasj

        if(type
id(*p) ==
typeid(tria
ngle)) t++;

        if(type
id(*p) ==
typeid(rect
angle))
r++;

        if(type
id(*p) ==

```



```

typeid(circle) c++;
    //
    Yuzasini
    hisoblash

    cout <<
    " S= " <<
    p->area()
    << endl;
    }
    cout <<
    endl;
    cout <<
    "Quyidagi
    ob'ektlar
    hosil
    qilindi:\n"
    ;
    cout << "
    Uchburchakl
    ar: " << t
    << endl;
    cout << "
    To'rtburcha
    klar: " <<
    r << endl;
    cout << "
    Doiralalar: "
    << c <<
    endl;
    system("pau
    se");
    return 0;
}

```

6. Dasturdajamibo'lib, nechao'zgartirishkiritildi. \_\_\_\_\_
7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.

<b>IKKINCHI ASSISMENT TOPSHIRIG'I</b>	
🎓	<p>Tipnialmas trishga oid berilgan quyidagi dastur bo'yicha berilgan topshiriqlar kerakli fragmentlarini o'stiring.</p> <p>👉 Bunday dasturda giba'zi o'zgartirishlarni topish orqali topshiriqlar bosqichma – bosqich amalga oshiriladi.</p>
dastur	topshiriqlar
<pre>#include "stdafx.h" #include &lt;iostream&gt; #include &lt;typeinfo&gt; using namespace std; class Base {     public:     virtual void f() { cout &lt;&lt; " Base sinfi orqali.\n"; } // . . . }; class myClass : public Base {     public:     void f() { cout &lt;&lt; "myClass sinfi orqali.\n"; }</pre>	<p>1. Dynamic_cast tipni almashtirishni ishlatmasligi uchun dasturning qaysi fragmentini o'chirish kerak.</p> <p>_____</p> <p>_____</p> <p>_____</p>
	<p>2. int berilgan o'zgarishni tipni almashtirish dastur fragmentining qaysi qismini o'chirish kerak. Shu fragmentni float tipidan boshqatipga almashtirish fragmentini tuzing.</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p>
	<p>3. static_cast&lt;bool&gt;(x); fragmentining qiymatini 0 chiqaradigan qilib dasturni o'zgartiring.</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p>
	<p>4. Dastur natijasida ekranga 2 marta Xatolik nishonlar chiqarish uchun dasturning kerakli fragmentlarini o'zgartirishlarni yozing.</p> <p>_____</p> <p>_____</p> <p>_____</p>

```

};
void funk
(const int
*p){
    int *v;
    v =
const_cast<i
nt *> (p);
    *v =
1980;
};
int main(){
    Base
*bp, b_ob;
    myClass
*dp, d_ob;

    bp =
dynamic_cast
<Base *>
(&d_ob);

    if(bp) {
        cout
<< " tipni
almashtirish
" <<"
(myClass* =>
Base *)
bajarildi.
";

        bp->f();
    } else
cout
<<"Xatolik\n
";
    cout <<

```

5. Dasturda reinterpret\_cast<int>ning natijasi 65  
chiqishi uchun qaysi fragmentni o'zgartirish kerak.

---



---



---

```



endl;
    dp =
dynamic_cast
<myClass *>
(&b_ob);
    if(dp)
cout
<<"Xatolik\n
";
    else
    cout <<"
tipni
almashtirish
"<<"(Base*=>
myClass*)
bajarilmadi.
\n";
    cout <<
endl;
    int x =
2004;
    cout <<"
x = " << x
<< endl <<
endl;
    funk(&x)
;
    cout <<"
x = " << x
<< endl;
    cout <<
endl;
    double
d;
    d =
static_cast<
bool>(x);
    cout <<"

```

```
x = " << dec
<< d <<
endl;
    cout <<
endl;
    char *p
= "Salom
Buxoro";
    x =
reinterpret_
cast<int>
(p);
    cout <<"
x = " << x
<< endl;
system("paus
e");
return 0;
}
```

6. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. \_\_\_\_\_
7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.

## UCHINCHI ASSISMENT TOPSHIRIG'I

 Nomlar fazosidan foydalanishga oid berilgan quyidagi dastur bo'yicha erilgan topshiriqlar kerakli fragmentlari asosida bajarilgan.
   
 Bundagi dasturda 'zio' zgartirishlari topish orqali topshiriqlar bosqichma – bosqich amalga oshiriladi.

dastur	topshiriqlar
<pre> #include "stdafx.h" #include &lt;iostream &gt; using namespace std;  namespace NS1{     class demo{         int i; public:     demo( int x) { i=x; }         void set(int x) { i=x; }         int get() { return i; } </pre>	<p>1. Dasturdagi std fazosining elementlari using namespace std; fragmentidan foydalanmasdan qanday o'zgarishlari kiritilishi aniqlang.</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>2. NS1 fazodagi str ko'rsatkichining qiymatini o'zgartirib chiqarish fragmentini yozing.</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>3. Dasturda yaratilgan fazolardan yopiq holda foydalanish uchun barcha fragmentlarni o'zgartiring.</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>4. Dasturda _____ 2 ga ajratilgan fazo yaratishni tekshirib ko'ring.</p> <p>_____</p> <p>_____</p> <p>_____</p>

<pre> };     char str[] = "Nomlar fazolarin i oʻrnatish \n";     int counter; }  namespace NS2 { int x, u; }  int main(){     NS1:: demo ob(10);     cout &lt;&lt;" ob = " &lt;&lt; ob.get() &lt;&lt; endl;     ob.se t(99);     cout &lt;&lt; "Endi ob = "&lt;&lt; ob.get() &lt;&lt; endl;      using NS1::str; cout &lt;&lt; str; </pre>	<p>5. Dasturdagi takrorlanish jarayoni kamayishta tartibdastaralanib chiqishi uchun qanday oʻzgartirish kiritish kerak.</p> <hr/> <hr/> <hr/> <hr/>
--	---

```
    using
namespace
NS1;
    for(c
ounter =
1;
counter<=
10;
counter++
)

    cout
<<
counter
<< ends;
    cout
<< endl;

    NS2::
x = 10;
NS2::u =
20;
    cout
<<
"o'zgaruv
chilar x,
y = "<<
NS2::x<<"
, " <<
NS2::u<<
endl;

    using
namespace
NS2;

demo
```



```
x_ob(x),  
y_ob(u);  
  
cout <<  
"x_ob,  
y_ob  
ob'ektlar  
ning  
qiymati:  
"<<  
x_ob.get(  
); cout<<  
" " <<  
y_ob.get(  
) <<  
endl;  
system("p  
ause");  
return 0;  
}
```

6. Dasturdajamibo'lib, nechao'zgartirishkiritildi. \_\_\_\_\_
7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.

## TO‘RTINCHI ASSISMENT TOPSHIRIG‘I

	<p>Buferlashtirilgankirishvachiqarishdanfoydalanishgaoidberilganquy idagidasturbo‘yichaberilgantopshiriqlarkeraklifragmentlariasosida bajaring.</p> <p>👉 Bundaydasturdagiba‘zio‘zgartirishlarnitopishorqalitopshiriqlarbos qichma – bosqichamalganoshiriladi.</p>
dastur	topshiriqlar
<pre>#include "stdafx.h"  #include &lt;iostream&gt; using namespace std; int main(){ char buf[] = "SalomBuxor o 1980 28.01 B"; istrstream istr(buf); char c;     while(! istr.eof()) {     istr.ge t(c);     if (!istr.eof( )) cout &lt;&lt; c;     }     cout&lt;&lt;e ndl;</pre>	<p>1. Dasturdagibufferlashtirilgankirish/chiqishoperatorl arningishlamasliginianiqlangvaxatosinituzatib, keraklijoygafragmentniyozing.</p> <p>_____</p> <p>_____</p>
	<p>2. Birinchiekrangachiqarishoperatorininimaniekrang achiqaradivabirinchi buferlashtirilganchiqarishmassivg animaniyozaadi.</p> <p>_____</p> <p>_____</p> <p>_____</p>
	<p>3. Dasturda <code>iostr.getline(str, 79);</code> fragmentningvazifasinianiqlang.</p> <p>_____</p> <p>_____</p>
	<p>4. Dasturdamatnitekslashvasonlarniishorasibilanchiq arishfragmentiniajratingvatekshiribko‘ring.</p> <p>_____</p> <p>_____</p> <p>_____</p>
	<p>5. Dasturdagitakrorlanishjarayoniqandayvazifanibaj aradi.</p> <p>_____</p> <p>_____</p> <p>_____</p>

```

char
str[80];
float f;
char ch;
int i;
istr >> str
>> i >> f
>> ch;
cout << str
<< ' ' << i
<< ' ' << f
<< ' ' <<
ch << '\n';


stringstream
iostr
(iobuf,
sizeof
iobuf) ;

    iostr
<< 100 <<
hex << ' '
<< 100 <<
ends;
    char
str[80];
int j ;
    iostr.g
etline(str,
79);
    iostr
>> dec >>
j;
    cout <<
str << ' '
<< j << '
';

```

<pre>         iostr     &gt;&gt; hex &gt;&gt;     j;         cout &lt;&lt; hex &lt;&lt; j &lt;&lt; endl;  cout.setf(i os::left); cout.width( 28); cout &lt;&lt; «mbbGroup» &lt;&lt; endl; cout.setf(i os::showpos ); int value = 15; cout &lt;&lt; value &lt;&lt; endl; system("pau se"); return 0; } </pre>	
<p>6. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____</p> <p>7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.</p>	

## 1.2. Konteynerlar (Kollektsiyalar).

 STL (Standard Template Library) kutubxonalaribilantanihib, uningto‘plamlarivaharto‘plamningvazifasi, ishlatishmaqsadi, maxsususullarivaalgoritmlarini, iteratorvadinamikstrukturaxususiyatlarini, iteratorlarningturlarini, iteratorlarningusullaridanfoydalanishni, barchato‘plamlardamavjudbo‘lganasosiyusullarni, STL kutubxonaalgoritmlarini, barchato‘plamelementlaritanlashvaulargaishlovberishusullari,

saralashusullari,  
 to'plama'zolariustidama'lumarifmetikamallarnibajarishusullari,  
 pridekatlar, oqimxavfsizligi, konteynersinflar, ketma-ketkonteynerlar,  
 assotsiativkonteynerlar, konteynerlarningxususiyatlari,  
 konteynersinflarningumumiyusullari, chiziqlikonreynerlar(array, vector,  
 deque, list, forward\_list), konteynerlarbilashishlashfunksiyalari,  
 allocator ajratuvchiningvazifasi, biraloqaliro'yxat<forward\_list>,  
 ikkialoqaliro'yxat (ikkilanganro'yxat) <list>,  
 ikkitomonlamanavbat<deque>, dinamikmassiv (vektor) <vector>,  
 statikmassiv<  
 array>kabito'plamlarningfazifalarivausullarikeltirilganbo'lib,  
 nazariybilimlarniasoslashuchundasturlartuzibko'rsatilgan.

Bilimlarnimustahkamlashuchun 20  
 tanazariysavolvaamaliyko'nikmavamalakalarnirivojlantrishuchun 3  
 taassimenttopshirig'ivaharassimentda 7 tatopshiriq, jami  
 21tatopshiriqberilgan.

Butopshiriqlarnibajarishmavzunimustahkamlashuchunxizmatqiladi.

***✍Kalitso'zlar.*** STL kutubxonasi, to'plam, shablon, vector, list, map, set, multimap, multiset, string, wstring, stringstream, satriqim, iterator, dinamikma'lumotlartuzilmasi, pridekat, konteynersinflar, ketma-ketkonteynerlar, chiziqlikonreynerlar(array, vector, deque, list, forward\_list), biraloqaliro'yxat<forward\_list>, ikkialoqaliro'yxat (ikkilanganro'yxat) <list>, ikkitomonlamanavbat<deque>, dinamikmassiv (vektor) <vector>, statikmassiv< array>, allocator.

***✔Bilishshartbo'lgantushunchalar.*** Tiptushunchasi, sinfvasinfob'ekti, to'plam, massiv, elementgamurojaat, funksiyavako'rsatkich, ma'lumotlarnikirishvachiqishi, oqim, dasturlashgaoiddastlabkitushunchalarvaC++tiliniqo'llabquvvatlovchimu hitdaishlashnibilishlozim.

***☞Bilibolasiz.*** STL kutubxonasiningimkoniyati, vector, list, map, set, multimap, multiset, string, wstring, stringstream, satriqimkabito'plamlarniishlatishusullarivavazifalar, iteratorvadinamikma'lumotlartuzilmasiishlatishvaturlari, pridekat, konteynersinflar, ketma-ketkonteynerlar, chiziqlikonteynerlar(array, vector, deque, list, forward\_list) xususiyatlarnivafunksiyalariniishlashtamoyllari, biraloqaliro'yxat<forward\_list>, ikkialoqaliro'yxat (ikkilanganro'yxat) <list>, ikkitomonlamanavbat<deque>, dinamikmassiv (vektor)

<vector>, statik massiv < array> kabikonteynerlarni dasturlashdagi o'rniva allocator orqali yaratish va foydalanish usullarini o'rganishingiz mumkin.

## REJA

1. STL kutubxonalari.
2. Konteynersinfi.
3. Chiziqli konteynerlar (array, vector, deque, list, forward\_list).

## KIRISH

Dasturlash texnologiyalarining rivojlanishi dasturlashtillari ishlabchi qaruvchilarni jadalga jiddiy masallarni paydo qilmoqda.

Shulardan bir budasturlashtillari uchun turlixi to'plamlar bilan ishlashdir.

Masalan, xayotdan olib qaraydigan bo'lsak, ammo qonuniyat harxil. barchanarsalar qandaydir to'plam, ammo qonuniyat harxil.

Bu to'plamlarni qanday dasturlash kerak degan muammoga paydo bo'ladi.

Shuning uchun barcha ob'ektga o'zgartirilgan dasturlashtillarida STL, ya'ni standart shablonlar kutubxonasi tushunchasi kiritilgan. STL (Standard Template Library) kutubxonasi, to'plamlari, xususiyatlari, usullari va funksiyalar,

konteyner va iteratorlar bo'yicha nazariy materiallarni keltiramiz.

**STL (Standard Template Library) kutubxonasi.** Shablon mexanizmlari C++

kompilyatoriga moslab qurilgan bo'lib, dasturchilarga umumiy dasturlash yordamida dastur fragmentlarini qisqartirishga imkon beradi. Tabiiyki,

bunday mexanizmlarni amalga oshiruvchi standart kutubxonalar ham mavjud. Bugungi kunda C++ dasturlashtilida eng samarali STL kutubxonasi hisoblanadi.

STL kutubxonasi ning ko'plab tatbiqlari mavjud bo'lib, ularning har birini qstandart doirasida yaratilgan bo'lsa-da,

o'z kengaytmalariga ega. Ammo bunday yondashuvning birkamchiligi bor: dastur fragmentini hardoi turlik kompilyatorlar bilan bir xil tarzda ishlaydi.

Shuning uchun, dasturchi qanchalik mohirlik bilan kutubxonani yaratish va foydalansa,

o'ziga xos bajarilishini tushunsa ham, imkon qadar anaviy usullardan foydalanishni tavsiya qilamiz.

C++ dasturlashtilining kutubxonalaridan eng mashhur to'plamlarni ko'rib chiqaylik.

Ular har bir muhim vazifalarni hal qilinishi mumkin doirasi uchun o'z shablon parametrlariga ega.

To'plamlar ni dastur fragmentida ishlatish uchun quyidagi fragmentdan foydalaniladi.

```
#include <T>
```

Bunda T – to'plamning nomi.

Odatda quyidagi to'plamlar ko'p ishlatiladi.

**vector** – elementlar to'plami, o'lchamini o'zgartirish kerak bo'lgan massivdasaqlanadigan elementlar to'plami (odatda ortib boradigan); Dastur gaulanish uslubi:

```
#include <vector>
```

**list** – elementlar to'plami, elementlarni ikki tomonlama bog'langan ro'yxat sifatida saqlaydigan to'plam; Dastur gaulanish uslubi:

```
#include <list>
```

**map** – elementlar to'plami, har bir elementning shakli <const kalit, T> juftlikdasaqlanadigan to'plam, bu oddiy bir juftlik <kalit, qiymati> juftligi (har bir kalit bitta qiymatni mos keladi). Kalit – taqqoslash amali uchun qiymatini tavsiflovchi ma'lum bir qiymati. Juftlikdagi kalit tez qidirishni amalga oshirish imkonini beradi, kalitni tartiblashtirilgan shaklida saqlanadi. Tartiblashni amalga oshirish uchun oldindan tashkil qilish qonuniyatini aniqlash kerak. Dastur gaulanish uslubi:

```
#include <map>
```

**set** – elementlar to'plami, faqat kalitlarning qiymatini bo'yicha tartiblangan to'plamdir, ya'ni taqqoslash amali qo'llaniladigan, ammo takrorlanmaydigan qiymatlar — har bir kalit to'plamda (inglizcha set-to'plam degan ma'noni beradi) faqat birmarta foydalaniladi; Dastur gaulanish uslubi:

```
#include <set>
```

**multimap** – map, juftlikdagi kalitlar unikal emas, takrorlanadigan to'plam. Agarkalit bo'yicha qidirsangiz, siz bitta qiymatni emas, balki bir xil kalit qiymatiga ega bo'lgan elementlar to'plamini olasiz. Dastur gaulanish uslubi:

```
#include <map>
```

**multiset** – set, juftlikdagi kalitlar unikal emas, takrorlanadigan to'plam. Agarkalit bo'yicha qidirsangiz, siz bitta qiymatni emas,

balkibirxilkalitqiymatigaegabo‘lganelementlarto‘plaminiolasiz.

Dasturgaulanishuslubu:

```
#include <set>
```

Belgilarto‘plamibo‘lgansatr (qator)nihamto‘plamsifatidaqarashmumkin. Shuninguchunixtiyoriy kutubxonasatrlar bilanishlashvaifodalashuchuno‘z iningsinflargaega. STLdasatrlar ASCII va Unicode formatlaridaifodalanadi.

**string** — to‘plam, ASCII formatidagibirbaytlibelgilarto‘plami;

**wstring** — to‘plam, Unicode formatidagiikkibaytlibelgilarto‘plami;

Dasturgaulanishuslubu:

```
#include <string>
```

```
#include <wstring>
```

Shuningdek, belgilarto‘plamibo‘lgan, satrliqimlarnihamto‘plamsifatidaqarashmumkin. **stringstream** oddiyma’lumotlartiplarini STL-string ko‘rinishadasaqlashnitashkilqilishuchunishlatiladi.

Dasturgaulanishuslubu:

```
#include <stringstream>
```

Dasturlarnitahlilqilishnishopinfandanboshlaymiz.

2.1-dastur. Satrliqimto‘plamsifatidaishlatish

```
#include "stdafx.h"
#include <iostream>
#include <stringstream>
#include <string>
using namespace std;

int main(){
    stringstream xstr;
    for (int i = 0; i < 10; i++)
    {
        xstr << "Demo " << i << endl;
    }
    cout << xstr.str() << endl;
    string str;
    str.assign(xstr.str (), xstr.pcount());
    cout << str.c_str() << endl;
```



```

system("pause");
return 0;
}

```

```

2.1-dasturnatijasi. Output
Demo 0
Demo 1
Demo 2
Demo 3
Demo 4
Demo 5
Demo 6
Demo 7
Demo 8
Demo 9
=====xxxxxlllllllllyu■yu■
Demo 0
Demo 1
Demo 2
Demo 3
Demo 4
Demo 5
Demo 6
Demo 7
Demo 8
Demo 9

```

Satrlinqim - oqimoxiridabir null terminatorbilantugaydiganbuferdir. Shuninguchununingbo'shqolganelemetlariixtiyoriybelgibilanto'ldiriladi. Satrlinqimninghaqiqiyqisminiolishuchun pcount() elementlarnisanagichidanfoydalanishmumkin. Keyinesa, satrlinqimning "haqiqiyqismi" olinadivachopqilinadi.

Iteratorlarhamdinamikstrukturagaegato'plamdir. Iteratordinamikma'lumotlartuzilmalariniamalgaoshirishdajudamuhimtus hunchadir. Tushunishimizuchun, iteratornima'lumbircheklovlarbilanbirko'rsatkichsifatidaabstraktko'rinishdaaniqlashmumkin. Sirasiniaytganda, iteratorumumiytushunchabo'libvabirko'rsatkichuchunob'ektto'plamibo'l

adi, ammodo'rsatkichbuitatoremas.  
Iteratorsinfiniquyidagichaquribolishmumkin:  
2.2-dastur. Iteratorninginfiniqurish.

```
class Iterator{
    T* pointer;
public:
    T* GetPointer (){
        return this -> pointer;
    }
    void SetPointer (T* pointer){
        this -> pointer = pointer;
    }
};
```

Iteratorningba'zibirformallashtirilganta'riflari:  
Iteratorlarto'plamelementlarigakirishnita'minlaydi. Harbiraniq  
STL sinfiuchuniteratorlarto'plamdasinfinichidaalohidaaniqlanadi.

Iteratorlarninguchturimavjud:

- (forward) iterator - to'plamnikiichikindeksdankattaroqindeksgaotkazishuchun;
- reverse iterator - to'plamnikattaindeksdankichikroqindeksliotkazishuchun;
- random access iterator - to'plamdantasofidiy, harqandayyo'nalishdatanlashuchun.

To'plamningyarimelementlarinio'chirishgadoirmisolkeltiramiz.

2.3-dastur. To'plamningyarimelementlarinio'chirish.

```
#include "stdafx.h"
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

void printInt (int number);

int main (){

    vector<int> myVec;
    vector<int>::iterator first, last;
    for (long i=0; i<10; i++){
```

```

        myVec.push_back(i);
    }
    first = myVec.begin();
    last = myVec.begin() + 5;
    if (last >= myVec.end()){
        return - 1;
    }
    myVec.erase(first, last);
    for_each (myVec.begin(), myVec.end(),
printInt);
    system("pause");
    return 0;
}
void printInt (int number)
{
cout << number << endl;
}

```

### 2.3-dasturnatijasi. Output

```

5
6
7
8
9

```

2.3-dasturdagiba ‘zifunksiyalarnikeyinroqtushuntiribo‘ tamiz.

Shunihamtushunishkerakki,

to‘planningqandaydirelementiuchuniteratorolishdato‘plamniketma-  
keto‘zgartirishiteratorniyarozsizholgakeltirishmumkin.

Iteratorningiteratsiyasidaoldingavashungao‘xshashorqagao‘tishquy  
idagichafragmentiasosidasodirbo‘ladi:

```

for (iterator element = begin(); element < end();
element++)
{ t = (*element); }
for (iterator element = end(); element < begin();
element--)
{ t = (*element); }

```

Tasodifiytanlashiteratoridanfoydalanganda,

masalan,

quyidagichafragmentniyozishmumkin:

```
for (iterator element = begin(); element < end();
element+=2)
{ t = (*element); }
```

Barcha to'plamlardamavjud bo'lgan asosiy usullar quyidagi 2.1-jadvalga keltiramiz.

2.1-jadval. To'plam usullari.

No	nomi	vazifasi
1	empty	To'plamni bo'shligini tekshiradi
2	size	To'plamning o'lchamini qaytaradi
3	begin	To'plamning birinchi elementini ko'rsatadi
4	end	To'plamning oxirgi elementini ko'rsatadi. Elementi yo'q to'plamga oxirgisidan keyingacha
5	rbegin	Teskari to'plamning birinchi elementini ko'rsatadi
6	rend	Teskari to'plamning oxirgi elementini ko'rsatadi.
7	clear	To'plamni tozalash, barcha elementlarini o'chiradi.
8	erase	Ajratilgan elementlarni to'plamdan o'chiradi.
9	capacity	To'plamning sig'imini qaytaradi, ya'ni butun to'plam uchun mumkin bo'lgan elementlar soni (aslida to'plam uchun qancha xotira ajratilganini qaytaradi)

To'plamning sig'imi (hajmi), boshida aytib o'tilganidek, kerak bo'lganda o'zgaradi, ya'ni, agar to'plam uchun ajratilgan barcha xotira lart o'lgan bo'lsa, unda ang element qo'shilganda, to'plamning sig'imi oshiriladi va o'sishdan oldin unda g barcha qiymatlaryan gixotiramaydonigako'chiriladi - budasturchilar uchun juda qimmat amal bo'lib hisoblanadi. Dasturlashda to'plamning hajmi va quvvati ishonch hosil qilish muhim dasturlar ashelement hisoblanadi. To'plamning sig'imi turlicha ekanligiga ishonch hosil qilish uchun quyidagidasturni keltiramiz.

2.4-dastur. To'plamning sig'imini tekshirish.

```
#include "stdafx.h"
#include <iostream>
#include <vector>

using namespace std;
int main(){
```

```

vector<int> vec;
cout << "Real size of array in vector: " <<
vec.capacity() << endl;
for (int j = 0; j < 25; j++){
    vec.push_back (10);
}
cout << "Real size of array in vector: " <<
vec.capacity() << endl;
system("pause");
return 0;
}

```

### 2.3-dasturnatijasi. Output

```

Real size of array in vector: 0
Real size of array in vector: 28

```

**vector** – engko‘pishlatiladiganto‘plamvektorhisoblanadi. Buto‘plamning operator[] operatorigaegaekanligijudaqulay. Odatiy massivkabi ishlatiladi. Xuddishuningdek, bu operator map, deque, string i wstring to‘plamlarigahammavjud.

Vektorning quvvatidynamikravishdao‘zgarishinitushunish muhimdir. Odatda, multiplikativyondashuvhajminioshirishuchunishlatiladi: zarurbo‘lsa, vektoruchunajratilganxotiraningmartasoniortadi, ya‘ni, yangielementqo‘shibto‘plamsig‘iminingoshirishgasababbo‘lsa, operatsiontizimidasturiuchunyangixotiramaydoniajratadi. Masalan, ikkibarobarkattabo‘lishuchun, eskixotiramaydonibarchaelementlarinusxasiyangiqiyamatqilibqo‘shish.

### **STL kutubxonaalgoritmlari.** STL

kutubxonasini ishlabchiquvchilarshablonlima‘lumotlartuzilmalarimajmua sigaeagakutubxonayaratishdao‘zlarigaanchajiddiyvazifaqo‘yishgan. STL kutubxonasito‘plamlaribilanishlashimkoniniberadigan, mashhuralgoritmlarnioptimaltatbiqlarivakattamajmuinio‘zichigaoladi. Barchaamalgaoshirilganfunksiyalarniuchguruhgabo‘lishmumkin:

1. Barchato‘plamelementlaritanlashvaulargaishlovberishusullari :

```

sount, count_if, find, find_if, adjacent_find, for_ea
ch, mismatch, equal, search,copy, copy_backward, swap
, iter_swap,
swap_ranges, fill, fill_n, generate, generate_n, repl

```

```
ace,  
replace_if, transform, remove, remove_if, remove_copy  
,  
remove_copy_if, unique, unique_copy, reverse, reverse  
_copy, rotate, rotate_copy, random_shuffle, partition  
,  
stable_partition
```

Bu usularni vazifalarini ixtiyoriy manbadan, [splusplus.com](http://splusplus.com)  
vebsahifasidan foydalanib bilish mumkin.

## 2. Saralash usullari:

```
sort, stable_sort, partial_sort, partial_sort_copy,  
nth_element, binary_search, lower_bound, upper_bound,  
  
equal_range, merge, inplace_merge, includes, set_unio  
n,  
set_intersection, set_difference,  
set_symmetric_difference, make_heap, push_heap, pop_h  
eap, 'sort_heap, min, max, min_element, max_element,  
lexographical_compare, next_permutation,  
prev_permutation
```

Bu usularni vazifalarini ixtiyoriy manbadan, [splusplus.com](http://splusplus.com)  
vebsahifasidan foydalanib bilish mumkin.

## 3. To'plama' zolari ustidama' lumarifmetik amallarni bajarish usullari:

```
Accumulate, inner_product, partial_sum,  
adjacent_difference
```

Bu usularni vazifalarini ixtiyoriy manbadan, [splusplus.com](http://splusplus.com)  
vebsahifasidan foydalanib bilish mumkin.

Ushbu usularni sanabo'tishdan maqsad STL  
kutubxonasi tomonidan taqdim etilgan boy vositalar to'plamini bilantirish  
hdir. Qo'shimchama'lumot olish uchun,  
C++ dasturlashtirish tiliga oidsinflarning tegishli hujjatlaridan olish mumkin.

**Predikatlar.** Ko'pgina STL  
kutubxona algoritmlari uchun algoritmlar to'plamining muayyan zosibilanni  
maqilish kerakligini aniqlaydigan shartni o'rnatishning izlozim bo'ladi.  
Predikat - bu funksiya, bir nechta parametrlarni oladi va Boolean  
qiymatini qaytaradi (true/false). Standart predikatlar to'plamida mavjud.

**Oqim xavfsizligi.** Bu STL  
butunlay xavfsiz kutubxona emasligini tushunish muhim ahamiyatga ega.

Lekin buni ammonihal qilish juda oddiy:  
ikkioqimlar bir xil to'plamdan foydalanayotgan bo'lsa, Mutex  
seksiyasini amalga oshirish zarur.

STL cross-platform kutubxonasi hisoblanadi. Albatta, ushbu kutubxonakompilyatorning har qanday versiyasi uchun mavjudligiga mutlaq kafolat yo'q. Masalan, ukamdan- kam hollarda mobil qurilmalarda amalga oshiriladi, chunki amalga oshirilgan ma'lumotlartuzilmalarining aksariyati xotira niteja masdan, tezlik foydasini tanlaydi hamda xotira mobil platformalarda eng qimmatli texnik resursdir, kompyuterda esa hozir juda ko'p. Shuning uchun tez-tez STL kutubxonangizni lokalizatsiyasini yaratish kerak bo'ladi, masalan, ilova sinimobil platformaga ko'chirish uchun.

**Konteynersinflar.** Konteynersinflar muayyan tarzda tashkil qilingan ma'lumotlarni saqlash uchun mo'ljallangan sinflar.

Turli xil tipdagi ma'lumotlarni saqlash uchun bir xil turdagi konteynerdan foydalanishingiz mumkin.

Buxususiyat sinflar shablonlari yordamida amalga oshiriladi, shuning uchun C++ kutubxonasi konteynersinflarini, shuningdek algoritmlarni va iteratorlarni o'z ichiga olgan qismiy standart shablonlarni kutubxonasi (STL) deb ataladi.

Ma'lumotlarni konteynerlarda saqlanadigan vaular bilan turli amallarni konteyner usullari vosilasidan va algoritmlar bilan aniqlanadigan va bajariladi.

Iteratorlar bu ikki elementni bir-biriga bog'lagan holda ishlaydi.

Ulartu foydali har qanday algoritm har qanday konteyner bilan ishlashim mumkin.

Professional dasturlashni kutubxonasi sinflarisiz foydalanishni tasavvur qilishim mumkin emas, shuningdek alohida konteynerlarsiz ham.

Ulardan foydalanish dasturlarning ishonchliligi, joriy qilish samaradorligi, moslashuvchanligi va ko'pchilik irraliligini oshirish hamda dastur tuzish vaqtinika maytirish gaimkonini beradi.

Kutubxonani yaratish ko'pchilik va mashaqqatli qilib, amao, dastur yaratish vaqtida o'zini oqlaydi.

STL

kutubxonasi dasturlarni yozishda ishlatiladigan asosiy ma'lumotlartuzilmalarini amalga oshiruvchi konteynerlarni o'z ichiga oladi: vektorlar, navbatlar, ro'yxatlar, lug'atlar va to'plamlar.

Konteynerlarni ikki turiga bo'lishim mumkin: ketma- ket va assotsiativ konteynerlar .

**Ketma-ketkonteynerlar.** Ularuzluksizketma-ketlikdao‘xshashmiqdorlarningcheklisoninisaqlashnita‘minlaydi. Konteynerlarsifatidavektor (vector), ikkitomonlamanavbat (deque), ro‘yxat (list) vabiralaloqaliro‘yxati (forward\_list), shuningdekkonteynervariantlarasosidaadapterlar, stek (stack), navbat (queue) vaustuvorlikbilannavbat (priority\_queue) sinflarini‘zichigaoladi.

Massivhamamallarbilancheklanganholdakonteynerningyanabirturidir.

Konteynerningharbirturimalumotlarustidao‘zamallarto‘plaminita‘minlaydi.

Siztanlagankonteynerturidasturdagimalumotlarbilannimaqilishnixohlashingizgabog‘liq. Masalan, agarketma-ketliko‘rtasidamalumotlartez-tezjoylashtirishvao‘chirishkerakbo‘lsa, ro‘yxatlardanfoydalanishkerak,ma‘lumotlarnioxiridayokiboshida, birinchnavbatdajoylatrishkerakbo‘lsa, ikkitomonlamanavbatdanfoydalanishmaqsadgamuvofiq.

#### **Assotsiativkonteynerlar.**

Assotsiativkonteynerlarasosiyimalumotlargaakalitlarasosidatezkormuroja atqilishnita‘minlaydi. Bukonteynerlarmuvozanatlidaraxtlargaasoslangan. Assotsiativkonteynerlarningbeshturimavjud: lug‘atlar (map), ko‘plug‘atlar (multi) (multimap), to‘plamlar (set), multito‘plamlar (multiset) vabitlito‘plam (bitset).

Dasturchistandartkutubxonadamavjudbo‘lgansinflargaasoslanibo‘zkonteynersinflariniyaratishimumkin.

Bungakirishishdanoldinmuhimtushunchanibilishinggizshart, ya‘ni STLkutubxonasiiningfundamentaltushunchasibushablondir

Barchakonteynersinflaristandartlashtirilganinterfeysibilanta‘minlangan.

Turlikonteynerlaruchunbirxilamallarningmanosibirxilbo‘lishitabiiydir. Asosiyamallarkonteynerlarningbarchaturlariuchunqo‘llaniladi.

Standartesafaqatkonteynerinterfeysinibelgilaydi, shuninguchunturlixildasturlarsamaradorlikdakattafarqqilishimumkin.

Barchakonteynerlaro‘zxotirasinio‘ziboshqaradi, shuninguchundasturchibuhaqidao‘ylashninghojatiyo‘q.

Deyarliharqandaykonteynerlarningquyidagixususiyatlaribor:

2.2-jadval. Konteynerxususiyatlari.

№	Xususiyatnomi	mazmuni
1	value_type	Konteynerelementiningtipi
2	size_type	Elementlarindeksiningtipi



3	<code>iterator</code>	Iterator
4	<code>const_iterator</code>	O'zgarmasiterator
5	<code>reverse_iterator</code>	Teskariiterator
6	<code>const_reverse_iterator</code>	O'zgarma teskariiterator
7	<code>reference</code>	Elementgahavola
8	<code>const_reference</code>	O'zgarma elementgahavola
9	<code>key_type</code>	Kalittipi (assotsiativ konteynerlar uchun)
10	<code>key_compare</code>	Taqqoslash mezonini (assotsiativ konteynerlar uchun)

Iterator birelement uchun ko'rsatkichga ekvivalentdir. Ularni konteynerlarni to'g'ri yoki teskari yo'nalishda ko'rish uchun ishlatiladi. Iterator dani talab qilinadigan barcha amallarni konteyner elementiga murojaat qilish va uning keyingi elementiga o'tishni amalga oshirishdir. Konteynerlarning elementlarining qiymatlarini o'zgaritish va o'zgaritishni amalga oshirish uchun foydalaniladi.

Iteratorlarning yordamida haqiqiy ma'lumotlarni yaratish uchun o'ylamasdan, konteyner elementlarga murojaat qilish uchun foydalanish mumkin. Buning uchun, har biron konteyner quyidagi 2.3-jadvalda keltirilgan bir nechta usullardan foydalanishni tavsiya qilinadi. Har biron konteyner uchun butiylar va usullarni, ularni realizatsiya qilishni amalga oshirishga bog'liq tarzda belgilanadi. Shuningdek, ixtiyoriy konteynerlarda ularning hajmi haqidagi ma'lumotlarni olish uchun usullari mavjud:

`size()` – elementlar soni;  
`max_size()` – konteynerning maksimal o'lchami (1 milliardta element uchun);

`empty()` – mantiqiy usul, konteyner bo'shligini tekshiradi;  
Zaruriyat bo'lganda ustunlik konteynerning maydonlarini va usullarini ketma-ket o'zlashtirib boraveradi. Biz ham zaruriyat tug'ilganda foydalanamiz.

2.3-jadval. Konteynerlarning umumiy usullari.

No	Usullar	Izoh
1	<code>iterator begin()</code> <code>const_iterator begin () const</code>	Birinchi elementni ko'rsatadi
2	<code>iterator end()</code>	Oxirgisidan keyingi elementni ko'rsatadi

	<code>const_iterator end()</code> <code>const</code>	
3	<code>reverse_iterator rbegin()</code> <code>const_reverse_iterator rbegin () const</code>	Teskariketma- ketlikdabirinchielementniko‘rsatadi
4	<code>reverse_iterator rend()</code> <code>const_reverse_iterator rend () const</code>	Teskariketma- ketlikdaoxirgisidankeyingielementniko‘r satadi

**Chiziqlikonreynerlar**(*array*, *vector*, *deque*, *list*, *forward\_list*). Standartkonteynerlarni ikki taguruh gachiziqli va assosiativ konteynerlarga bo‘lishini bilamiz.

O‘z navbatidagi chiziqlik konteynerlarni bog‘langan ro‘yxat (*forward\_list* va *list*) va tasodifiy kirish konteynerlariga (*ekr* konteyner deb ham yuritiladi) (*deque*, *vector* va *array*) bo‘linadi.

Assosiativ konteynerlar quyidagi variantlarning kombinatsiyasida bo‘lgan sakkiz ta konteyner bilan ifodalangan (tegishli standart sinflar nomlarining qavslarichida beriladi): to‘plam (*\*set*) yoki lug‘at (*\*map*), elementlarni takrorlash gaimkon beruvchi (*\*multi\**) yoki ruxsat bermaydigan, tartibga solingansolinmagan (*unordered\**).

Barcha konteynerlar `iterator` va `const_iterator` tiplari o‘z ichiga olish uchun o‘z navbatida o‘qish-yozish va faqat o‘qish iteratorlarini aniqlaydi.

Konteyner tarkibida olgan iteratorlar oralig‘ini `begin()` va `end()` (o‘qish uchun iterator), shuningdek `cbegin()` va `cend()` (o‘qish uchun `const_iterator`) funksiyalari yordamida olish mumkin.

Konteynerlar ikki tomonlama iteratorlar bilan ham teskari o‘tish uchun `rbegin()`, `rend()`, `crbegin()` va `crend()` funksiyalar mavjud.

Barcha konteynerlarni `empty` funksiyasi tomonidan bo‘sh huncha tekshirish mumkin. Qachonki `cont.begin() == cont.end()` bo‘lsa, o‘z garmaskonteyner bo‘sh hisoblanadi.

Konteynerning oxirgi elementini keyingishartli element gahavola qiluvchi iteratorni qaytaradi. Elementlar sonini `size()` funksiyasi yordamida olish mumkin (*forward\_list* bundan mustasno).

Konteyner tarkibidagi funksiyalarni tozalash uchun `clear()` funksiyasidan foydalaniladi (*array* bundan mustasno).

Iteratorlarda `begin()`, `end()`, `cbegin()`, `cend()`, `rbegin()`, `rend()`, `crbegin()` va `crend()` erkin funksiyalarining shablonlarini aniqlash mumkin. Bular bir xil nomli funksiyalarga havola qilinadi, shuningdek, bunday aniqlangan funksiyalar statik massiv va `std::valarray` uchun aniqlandi. `for(:` takrorlanish operatori xuddi shunday shakligatayanaadi. Masalan, `cr` birkonteyner bo'lsin (shuningdek, statik massiv ham bo'lishi mumkin), shunda dasturda `for()` fragmenti quyidagicha ishlatiladi.

```
for (T x : cr) work(x);
```

Semantik jihatdan quyidagi fragmentga teng

```
{
    using std::begin;
    using std::end;
    const auto e = end(cr);
    for (auto p = begin(cr); p != e; ++p)
    { T x = *p; work(x); }
}
```

Buyerda `T` tipikonteyner elementlarining tipigamoskelishishartemasva `auto` asosida konstruktsiya yoki havola litip ham bo'lishi mumkin (eng ko'p ishlatiladigan variantlar `auto&` va `auto&&`).

Chiziqli konteynerlarni (`array` dan boshqqa) belgilangan qiymatlardan `assign()` funksiyasi nicha qirib to'ldirish mumkin (eski qiymalar o'chiriladi) va `resize()` funksiyasi bilan o'lchamlarini (hajmi kamayganda elementlar oxiridan o'chiriladi va hajmi ortganda, yangi elementlar oxiridan qo'shiladi) aniqlash mumkin.

Konteynerning birinchi elementiga to'g'ridan-to'g'ri murojaat (`front()` funksiyasi yordamida) amalga oshiriladi.

Iteratorlari ikki tomonlama bo'lgan barcha konteynerlarning oxirgi elementga murojaat uchun `back()` funksiyasi, shuningdek, teskari yo'naltirilgan iteratorlar `reverse_iterator` va `const_reverse_iterator` funksiyalaridan foydalaniladi.

Tegishli intervallarni `rbegin()`, `rend()` i `crbegin()`, `crend()` funksiyalari yordamida olish mumkin.

Barcha konteynerlarni tenglik va tengsizlik uchun taqqoslash mumkin va ularning mazmunini `swap()` funksiyasi yordamida almashtirish mumkin. Chegaralanmagan assotsiativlardan tashqari barcha konteynerlarni `<`, `<=`, `>` va `>=` operatorlari leksikografik jihatdan taqqoslash mumkin.

Standartkonteynerlardadinamikxotiraniboshqarishuchun Allocator ajratuvchimaxsussinflardanfoydalanadi. Allocator xotiraniboshqarishningminimalbirliginibelgilaydiganvabirqatoryordamc hita‘riflarnitaqdimetadiganelementtipigabog‘liqdir.

Buvazifato‘rtasosiyfunksiyalariyordamidaamalgaoshiriladi: elementlariberilganqatoruchunxotiraajratishdaallocate, xotiranitozalashuchundeallocate, konstruktorqurishuchunconstruct, vadestruktoruchundestroy.

Allokatorboshqatipdagielementlaruchunanalogallokatorolishda rebind metafunksiyasiniamalgaoshirishikerak.

Alloc – elementlarninganiqtiplariuchunallokatorberilganbo‘lsin. U tiplielementlaruchununingallokatoriniolishvariantiquyidagidasturfragme ntidakeltirilgan.

```
using AllocForU = typename Alloc::template rebind<U>::other;
```

C++dastandartkutubxona (<memory>sarlavhafayl bilan) new/new[] va delete/delete[] operatorlariniqo‘llashorqali allocator<T> nita‘minlaydi.

Bundanmodelvao‘zingizningallokatorlaringgizniyozishdafoydalanish mumkin.

**Biraloqaliro‘yxat<forward\_list>**. forward\_list<T, a = allocator<T>>birtomonlamaiteratororqali T tipdagielementlargakirishnita‘minlaydi.

Ro‘yxatningtugunlariniyaratishuchunA::rebindorqaliyaratilganallokator dayfoydalaniladi. Biraloqaliro‘yxatningo‘zigaxosligishundaki, faqatbelgilanganjoriytugundankeyinelementlarniqo‘shishvao‘chirishmu mkin:

insert\_after - elementqo‘shish.

emplace\_after – yangielementyaratish.

Bundabelgilanganparametrlaruchunkonstruktorchaqiriladi.

erase\_after - elemento‘chirish.

Birinchielementdanoldindeganko‘shimchaaniqlanganpozitsiyamavj ud. Bubebefore\_begin va cbefore\_begin (const\_iterator qaytaruvchilaruchunvariant) funksiyalariyordamidabajariladi. Shuningdek, fl.push\_front(item) funksiyasidanfoydalanib, elementlarnioldindanjoylashtirishmumkin (yuqoridagigaekvivalent), masalan, quyidagichadasturfragmenti:

```
fl.insert_after(fl.before_begin(), item)
```

Konstruktorgaharqandayparametrlarni joylashtirish uchun  
fl.emplace\_front(...) funksiyasi ishlatiladi.  
Bunga ekvivalent sifatida quyidagi dastur fragmentini yozish mumkin:

```
fl.emplace_after(fl.before_begin(), ...)
```

pop\_front funksiyasi ro'yxatdan birinchi elementni o'chiradi.

C++

standart kutubxonasi daro'yxat konteynerlar xususiyati uchun samarali bo'lgan amallarni bajarishga faqat iteratorlarda foydalanishga shonchsizligi yuqori darajali amallarni qo'llab-quvvatlash hisoblanadi:

merge - ikki tartiblangan ro'yxatni bir-biriga birlashtiradi, elementlarni o'chirmaydilekin o'ngdancha pro'yxatga o'tiladi

splice\_after -

berilgan ro'yxatni ko'rsatilgan elementdan keyin joylashtiradi.

remove -

berilgan elementni o'chirib tashlaydi.

remove - berilgan predikatasidagi barcha elementlarni o'chiradi.

reverse - elementlarni tartibiga murosul qilib qaytaradi.

unique - barcha ketma-ketlikdagi dublikatlarini o'chiradi.

sort - ro'yxatni joyida tartiblaydi.

Umuman olganda,

bu funksiyalar qandaydir standart algoritmlarga o'xshaydi, lekin juda tez va qulay ishlaydi.

Ularga murojaat qilish uchun umumiy ruqsat olish kerak, masalan, sort() funksiyasi uchun std::sort(from, to). Ammo, tasodifiy kirish iteratorlari kabitalab qilingan ro'yxatlar uchun amal qilmaydi.

***Ikkialoqli ro'yxat (ikkilangan ro'yxat) <list>.*** list<T, allocator<T>> ikki tomonlama iterator orqali tipdagi elementlarga kirishni ta'minlaydi.

Bir aloqli ro'yxatdan farqli o'laroq,

elementlarni belgilangan pozitsiyadan oldin joylashtirildi

(insert, emplace, splice funksiyalari), oxiridan qo'shish va o'chirish

(push\_back, emplace\_back, pop\_back),

va iterator ko'rsatayotgan elementni o'chirish (erase)

funksiyalar bilan amalga oshiriladi. \*\_after funksiyasi mavjud emas.

***Ikkimonlamali manavbat <deque>.*** deque<T, allocator<T>>, maxsus kirish iterator orqali elementlar uchun kirish imkonini beradi.

Faqat ro'yxat kabi,

usamarali qo'shish va har ikki tomonidan ma'lumotlar olib tashlash imkonini beradi.

eradi. Indeks **gakirish** uchun **ikkita** funksiya ishlatiladi:  $[]$   
 operator **ivaat** (indeks). **Birinchi** dan **farqliq** o'laroq,  
 ikkinchi **indeks** nite **kirish** radi va **qiyamat** haq **qiy** bo'lmasa,  
`out_of_range` **istisno** ga **murojaat** qiladi.

Ikki **tomon** **lamanavbat** konteyneri uchun **ro'yxat** **nibirx** **iltarzd** **aharqanda**  
 yholatda **elementlarni** **kiritish** va o'chirish **imkon** **in** **iberadi**.  
 Lekin **ikkita** **tomon** **lamanavbat** **dabu** **amallar** **larkonteyner** **hajmi**,  
 vaqt **chiziq** **libo'lish** **initalab** **qilish** **im** **umkin**. Bundan **tashqari**,  
 oldin **dan** **iterator** **gasaqlanganelementlarni** **tartib** **in** **kiritish** **va** **o'chirish** **nibuz**  
**ish** **im** **umkin** **shuning** **uchun** **xotira** **dasaqlanganelementlarning** **ko'rsatkichlar**  
**ini** **yoddasaqlash** **maqsad** **gamuvofiq**  
 (Agar **iteratorlar** **ro'yxat** **kabisaqlanayotgan** **bo'lsa**,  
**elementlar** **gabo'lgan** **ko'rsatkichlar** **o'chirilmaydi**).

**Dinamik massiv (vektor) <vector>**. **Vektor**  $\langle T, a = \text{allocator} \langle T \rangle \rangle$  **elementlar** **ga** **maxsus** **kirish** **iterator** **orqali** **kirish** **nita'min** **laydi**. **Deque**  
**dan** **farqliq** o'laroq,  
**ukiritish** **va** **boshida** **elementlarni** **o'chirish** **uchun** **ruxsat** **bermaydi**.

**Dinamik massiv** **xotira** **ning** **uzluksiz** **bo'lim** **idaketma-**  
**ketsaqlanganelementlarning** **joylashish** **inika** **folat** **laydi**, **uning** **manzili** **data**  
**funksiyasi** **bilan** **qaytariladi**.

**Elementlarni** **qo'shish** **va** **oldin** **dan** **ajratilgan** **saqlash** **nitugatish** **da** **elementlar**  
**ko'chiriladigan** **joy** **ga** **kattahajm** **dagi** **yangi** **dinamik** **qator** **ajratish** **im** **umkin**.  
**Eski** **saqlabo'chiriladi**, **vabarchasaqlangan** **iteratorlar** **yo'qoladi**,  
**o'chirilgan** **objekt** **lar** **uchun** **ko'rsatkich** **bo'lib** **qoladi**.

**Dinamik massiv** **reserve** **funksiyasi** **yordam** **ida** **yetarli** **hajm** **dasaqlash** **uchun**  
**oldin** **dan** **massivni** **tayyor** **lash** **imkon** **in** **iberadi**  
 (uko'chibo'tishda **elementlarni** **olib** **kelish** **im** **umkin**).

**Saqlash** **hajmi** **capacity** **funksiyasi** **yordam** **ida** **bilib** **olish** **ingiz** **im** **umkin**.  
`shrink_to_fit` **funksiyasi** **haqiqiy** **ishlatiladigan** **hajmi** **ni** **ajratadi** **va** **elementla**  
**rni** **foydalanilmagan** **xotira** **dan** **olib** **tashlaydi**.

**Statik massiv**  $\langle \text{array} \rangle$ .  $\text{array} \langle T, N \rangle$  **iterator** **ning** **tasodifiy** **kirish** **orqali** **elementlar** **ga** **kirish** **imkon** **in** **iberadi** **va** **s**  
**tatik**  $\text{array} \langle T[N] \rangle$  **to'plam** **ini** **hisob** **lanadi**, **kerak** **li** **manzili** **data**  
**funksiyasi** **tomon** **idan** **olin** **ish** **im** **umkin**.  $T[N]$  **massiv** **dan** **farqliq** **iravish** **da**  
 $\text{array} \langle T, N \rangle$  **-** **qiyamatlar** **ko'chirilishi**  
 (qiyamat **bo'yicha** **funksiyalar** **ga** **o'tishi**)  
**im** **umkin** **bo'lgan** **vaqiyamatlar** **ia** **vtomatik** **kravish** **da** **ko'rsatkichlar** **ga** **aylantiril**  
**maydigan** **to'la** **huquq** **litip** **dir**.  $\text{array} \langle T, N \rangle$  **dan** **tayanch** **sifat** **ida** **foydalanish** **im** **umkin** (lekin **ehtiyo** **tkorlik** **bilan**,

faqatharqandaykonteynerkabi, standartkonteynerlarbumaqsaduchunmo‘ljallanganemas, chunki, xususan, virtual funksiyalarnio‘zichigaolmaydi). deque va vectorkabibirxiltarzdaindeksitomonidanelementlarga murojaat qilishmumkin. Elementlarsoninio‘zgartirishmumkinemas, shuninguchun array elementlarnikiritishyokio‘chirishuchunhechqandayfunksiyalaraniqlanmaydi. fillfunksiyasimassivniqiymatnususxalaribilanto‘ldiradi.

2.4-jadval. Ketma-ketkonteynerlaruchunasosiyfunksiyalar

Headers		<array>	<vector>	<deque>	<forward_list>	<list>
Members		array	vector	deque	forward_list	list
	constructor	implicit	vector	deque	forward_list	list
	destructor	implicit	~vector	~deque	~forward_list	~list
	operator=	implicit	operator=	operator=	operator=	operator=
iterators	begin	begin	begin	begin	begin before_begin	begin
	end	end	end	end	end	end
	rbegin	rbegin	rbegin	rbegin		rbegin
	rend	rend	rend	rend		rend
const iterators	cbegin	cbegin	cbegin	cbegin	cbegin cbefore_begin	cbegin
	cend	cend	cend	cend	cend	cend
	crbegin	crbegin	crbegin	crbegin		crbegin
	crend	crend	crend	crend		crend
capacity	size	size	size	size		size
	max_size	max_size	max_size	max_size	max_size	max_size
	empty	empty	empty	empty	empty	empty
	resize		resize	resize	resize	resize
	shrink_to_		shrink_to_	shrink_to_		

	fit		fit	fit		
	capacity		capacity			
	reserve		reserve			
element access	front	front	front	front	front	front
	back	back	back	back		back
	operator[]	operator[]	operator[]	operator[]		
	at	at	at	at		
modifiers	assign		assign	assign	assign	assign
	emplace		emplace	emplace	emplace_after	emplace
	insert		insert	insert	insert_after	insert
	erase		erase	erase	erase_after	erase
	emplace_back		emplace_back	emplace_back		emplace_back
	push_back		push_back	push_back		push_back
	pop_back		pop_back	pop_back		pop_back
	emplace_front			emplace_front	emplace_front	emplace_front
	push_front			push_front	push_front	push_front
	pop_front			pop_front	pop_front	pop_front
	clear		clear	clear	clear	clear
	swap	swap	swap	swap	swap	swap
list operations	splice				splice_after	splice
	remove				remove	remove
	remove_if				remove_if	remove_if
	unique				unique	unique
	merge				merge	merge
	sort				sort	sort
	reverse				reverse	reverse
observers	get_allocator		get_allocator	get_allocator	get_allocator	get_allocator
	data	data	data			



## NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1. C++



- dasturlashtilida eng samarali qaysi kutubxonasi hisoblanadi va nima uchun?
2. Kalitlarning qiymati bo'yicha tartiblangan qanday to'plamlarni bilasiz?
  3. Bir va ikki baytli belgilarni to'plamini ma'noni bildirgan va ularning formatlarini bo'yicha nimalarni bilasiz?
  4. Iteratorlarni qanday to'plam va nima uchun?
  5. Har bir aniq STL sinfi uchun iteratorlarni to'plamini ma'noni bildirgan va ularning hatasi? Har bir turini tushuntirib bering?
  6. STL kutubxonasi to'plamlar bilan ishlashimni ko'rib beradigan, mashhur algoritmlarni optimal tatbiqlarini vakattamajmi'ini o'z ichiga oladi. Bu algoritmlarni nechaguruhga bo'linadigan va qaysilar?
  7. Konteynerlar qanday sinflar hisoblanadi?
  8. Konteynerlarni nechta turiga bo'lishim mumkin va qaysilar?
  9. Har qanday konteynerlarning bo'lishi shart bo'lgan xususiyatlarni sanab bering?
  10. Ixtiyoriy konteynerlarning hajmi haqidama'lumot olish uchun qanday usullar mavjud?
  11. Iteratorlarda begin(), end(), cbegin(), cend(), rbegin(), rend(), crbegin() va crend() erkin funksiyalarini nimalarni aniqlashim mumkin?
  12. Chiziqlik konteynerlarni belgilangan qiymatlardan qaysi funksiyani sinchaqirib to'ldirishim mumkin?
  13. Barcha konteynerlarni tenglik va tengsizlik uchun taqqoslashim mumkin va ularning mazmunini qanday funksiyalar yordamida almashtirishim mumkin?
  14. Allocator nima uchun ishlatiladi?
  15. Allocator xotira boshqarishning minimal birligini belgilaydigan va bir qator yordamchilarni taqdim etadigan element tipiga bog'liqdir.

Bu vazifa nechta vaqanda asosiy funksiyalari yordamida amalga oshiriladi?

16. Bir aloqali ro'yxatning nimalarini yaratish uchun `std::vector` qanday yaratilgan alokator foydalaniladi?
17. Qaysi funksiyadan foydalanib, elementlarni oldindan joylashtirish mumkin?
18. Ikki aloqali ro'yxat va ikki tomonlama navbatning farqlarini sanab bering?
19. Dinamik va statik massivlarning o'xshash tomonlarini misollar yordamida tushuntirib bering?
20. Statik massiv elementlarni kiritish yoki o'chirish uchun qanday funksiyalar aniqlangan?



## AMALIY KO'NIKMA VA MALAKALARNI ANIQLASH HAMDA RIVOJLANTIRISH UCHUN ASSISMENT TOPSHIRIQLARI.

<b>BIRINCHI ASSISMENT TOPSHIRIG'I</b>	
	<p>STL kutubxonasiga oid berilgan quyidagi dastur bo'yicha berilgan topshiriqlar kerakli fragmentlari asosida bajarilgan.</p> <p> Bunday dasturda <code>std::vector</code> o'zgartirishlarini topshiriqlar bosqichma – bosqich amalga oshiriladi.</p>
dastur	topshiriqlar
<pre>#include &lt;iostream&gt; #include &lt;vector&gt;  using namespace std;  int main(){     vector v;     int i;</pre>	<p>1. Dasturda <span style="float: right;">STL</span> kutubxonasining nechasi fivauning funksiyalar ishlatilgan.</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>2. Dasturdagi STL sinfini o'zgartirib dasturni qayta tuzing. Qaysi qismlar o'zgartirilganini qayd qiling.</p> <p>_____</p> <p>_____</p> <p>_____</p>

<pre>         cout &lt;&lt; v.size() &lt;&lt; endl;          for(i=0; i&lt;10; i++) v.push_back(i) ; </pre>	<p>3. Dasturdavektorelementlarigaxaqiqiytiqliqimat larniberishuchundasturfragmentaniqyozing.</p> <hr/> <hr/> <hr/> <hr/>
<pre>         cout &lt;&lt; v.size() &lt;&lt; endl; </pre>	<p>4. cout &lt;&lt; v.front() &lt;&lt; endl;fragmentgaalmatirishvariantlarinianiqlangvada sturgayozib, tekshiribko‘ring.</p> <hr/> <hr/>
<pre>         for(i=0; i&lt;10; i++) cout &lt;&lt; v[i] &lt;&lt; " ";         cout &lt;&lt; endl;          cout &lt;&lt; v.front() &lt;&lt; endl;         cout &lt;&lt; v.back() &lt;&lt; endl;          vector&lt;int &gt;::iterator p = v.begin();         while (p != v.end()) {             cout &lt;&lt; *p &lt;&lt; " " ;             p++;         } </pre>	<p>5. vector&lt;int&gt;::iterator p = v.begin();dasturningfragmentidanima o‘zaksinit opgan.</p> <hr/> <hr/> <hr/> <hr/>

<pre> system("pa use"); return 0; } </pre>	
6. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____ 7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.	

<b>IKKINChIASSIMENTTOPShIRIG‘I</b>	
🎓	Konteynersinflargaoidberilganquyidagidasturbo‘yichaberilganto pshiriqlarkeraklifragmentlariasosidabajaring. 👉 Bundadasturdagiba‘zio‘zgartirishlarnitopishorqalitopshiriqlarb osqichma – bosqichamalganoshriladi.
dastur	topshiriqlar
<pre> #include "stdafx.h"  #include &lt;iostream&gt;  int main() {  std::vector&lt;int&gt; myVector; for (int count=0; count &lt; 5; ++count)  myVector.push_bac k(count);  std::vector&lt;int&gt;: </pre>	1. Dasturdagiengkattaxonitoping. _____ _____ _____ 2. Dasturdaaniqlangan myVector, myList, mySet konteynerlarningelementlaribirxilbo‘lishiuchun keraklidasturfragmentlariniyozing. _____ _____ _____ _____ 3. itV qandaykonteynervanimauchun. _____ _____ _____

<pre> :const_iterator itV;     itV = myVector.begin();     while (itV != myVector.end())         {             std::cout &lt;&lt; *itV &lt;&lt; " ";             ++itV;         }          std::cout &lt;&lt; '\n';          std::list&lt;int &gt; myList;         for (int count=0; count &lt; 5; ++count) myList.push_back( count);  std::list&lt;int&gt;::c onst_iterator itL;     itL = myList.begin();     while (itL != myList.end())         {             std::cout &lt;&lt; *itL &lt;&lt; " ";             ++itL;         } </pre>	<p>4. Dasturdasturchiningyanabiroddiyammo ko'rolxatosinitoping.</p> <hr/> <hr/> <hr/> <p>5. Dasturda gibarchakonteynerlarniyarimele mentlarinichiqarishdasturfragmentlarinituzing.</p> <hr/> <hr/> <hr/>
---	--

```

    std::cout <<
'\n';

    std::set<int>
mySet;

mySet.insert(28);

mySet.insert(01);

mySet.insert(80);

mySet.insert(9);

mySet.insert(3);

std::set<int>::co
nst_iterator itS;
    itS =
mySet.begin();
    while (itS !=
mySet.end())
    {
        std::cout
<< *itS << " ";
        ++itS;
    }

    std::cout <<
'\n';
    system("pause
");
}

```

6. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. \_\_\_\_\_
7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.

## UChINChIASSISMENTTOPShIRIG'I

🎓 Ketma-ketkonteynerlardanfoydalanishgaoidberilganquyidagidasturbo'yichaberilgantopshiriqlarkeraklifragmentlariasosidabajaring.  
 👆 Bundaydasturdagiba'zio'zgartirishlarnitopishorqalitopshiriqlar bosqichma – bosqichamalganoshiriladi.

dastur	topshiriqlar
<pre>#include "stdafx.h" #include &lt;iostream&gt;</pre>	<p>1. Dasturdagi yangi tiplarni aniqlang.</p> <hr/> <hr/> <hr/>
<pre>#include &lt;vector&gt; #include &lt;cstring&gt; #include &lt;string&gt;  using namespace std;  struct student {     char fio[80];     short age;      student(const char* fio, short age) : age(age) {  strncpy_s(this -&gt;fio, fio,</pre>	<p>2. Dasturdagi ketma-ketkonteynerlardasturfragmentini yozing.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
<pre>strncpy_s(this -&gt;fio, fio,</pre>	<p>3. Dasturdaki raktsiz bo'lgan dasturfragmentlarini o'chirib tashlang</p> <hr/> <hr/> <hr/>





```

    }
};
inline
ostream&
operator
<<(ostream&
out, const
trimatrix&
obj) {
    for (int i =
0; i <
obj.size());
i++) {
    for (int j =
0; j <
obj[i].size());
j++)
        cout <<
obj[i][j] << "
";
        cout << endl;
    }
return out;
}
int main(int
argc, char
**argv) {

    setlocale(LC_A
LL, "rus");

    vector<student
> group;

    group.push_bac
k(student("Bol
tavev B.B.",

```

```

18));

group.push_back(student("Ami
nov A.A.",
21));

group.push_back(student("Kar
imov A.Sh.",
19));

    for (int i =
0; i <
group.size();
i++)
    cout <<
group[i] <<
endl;

    cout <<
trimatrix(4);

    int k, n;
    vector<bool>
a(n = rand() +
1, true);
    a[0] = a[1] =
false;
    for (k = 2; k
< n; k++)
        for
(vector<bool>:
:iterator j =
a.begin() + k
+ k;
            j <
a.end()); j +=

```

```


k)
    *j =
false;
    const int
line = 10;
    k = 0;
    for
(vector<bool>:
:iterator j =
a.begin(); j <
a.end(); j++)
    if (*j) {
        cout << j -
a.begin() <<
"\t";
        if (0 == ++k
% line) cout
<< endl;
    }
    if (k % line
!= 0) cout <<
endl;

system("pause"
);
    return 0;
}

```

6. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. \_\_\_\_\_
7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.

### 1.3. Assosiativvatartiblanmaganassosiativkonteynerlar

 Assosiativ konteynerlarningasosiyfoydalanishko‘nikmalari, assosiativkonteynerlar (set, map, multiset, multimap) vatartiblanmaganassosiativkonteynerlar (unordered\_set, unordered\_map, unordered\_multiset, unordered\_multimap) bilanyaqindantanihib, ularningxususiyatlar, funksiyalari, sinflari, yaratishvafoydalanishuslublari, amallari, talablari, vazifalarivausullarikeltirilganbo‘lib,

dasturchining to'plamlar bilan ishlashdagitanlashim koniyati bo'yicha tajrib axulosalari, nazariy bilimlarni asoslash uchun dasturlar tuzib ko'rsatilgan. Shuningdek,

assosiativ konteynerlarda qidirish ustunligi va uni dastur data qqoslash bo'yicha hamiosllarkeltirilgan. Bilimlarni mustahkamlash uchun 32 tanazariy savol va amaliy ko'nikma ham damalaka larni rivojlantrish uchun 4 ta assisment topshirig'ivahar bir assismentda 10(12) ta topshiriq, jami 44 ta topshiriq berilgan.

Butopshiriqlarni bajarish mavzu ni mustahkamlash uchun xizmat qiladi.

**✍️ Kalit so'zlar.** to'plam, shablon, map, set, multimap, multiset, unordered\_set, unordered\_map, unordered\_multiset, unordered\_multimap, satrli oqim, iterator, dinamik ma'lumotlartuzilmasi, pridekat, konteynersinflar, allocator, equal\_range, operator==, operator=, tartiblangan assosiativ konteyner, tartiblanmagan assosiativ konteyner, unikalkalit, xesh funksiya, xeshlash, konstruktor, modifikator, qidirish usullari, pair, try - catch funksiyasi.

**📑 Bilish shart bo'lgan tushunchalar.** Tiptushunchasi, sinf va sinf ob'ekti, to'plam, statik va dinamik massiv, element gamurojaat va ko'rsatkich, funksiyavako'rsatkich, konteyner, ketma-ket konteynerlar, xesh, tasodifiy qiymat, istisno holat, ma'lumotlarni kirish va chiqishi, oqim, dasturlashga oid dastlabki tushunchalar va C++ tilini oq'lab quvvatlovchimu hitda ishlashni bilish lozim.

### 📖 Bilib olasiz.

Assosiativ konteynerlarning asosiy foydalanish ko'nikmalari, assosiativ konteynerlar (set, map, multiset, multimap), tartiblanmagan assosiativ konteynerlar (unordered\_set, unordered\_map, unordered\_multiset, unordered\_multimap) bilanya qindantanishib, ularning xususiyatlar, funksiyalari, sinflari, yaratish va foydalanish uslublari, amallari, talablari, vazifalari va usullarini, dasturchining assosiativ to'plamlar bilan ishlashdagitanlashim koniyati bo'yicha tajrib axulosalari, xeshlash, tartiblangan va tartiblanmagan konteynerlarning sinflari xususiyatlari, usullarini, dasturlashdagio'rniva allocator orqali yaratish va foydalanish usullarini o'rganishingiz mumkin.

### REJA

1. Assosiativ konteynerlar (set, map, multiset, multimap).

2. Tartiblanmagan assosiativ konteynerlar (*unordered\_set*, *unordered\_map*, *unordered\_multiset*, *unordered\_multimap*).

### KIRISH

Konteynerlar vaketma-ket konteynerlar bo'yicha nazariy va amaliy imkoniyatlarini bilamiz. Shuning uchun konteynerlarning ikkinchi turidagi assosiativ konteynerlar ni boshqa konteynerlar farqlib olishi kerak. Assosiativ konteynerlar kalit soisdama' lumotlarni topiladigan dantezizlab topi shunchun ishlatiladi.

Tartiblangan konteynerlar muvozanatlashgan binardaraxt ustiga quriladi va qat'iy tartibga tayanadi ("kichik" amali [ $<$ ] operatori asosida hisoblangan). Ikkita element bir-biridan kichik bo'lmasa, ekvivalent hisoblanadi.

Buning kutubxonasi 4 ta asosiy set (to'plam), *multiset* (multito'plam, ko'pto'plam), *map* (lug'at) va *multimap* (mulilug'at, ko'plug'at) assosiativ konteynerlar bilan ishlashga qaratilgan. Bu konteynerlar *Key* kalitini o'zlariga asosiy paramert sifatida o'ladi va *Compare* munosabat bo'yicha tartiblaydi.

Tartiblash *Key* paramert bilan amalsha oshiriladi. Shuningdek, *map* va *multimap* erkin tipida *Key* bilan assosiativlanadigan (sherikdir). *Compare* ob'ektining tipiga qoslanaligan konteyner ob'ekti (*comparison object*) deb aytiladi.

Shuning uchun bu konteynerlarning umumiy xususiyati va amallaridan boshlaymiz. Barcha assosiativ konteynerlar quyidagi amallarni qo'llab-quvvatlaydi:

*count* — elementlar sonini qaytaradi, belgilangan kalit bo'yicha elementlar sonini qaytaradi;

*find* — element gacha tekshirib qaytaradi, agar bunday bo'lmasa *end()* funksiyasini vazifasini bajaradi.

*equal\_range* - berilgan intervaldagi barcha elementlar uchun iteratorlar juftligini qaytaradi.

Kalitlarning tenglik va munosabatlarning ekvivalentligi kalitlar uchun shartli taqqoslash va (*not operator*) tahlil qilamiz.

*key1* va *key2* kalitlar o'zaro teng hisoblanadi, agar taqqoslanadigan ob'ektlar uchun *comp* chinqiyimat qaytarsa, ya'ni:

$comp(k1, k2) == false \ \&\& \ comp(k2, k1) == false$
--

Assosiativ konteynerlar har qanday kalit qiymat uchun bitta eng katta elementning qiymatini saqlasa, unikal (takrorlanmaydigan) kalitlarni (*unique keys*) qo'llab-quvvatlaydi. Akshod aularteng qiymatlik kalitlarni (*equal keys*) ham qo'llab-quvvatlaydi.

*set* va *map* konteynerlarunikalkalidlarni, *multiset* va *multimap* konteynerlartengqiymatlikalidlarbilanishlashgamo‘l jallangan.

*set* va *multiset* konteynerlaruchunaniqlanadigantipvakalittipibirxilbo‘ladi.

*map* va *multimap* konteynerlaruchunpair<const Key, T>jufligibo‘ladi.

Assosiativkonteynerlarningiteratorlariikkimonlamaiteratorlarsinfi gakeradi. *Insert*konteynerhavolalarivaiteatorlarninginkorqilmaydi. Elementlarnio‘chirishda*erase* faqatiteratorvahavolalarniinkorqiladi.

3.1-jadvaldaassosiativkonteynerlarningtalablarikeltirilgan (konteynerlargaqo‘shimacha). Undaxbuassosiativsinf,  $a$  –  $X$  ningqiymati, Agar $X$  unikalkalitniko‘llabquvvatlasa,  $a\_uniq$  –  $X$  ningqiymati, Agar $X$  birnechtakalitniko‘llabquvvatlasa,  $a\_eq$  –  $X$  ningqiymati,  $i$  va  $j$  – iteratorningkirishtalablariniqanoatlantiruvchi va  $value\_type$  elementniko‘r satadi (beradi),  $[i, j)$  - interval,  $p$  –  $a$  uchunruxsatberilganiterator,  $q$  –  $a$  uchuno‘zgartiriladiganiterator,  $[q1, q2)$  -  $a$  daruxsatberilganinterval,  $t$  -  $X::value\_type$  ningqiymati,  $k$  -  $X::key\_type$  ningqiymati.

3.1-jadvaldaassosiativkonteynerlarningtalablari.

Ifoda	Qaytariladigantipi	tasdiq/ oldingiyokikeyingiholatigaizo x	murakkabligi
$X::key\_type$	Key	.	Kompilyatsiyavaqti
$X::key\_compare$	Compare	less<key_type>joriyholat	Kompilyatsiyavaqti
$X::value\_compare$	binarpredikator	xuddi, key_compare uchun set va multiset; munosabatitartiblanganjuftlik, map va multimap uchunchaqirilganbirinchielementdek (Key)	Kompilyatsiyavaqti
$X(c)$	.	Bo‘shkonteyner yaratish;	Doimiy

X a(c);		taqqoslashob'ektisifatidafoydalanishuchun.	
X() X a;	.	Bo'shkonteyneryaratish; taqqoslashob'ektisifatidaCompare() danfoydalanishuchun	Doimiy
X(i,j,c) X a(i,j,c);	.	Bo'shkonteyneryaratish [i, j)intervaldanqiymatqo'shish; taqqoslashob'ektisifatidafoydalanishuchun..	NlogN (N - idan jgacha); chiziqli, agar [i, j) value_comp()bilansaralanganbo'lsa
X(i,j) X a(i,j);	.	Bo'shkonteyneryaratish [i, j)intervaldanqiymatqo'shish; taqqoslashob'ektisifatidaCompare() danfoydalanishuchun	NlogN (N - idan jgacha); chiziqli, agar [i, j) value_comp()bilansaralanganbo'lsa
a.key_comp() )	X::key_compare	Taqqoslashob'ektiniqaytaradi	doimiy
a.value_comp() )	X::value_compare	Taqqoslashob'ektiuchunyaratilgan value_compare ob'ektniqaytaradi	doimiy
a_unique.insert(t)	pair<iterator, bool>	Agarkonteynerdat kalitlielementbo'lmasa, t niqo'shish, juftlikdakomponentning bool qiymatiqo'shishbajarilganligini, iteratorkalitlielementniko'rsatadi	logarifmik
a_eq.insert(t)	iterator	tniqo'shadi iteratorniqaytaradi, qo'shishilganelementniko'rsatadi.	logarifmik

a.insert(p, t)	iterator	tniqo‘shish, agar faqat vafaqat unikalitlik konteynerda kalitga teng kalitli element bo‘lmasa; hardim nusxalar bilan konteynerlarga tniqo‘shish. hardim kalitli elementni ko‘rsatuvchi iteratorni qaytaradi. p iterator – qo‘shishqa erdan boshlanishini ko‘rsatuvchi izoh	umuman logarifmik, lekin domiy gaintiladi, agar t to‘g‘ri p niyoni ga qo‘yilgan bo‘lsa.
a.insert(i, j)	Natija ishlatilmaydi	[i, j) intervaldankonteynerga elementlarni qo‘shadi;	Umuman, $N \log(\text{size}() + N)$ (N – idan j gacha); chiziqli, agar [i, j) value_comp() bilan kelishgan holda saralangan bo‘lsa
a.erase(k)	size_type	konteynerda k kalitga teng bo‘lgan barcha elementlarni o‘chiradi. o‘chirilgan elementlarning soni niqaytaradi.	$\log(\text{size}()) + \text{count}(k)$
a.erase(q)	Natija ishlatilmaydi	q bilan ko‘rsatilgan elementni o‘chiradi	domiy gaintiladi
a.erase(q1, q2)	Natija ishlatilmaydi	[q1, q2) intervaldagi barcha elementlarni o‘chiradi.	$\log(\text{size}()) + N$ , N - q1 dan q2 gacha.
a.find(k)	iterator; const_iterator – auchuno‘zgar mas	k kalitli elementni ko‘rsatuvchi iterator yoki agar bunday element to‘pilmasa a.end() qaytaradi	logarifmik
a.count(k)	size_type	k kalitli elementlarning soni niqaytaradi.	$\log(\text{size}()) + \text{count}(k)$



a.lower_bound(k)	iterator; const_iterator auchun o'zgarmas	k kalitli elementdankichikbo'lgan birinchi elementni ko'rsatuvchi iteratorni qaytaradi	logarifmik
a.upper_bound(k)	iterator; const_iterator auchun o'zgarmas	k kalitli elementdankattabo'lgan birinchi elementni ko'rsatuvchi iteratorni qaytaradi	logarifmik
a.equal_range(k)	pair<iterator, iterator>; pair<const_iterator, const_iterator> auchun o'zgarmas	make_pair(lower_bound(k), upper_bound(k)) ga ekvivalent	logarifmik

Assotsiativ konteyner iteratorlarining asosiy xususiyatishundaki, ular konteynerlar orqali o'sis tartibidagi kalitlartartibida iteratsiyalanadi, buyerdan o'sis tartibi ularni yaratish uchun ishlatilgan taqqoslash bilan belgilanadi.

Ikki iaixtiyoriy o'zguruvchan  $I$  va  $j$  iteratorlar uchun  $I$  dan  $j$  gachamasofamusbato'ladi, ya'ni  $value\_comp(*j, *i) == false$ . Unik kalitli assotsiativ konteynerlar uchun kuchli roq  $value\_comp(*i, *j) == true$  holatsaqlanib turibdi.

Barcha tartiblangan konteynerlar ikki tomonlama iteratorlaryordamida elementlarga murojaat qilish imkoniniberadi.

Kalita sosidaqidirishdaberilgan kichikbo'lgan birinchi elementni `lower_bound()` funksiyasi va birinchi katta elementni `upper_bound()` funksiyasi yordamida amalga oshiriladi (3.1-jadvalga qarang). Agar `mymap` konteyner berilgan bo'lsa, unda `mymap.equal_range(key)` funksiyasi bilan ekvivalent bo'ladi (3.1-dasturga qarang).

3.1-dastur. Tartiblangan konteynerlar funksiyalari.

```
// created by Mbbahodir
#include "stdafx.h"

#include <iostream>
```

```

#include <map>

using namespace std;
int main ()
{
    map<char,int> mymap;
    map<char,int>::iterator itlow,itup;

pair<map<char,int>::iterator,map<char,int>::iterator>
ret, make_ret;

    mymap ['a']=101;    mymap ['c']=303;    mymap
['f']=404;    mymap ['b']=202;

    cout << mymap.count('c') << endl;

    cout << mymap.find('c')->second << endl;

    ret = mymap.equal_range('b');

    char key = 'b';
    make_ret =
make_pair(mymap.lower_bound(key),mymap.upper_bound(ke
y));

    cout << "Kichik bo'lmagan element: ";
    cout << ret.first->first << " => " << ret.first-
>second << '\t';
    cout << make_ret.first->first << " => " <<
make_ret.first->second << '\t';
    itlow = mymap.lower_bound ('b');
    cout << itlow->first << " => " << itlow->second <<
endl;

    cout << "katta bo'lgan birinchi element: ";
    cout << ret.second->first << " => " << ret.second-
>second << '\t';
    cout << make_ret.second->first << " => " <<

```

```

make_ret.second->second << '\t';
    itup = mymap.upper_bound ('b');
    cout << itup->first << " => " << itup->second <<
endl;

system("pause");
return 0;
}

```

### 3.1-dastur. Output

```

1
303
Kichik bo'lmagan element: b => 202    b => 202    b => 202
katta bo'lgan birinchi element: c => 303    c => 303    c => 303

```

3.1-dasturtahlili. Dasturda 4 taelementdaniboratmap konteynerberilgan. 's' kalitlielementsoni count() funksiyabilananiqlagan. Xuddishukalitbilanuningqiymati find() funksiyabilantopilgan. equal\_range('b'), mymap.lower\_bound('b'); mymap.upper\_bound('b'); make\_pair(mymap.lower\_bound(key), mymap.upper\_bound(key));funksiyalariningekvivaletishlariko'rsatilgan.

#### **Takrorlanmaydigan (unikal)**

**tartiblanganelementlarto'plami<set>**. set<K, C = less<K>, A = allocator<K>>

Bukonteynerjoyidaqiymatlarnio'zgartirishgayo'lqo'ymaydi. <set> buassosiativkonteynerbo'lib, unikalkalitniqo'llabquvvatlaydi (bixilqaymatlikalitlarniqo'llabquvvatlamaydi) vakalityordamidatezqidirishimkoniniberadi.

set<K>::iterator va set<K>::const\_iterator iteratorlarfunksionalekvivaletthisoblanadivao'zgarganda const K& qiymatqaytaradi. Set

to'plamdaqiymatnio'zgartirishuchunavvalo'chiribtashlabso'ngyangiqiy matniyokio'zgartirilganvariantiqo'shiladi.

<set>ningshablioni:

```

template <class Key, class Compare = less<Key>,
         template <class U> class Allocator = allocator>

```

<set>sinfnigochiqususiyatlari, operatorlarivafunksiyalari:

1. Typedefoperatorlari:

```

typedef Key key_type;
typedef Key value_type;
typedef Allocator<Key>::pointer pointer;
typedef Allocator<Key>::reference reference;
typedef Allocator<Key>::const_reference
const_reference;
typedef Compare key_compare;
typedef Compare value_compare;
typedef iterator;
typedef iterator const_iterator;
typedef size_type;
typedef difference_type;
typedef reverse_iterator;
typedef const_reverse_iterator;

```

## 2. Xotiraniajratishvabo‘shatishoperatorlari (allocation/deallocation):

```

set(const Compare& comp = Compare());
template <class InputIterator>
set(InputIterator first, InputIterator last,
    const Compare& comp = Compare());
set(const set<Key, Compare, Allocator>& x);
~set();
set<Key, Compare, Allocator>& operator=(const
set<Key, Compare,
    Allocator>& x);
void swap(set<Key, Compare, Allocator>& x);

```

## 3. Ruxsatberishvositalarinigoperatorlari (accessors):

```

key_compare key_comp() const;
value_compare value_comp() const;
iterator begin() const;
iterator end() const;
reverse_iterator rbegin() const;
reverse_iterator rend() const;
bool empty() const;
size_type size() const;
size_type max_size() const;

```

## 4. Qo‘shishvao‘chirishoperatorlari (insert/erase):

```

pair<iterator, bool> insert(const value_type& x);
iterator insert(iterator position, const value_type& x);

```

```

template <class InputIterator>
void insert(InputIterator first, InputIterator last);
void erase(iterator position);
size_type erase(const key_type& x);
void erase(iterator first, iterator last);

```

5. To'plamamallari (set operations):

```

iterator find(const key_type& x) const;
size_type count(const key_type& x) const;
iterator lower_bound(const key_type& x) const;
iterator upper_bound(const key_type& x) const;
pair<iterator, iterator> equal_range(const
key_type& x) const;

```

6. Taqqoslashoperatorlari:

```

template <class Key, class Compare, class Allocator>
bool operator==(const set<Key, Compare, Allocator>&
x,
    const set<Key, Compare, Allocator>& y);

template <class Key, class Compare, class Allocator>
bool operator<(const set<Key, Compare, Allocator>& x,
    const set<Key, Compare, Allocator>& y);

```

<set>to'plamda:

**iterator** ( **iterator** )bu – const value\_type ko'rsatuvchidomi yikkitom nlamaiterator.

Aniqtipni belgilash realizatsiya qilish gabog'liqva Allocator da aniqlanadi.

**const\_iterator** - ham iterator ( iterator ) kabidir.

**size\_type** - ishora sizbutuntip, Aniqtipni belgilash realizatsiya qilish gabog'liqva Allocator da aniqlanadi.

**difference\_type** - ishora libutuntip, Aniqtipni belgilash realizatsiya qilish gabog'liqva Allocator da aniqlanadi.

Elementlarni qo'shishi insert() funksiyasi tomonidan amalga oshiriladi va bu funksiyaning bir nechta variantlari mavjud:

- juft iteratorlarni kiritishda interval dan qiymat qo'shish;

```

template <class InputIterator>
void insert (InputIterator first,
InputIterator last);

```

-qo'yish mumkin bo'lgan joyga ko'rsatish orqali bitta qiymat qo'shish (agaryang element bevosita oldin pozitsiyani egallasa, o'rnatish o'z garmas vaqt mobaynida amalga oshiriladi);

```
pair<iterator, bool> insert (const value_type& val);
```

-berilgan qiymatni qo'shish (asosiy usul);

```
iterator insert (iterator position, const value_type& val);
```

Oxirgi variant insert funksiyasi birjuftli (iterator, Boolean) qaytaradi.

Juftlikdagi birinchi element qo'shish joyini yoki topilgan qiymatini, ikkinchi joylashtirilgan qiymatni qo'shish bajarilganligi (true) yoki qo'shish vaqtida topilgan elementning (false) bildiradi.

Elementlarni o'chirish erase funksiyasi orqali amalga oshiriladi. Bu funksiya qiymat yoki berilgan elementlarni o'chirish yoki iteratorlar intervalini qabul qiladi.

```
(1) size_type erase (const value_type& val);  
(2) void erase (iterator position);  
(3) void erase (iterator first, iterator last);
```

Birinchi parametrga elementni o'chiradi, qolgan ikki parametrga elementni ko'rsatadigan iterator qaytaradi. Birinchi variant o'chirilgan elementlarning sonini beradi (set bo'lsa, 1 yoki 0 qiymat beradi).

3.2-dastur. Set to'plamga elementlarni qo'shish.

```
// created by MBBahodir  
#include "stdafx.h"  
#include <iostream>  
#include <set>  
using namespace std;  
  
void erase_subs(set<int>&s, const int &subs)  
{  
    set<int>::iterator p, pe, pd;  
    p = s.begin(), pe = s.end();  
  
    while (p != pe)  
    {
```

```

        if (p == s.find(subs))
            p = s.erase(p);
        else
            ++p;
    }
}

int main ()
{
    set<int> myset;
    set<int>::iterator it;
    pair<set<int>::iterator,bool> ret;

    for (int i=1; i<=5; ++i) myset.insert(i*10);

    ret = myset.insert(20);

    if (ret.second == false) it=ret.first;

    myset.insert (it,25);
    myset.insert (it,24);
    myset.insert (it,26);

    int myints[] = {5,10,15,60,28};
    myset.insert(myints,myints+5);

    cout << "myset contains after insert:";
    for (it=myset.begin(); it!=myset.end(); ++it)
        cout << ' ' << *it;
    cout << '\n';

    // set to 'plam tltmentlarini o'chirish
    it = myset.begin();
    ++it;

    myset.erase (it);

    myset.erase (40);
}

```

```

it = myset.find (60);
myset.erase (it, myset.end());

cout << "myset contains after erase :";
for (it=myset.begin(); it!=myset.end(); ++it)
    cout << ' ' << *it;
    cout << '\n';

erase_subs(myset,24);

cout << "myset contains after erase_subs :";
for (it=myset.begin(); it!=myset.end(); ++it)
    cout << ' ' << *it;
    cout << '\n';
system("pause");
return 0;
}

```

### 3.2-dastur. Output

```

myset contains after insert: 5 10 15 20 24 25 26 28 30 40 50 60
myset contains after erase : 5 15 20 24 25 26 28 30 50
myset contains after erase_subs : 5 15 20 25 26 28 30 50

```

**Tartiblangan multito'plam <set>** - <multiset>. <set> to'plamdan farqli o'laroq, <multiset> insert funksiyasi qiymat qo'shish uchun qo'shiladigan qiymatni ko'rsatuvchi iterat orqaytaradi. <multiset> - assosiativ konteyner to'lib, teng qiymatliklarni saqlaydi (mumkin qadar birkalit qiymatli elementlarni saqlaydi) va kalit orqalitezqidirishim koniniberadi. <multiset> ning shabloni (<set> nishabloni bilan birxil):

```

template <class Key, class Compare = less<Key>,
         template <class U> class Allocator = allocator>

```

<multiset> sinfning ochiq xususiyatlari, operatorlar va funksiyalari:

1. Typedef operatorlari - <set> ni kibirxil.
2. Xotirani ajratish va bo'shatish operatorlari (allocation/deallocation):

```

multiset(const Compare& comp = Compare());

```



```

template <class InputIterator>
multiset(InputIterator first, InputIterator last,
         const Compare& comp == Compare());
multiset(const multiset<Key, Compare, Allocator>& x);
~multiset();
multiset<Key, Compare, Allocator>& operator=(const
multiset<Key,
                                     Compare, Allocator>& x);
void swap(multiset<Key, Compare, Allocator>& x);

```

3. Ruxsatberishvositalariningoperatorlari (accessors)

<set>nikibilanbirxil.

4. Qo'shishvao'chirishoperatorlari (insert/erase):

```

iterator insert(const value_type& x);
iterator insert(iterator position, const value_type& x);
template <class InputIterator>
void insert(InputIterator first, InputIterator last);
void erase(iterator position);
size_type erase(const key_type& x);
void erase(iterator first, iterator last);

```

5. To'plamamallari (set operations) - <set>nikibirxil.

6. Taqqoslashoperatorlari:

```

template <class Key, class Compare, class Allocator>
bool operator==(const multiset<Key, Compare, Allocator>& x,
               const multiset<Key, Compare, Allocator>& y);

template <class Key, class Compare, class Allocator>
bool operator<(const multiset<Key, Compare, Allocator>& x,
              const multiset<Key, Compare, Allocator>& y);

```

<multiset>to'plamda:

**iterator** ( **iterator** )bu — const value\_type ko'rsatuvchidomiyikkitomnlamaiterator.

Aniqtipnibelgilashrealizatsiyaqilishgabog'liqva Allocatordaaniqlanadi.

**const\_iterator** - hamiterator ( iterator ) kabidir.

**size\_type** - ishorasizbutuntip, Aniqtipnibelgilashrealizatsiyaqilishgabog'liqva Allocatordaaniqlanadi.

**difference\_type** - ishoralibutuntip, Aniqtipnibelgilashrealizatsiyaqilishgabog'liqva Allocatordaaniqlanadi.

<set> va <multiset> to'plamlarning xususiyatlari, operatorlari va funksiyalari umummanoligandabirxil. Shuning uchun 2 tato'plamni birga o'rganish maqsadga muvofiq.

*set* va *multiset*

uchun foydalaniladigan iteratorlar o'z garmasik tomonlama iterator bo'lga nligi uchun *algorithm* kutubxonasi damazkur tipni qo'llamaydigan funksiyalari uzatish mumkin emas.

Shuning uchun joriy sinfda aniqlangan massiv elementlari bilan ishlaydigan usullari, funksiyalardan (umumlashgan algoritmlar o'rnida) foydalanish kerak.

Yuqoridagi kodlab o'tilganidek, *multiset* to'plamning *set* to'plamdan farqi faqat kalit bir nechta bir xil qiymatlik kalitlarni saqlashidir. Shuning uchun ixtiyoriy sohada ishlatishda *multiset* sinfining ishlatilishi *set* sinfiniki dan farq qilmaydi.

*set* va *multiset* sinflarining ob'ektlarining tipikalit bilan yonma-yan bittashablonli parametrolishi mumkin.

Bu shablon taqqoslash (*comp*) funksiyasidir.

Agar shundan funksiyamavjud bo'lsa, oshkormas `less<>` funksiyasi bilan beriladi (`amali`).

*set* (*multiset*) sinflarining ob'ektlarini quyidagi konstruktorga bilan yaratish mumkin:

Bo'sh to'plam konstruktoriga - `set<type, comp> ar;` yoki `set<type, comp> ar(Comp);`

Nusxalash konstruktoriga - `set<type, comp> ar(other);`

Iteratorlaryordamida qo'shish konstruktoriga - `set<type, comp> ar(first, last);` yoki `set<type, comp> ar(first, last, Somp);`

Ro'yxat asosidagi initsializatsiya qilish konstruktoriga - `set<type, comp> ar {init};` yoki `set<type, comp> ar(init);` yoki `set<type, comp> ar(init, Comp);`

Bu konstruktorgalarda `Comp` konteynerlarning kalitlarini taqqoslash funksiyasi (ixtiyoriy). Agarda sturcho'zining allokatord funksiyasi ni yaratasa, qo'shish uchun taqqoslash funksiyasini yonida konstruktorga majburiy bo'lmagan `Allocator()` funksiyasi bor.

*set* (*multiset*) sinflarining ob'ektlarini o'chirish uchun destruktorga - `ar.~set();`

**Hajmlar va o'lcham bilan ishlash usullari:** `empty()` - true qaytaradi, agarda to'plam bo'sh bo'lsa `false`, `size()` -

to'plamdagi elementlar sonini qaytaradi, `max_size()` – mumkin bo'lgan elementlar sonini qaytaradi.

**Modifikatorlari:** `clear()` – konteynerni tozalaydi, `insert()` – element qo'shish, `erase()` – elementni o'chirish, `swap()` – o'rinalmashtirish, `emplace()` – joydagi elementni yaratish.

**Qidirish usullari:** `set` to'plam bilan ishlaganda qidirish muhim xisoblanadigan aspekt bo'lib hisoblanadi.

Qidirish nisamaralitashkil qilish uchun bir nechta o'zining usullariga ega: `count(key)`, `find(key)`, `equal_range(key)`, `lower_bound(key)`, `upper_bound(key)`

**Taqqoslash va birlashtirish amallari:** `set` to'plam uchun barcha taqqoslash amallari amal qiladi. `ning ketma-ket konteynerlarga amal qilgan barcha amal o'rinli.`

`[=]` `(operator=)` operatorni elementlar to'plamini olish uchun ishlatiladi. Masalan, `other` to'plamning nusxasini yangi bir `other_one` to'plamga nusxalash uchun ishlatiladi (`other_one = other`).

**Semantik nusxalashni amalga oshirish usuli** `other` to'plamning nusxasini yangi bir `other_one` to'plamga to'liq nusxalash uchun ishlatiladi va `other` to'plamni o'chirilmaydi, ammo uning holatini aniqlab bo'lmaydi `other_one = move(other)`.

`<set>` va `<multiset>` doirasi masalalar va dasturlarni keltiramiz.

1-masala. `[10, 50]` intervalda `N` ta ketma-ket sonlarni tasodifiy berilgan. `{10, 20, 30, 40, 50}` berilgan sonlardan joriy ketma-ketlikdanechtadan bor.

3.3(a)-dastur. 1-masalaning dasturi.

```
#include "stdafx.h"

#include <iostream>
#include <vector>
#include <set>
#include <random>
#include <ctime>
using namespace std;

int main() {
    default_random_engine rnd(time(0));
    uniform_int_distribution<unsigned> g(10, 50);
    vector<int> myVektor;
```

```

int n, k = 0;
cout << "n = "; cin >> n;
for (int i = 0; i < n; i++) {
    int d = g(rnd);
    myVektor.push_back(d);
    cout << myVektor[i] << " ";
}
set<int> mySet;
for (int i=1; i<=5; ++i) mySet.insert(i*10);
cout << endl;
for (int i = 0; i < n; i++)
    if (mySet.count(myVektor[i])) k++;
cout << k << " ta element topildi." << endl;
system("pause");
return 0;
}

```

### 3.3(a)-dastur. Output

```

n = 255
47 50 18 29 44 47 31 31 25 19 40 32 43 29 38 22 10 39 47 10 43 46 26
47 18 50 30 24 34 47 31 13 35 41 29 26 25 26 19 46 18 17 23 32 18 47
37 14 13 13 32 28 14 42 41 29 21 19 23 38 12 31 26 48 34 44 20 21 34
13 43 42 30 26 44 10 34 22 15 26 23 26 30 22 44 41 44 12 36 33 29 37
20 17 10 28 33 26 19 12 50 31 49 33 50 41 32 42 38 35 27 23 18 25 28
31 12 36 18 42 36 44 24 37 10 34 49 26 29 11 39 33 31 15 11 42 37 40
49 34 41 20 50 39 19 29 30 24 38 47 14 41 37 21 49 21 42 14 37 45 12
40 17 46 17 10 45 43 37 19 25 39 11 49 33 17 11 12 10 22 19 26 12 47
34 16 21 33 25 44 26 40 28 38 35 15 31 26 10 32 40 42 45 32 17 20 41
30 36 13 23 24 44 19 23 29 21 38 30 26 20 23 33 31 39 20 22 50 34 18
24 21 30 21 42 13 22 41 19 35 40 14 34 37 16 14 35 17 44 40 13 37 28
45 24
34 ta element topildi.

```

Oldinroqjudatez-  
tezelementlarnikiritishvao‘chirishalgoritmlarnifoydalanishkerakemasdeb  
aytgan. Vektorvato‘plambilanishlashsamaradorliginitaqqoslaylik.  
To‘plamvavektorningmassivlarinitasodifiysonlarbilanto‘ldiradiganvabel  
gilanganqiymatniqidiradigandasturyarataylik.

Vektorvato‘plambilanishlashsamaradorliginitaqqoslash.

```
// Created by MBBahodir

#include "stdafx.h"

#include <iostream>
#include <set>
#include <vector>
#include <algorithm>
#include <random>
#include <ctime>
#include <chrono>
using namespace std;

int main() {
    int n;
    default_random_engine rnd(time(0));
    uniform_int_distribution<unsigned> g(1, 10000);
    cout << "n = "; cin >> n;
    const int k = 100;
    int a = n;

    auto start = chrono::system_clock::now();
    vector<int> ar1;
    while (a--) ar1.push_back(g(rnd));
    auto result1 = find(ar1.begin(), ar1.end(), k);
    auto end = chrono::system_clock::now();
    auto elapsed = end - start;
    cout << "vector => "
<< elapsed.count()
<< endl;
    a = n;

    start = chrono::system_clock::now();
    set<int> ar2;
    while (a--) ar2.insert(g(rnd));
```

```

        auto result2 = ar2.find(k);
    end = chrono::system_clock::now();
    elapsed = end - start;
    cout << "set => " << elapsed.count() << endl;

    system("pause");
    return 0;
}

```

3.3(b)-dastur. Output
n = 10000 vector => 289989 set => 1160000

To‘plamning ishlashivektordansezilarlidarajadapastbo‘ladi. Shuniyoddatutingki, to‘plamni to‘ldirish amal isaralash bilan amalga oshiriladi vabualbatta, qimmatbaho amal. Lekin, faqat qidirish uchun, har ikki konteynerlarni ishlash solishtiradigan bo‘lsa, vaziyat <set> foydasiga o‘zgaradi:

3.3(c)-dastur. Output
n = 10000 vector => 70042 set => 20044

Bundanqanday xulosachiqarish mumkin? set yordamida (yoki multiset)

takrorlanish jarayonlarini dasturlashda ketma ketlikdanelementlarni olish, yokitez-tezo‘chirish va elementlarni kiritish bunoto‘g‘ri yondashuv hisoblanadi. Set to‘plam ob‘ektini yaratish uchun takrorlanish jarayonlari asosidashakllantiril ganketma ketlik foydalanish kerak yoki tayyor to‘plam uchun insert funksiyasidan foydalanish maqsadga muvofiq.

Ammo, dastur tez-tezelementlarni qidirish ishlatilsa, set raqobatgacha qo‘lmaydi.

pair - yordamchisi ni. Boshqa assotsiativ konteynerlar bilan ishlashni yanada ko‘ribchi qish uchun pair yordamchisi ni foblab tanishishimiz kerak. Businfi birlik sifatida ikki ta ob‘ektlarni saqlashimkoniyatini beradi. pair STD

ningturlijoylarida, xususan minmax() algoritmida, set equal\_range() sinfiningusulida, shuningdekjuft (key, value) bo'lganassosiativkonteynerlarelementlaribilanishlashuchunishlatiladi.

pair

birjuftob'ektyaratishuchunquyidagikonstruktorsintaktikiishlatiladi:

```
pair<T1, T2> p1;  
pair<T1, T2> p2(val1, val2);  
pair<T1, T2> p3(p1);
```

Amalari:

p.first – havolabo'yichajuftlikningbirinchielementigamurojaat.

p.second – havolabo'yichajuftlikningikkinchielementigamurojaat.

p->first – ko'rsatkichbo'yichajuftlikningbirinchielementigamurojaat.

p->second – ko'rsatkichbo'yichajuftlikningikkinchielementigamurojaat.

p = other\_p – qiymatqilibberish (C++ qoidasigaasosanoshkormastipalmatirishyordamida).

p.swap(other\_p) yoki swap(p, other\_p) –p va other\_p laruchuno'rinalmashtirish.

<, <=, >, >=, ==, != – taqqoslashamallari

make\_pair(val1, val2) – juftlikniberadi.

3.4-dastur. Pairkonstruktorivaamallariniishlatish.

```
// Created by MBBahodir  
#include "stdafx.h"  
  
#include <iostream>  
using namespace std;  
  
int main() {  
    int a1 = 16, b1 = 11;  
    int a2 = 18, b2 = 03;  
    pair<int, int> par1(a1, b1);  
    pair<int, int> *par2 = new pair<int, int>(a2,  
b2);  
    cout << par1.first << '.' << par1.second <<  
endl;  
    cout << par2->first << '.' << par2->second <<  
endl;  
    system("pause");  
    return 0;
```

```
}
```

### 3.4-dastur. Output

16.11

18.3

Assosiativkonreynerlaruchunmap va multimap sinflari. map assosiativkonteyneri (xarita, lug‘at) o‘z-o‘zini muvozanatlovchidaraxtga (qizil-qoradaraxt)

asoslanganto‘plamlar bilan bir xil tarzda amalga oshiriladi.

To‘plam valug‘ato‘rtasidagi asosiy farq shundaki, map massiv kalitlardan tashkil topgan elementlar juftlarining tartibli assosiativ massivi (konteyneri) va ularning mos qiymatlaridan iborat.

Lug‘atda kalitlari unikal (takrorlanmaydigan) va multimap bir nechta nusxadagi kalitlar ibo‘ladi.

Lug‘atlarda kalit va qiymat turlari fundamental tip yoki abstrakt tip bo‘lishi mumkin. Agarkalit tipini abstrakt tip bo‘lsa,

uning uchun saralash yo‘nalishini belgilovchi komparator funksiyasi (taqqoslash funksiyasi) aniqlanishi kerak.

Kalit doimiy qiymatga ega va kalit qiymatini o‘zgartirish mumkin emas.

Juftlikning ikkinchi element qiymatini (asosiy qiymatini) o‘zgartirish mumkin.

Map (yoki multimap) sinflari bilan ishlashni boshlash uchun dastursarlavhasiga (ikkalasini bilan birgalikda) quyidagi dastur fragmenti yozildi (bu oldin ham barcha assosiativ konteynerlar uchun aytilgan edi):

```
#include <map>
```

Konstruktorlar. Lug‘atob‘ektlar ishlablon parametrlarini: kalit tip va kalitning qiymat tipi (key va T), taqqoslash (comp) funksiyasini o‘z ichiga oladi. Agar bunday funksiya bo‘lmasa, uoshkormas less<> funksiya (operation <>) tomonidan o‘rnatiladi.

Quyidagi konstruktorlaryordamida map sinfi ob‘ektlarini yaratiladi:

Oddiy map konreyner konstruktorlari: map<Key, T, comp> ar; yoki map<Key, T, comp> ar(Comp);

Nusxalash uchun konstruktor: map<Key, T, comp> ar(other);

Iteratorlaryordamida qo‘shish uchun konstruktorlar: map<Key, T, comp> ar(first, last); yoki map<Key, T, comp> ar(first, last, Somp);

Ro‘yxat asosidagi initsializatsiya qilish konstruktorlari: map<Key, T, comp> ar {init}; yoki map<Key, T, comp> ar(init); yoki map<Key, T,



comp> ar(init, Comp); *Eslatma.*

Ro'yxatni initsializatsiya qilishda har bir juftlik alohida figurali qavs ichiga olinishi kerak!

```
map<string, int> ar { {"a1", 10}, {"www", 17}, {"j8", 100} };
```

Buyerda Comp konteyner kalitlarisolishtirish uchun funksiyasi (ixtiyoriy). Bundantashqari, taqqoslash funksiyaning yonida konstruktorixtiyoriy Allocator() vazifasini o'z ichiga oladi (soddaligi uchun yozilmaydi), dasturchiuning allocator vazifasini ham belgilaydi.

Map sinfining destruktori: ar.~map();

Lug'atusullari. Lug'atusullari set to'plam usullari bilan bir xil bo'lganligi uchun ularni takrorlab keltirmaymiz. Bu usullar qanday ishlashini misol yordamida ko'rsatamiz. Aytaylik, quyidagi lug'at yaratishimiz kerak: kalitgatasodifiy (string), qiymatgatasodifiy (integer) o'rnatiladi. So'ngra key1 va key2 orasidagi barcha key = > qiymat juftliklarining chiqish talab qilinadi. Dasturoxirida key 3 kalit uchun maksimal qiymatni chiqarish talab qilinadi (dasturning o'zidagi izohlarga qarang).

3.5-dastur. Lug'atusullaridan foydalanish.

```
// Created by MBBahodir
#include "stdafx.h"

#include <iostream>
#include <random>
#include <ctime>
#include <string>
#include <map>
#include <vector>

using namespace std;

int main() {
    map<string, int> rat;
    vector<string> lit;
    lit.push_back("a1"); lit.push_back("a2");
    lit.push_back("a3");
    lit.push_back("b1"); lit.push_back("b2");
    lit.push_back("b3");
    lit.push_back("c1"); lit.push_back("c2");
```

```

lit.push_back("c3");
    default_random_engine rnd(time(0));
    uniform_int_distribution<int> d(-10, 20);
    uniform_int_distribution<int> c(0, 8);

//=====1=====
=====//
    // lugʻatni toʻldirish: tasofidiy kalit tasodifiy
olinadi    //

//=====
=====//
    int n;
    cout << "Lugʻat elementlari sonini kirit: ";
    cin >> n;
    while (n--) {
        string S = lit[c(rnd)];
        int D = d(rnd);
        pair<string, int> tpr = make_pair(S, D);
        rat.insert(tpr);
        // emplace uchun juftlik yaratish shartmas:
        //rat.emplace(S, D);
    }

//=====2=====
=====//
    // 2 ta har xil amal bilan kalit va qiymat
qoshish    //

//=====
=====//
    rat.emplace("d1", 10);
    rat.insert(pair<const string, int>("d3", -15));
    rat.insert(pair<const string, int>("d3", 6));

//=====3=====
=====//
    // birinchi va ikkinchi kalit orasidagi

```

```

elementlarni chiqarish//

//=====
=====//
    string el_l, el_r;
    cout << "kalitlarni kirit:\n";
    cout << "1-kalit: "; cin >> el_l;
    cout << "2-kalit: "; cin >> el_r;
    auto fst = rat.lower_bound(el_l);
    auto lst = rat.upper_bound(el_r);
    while (fst != lst) {
        cout << fst -> first
<< " => "
<< fst -> second
<< endl;
        fst++;
    }

//=====4=====
=====//
    // kalitga mos eng katta elementni aniqlash
//

//=====
=====//
    string tmp;
    cout << "Kalit kirit: ";
    cin >> tmp;
    auto first = rat.equal_range(tmp).first;
    auto last = rat.equal_range(tmp).second;
    if (first == last)
        cout << 0 << endl;
    else
        while (first != last) {
            auto r = first -> second;
            if (r >= 10)
                cout << r << " ";
            first++;

```

```
}  
    system("pause");  
return 0;  
}
```

### 3.5-dastur. Birinchisinov Output

Lugʻat elementlari sonini kirit: 10

a2 => 14

a3 => 13

b2 => -8

b3 => 3

c2 => -4

c3 => -7

d1 => 10

d3 => -15

kalitlarni kirit:

1-kalit: a1

2-kalit: c2

a2 => 14

a3 => 13

b2 => -8

b3 => 3

c2 => -4

Kalit kirit: c4

0

### 3.5-dastur. Ikkinchisinov Output

Lugʻat elementlari sonini kirit: 9

a1 => 11

a2 => 10

b1 => 2

b2 => -10

c1 => 3

c2 => 9

d1 => 10

d3 => -15

kalitlarni kirit:

1-kalit: a1

2-kalit: c2

```
a1 => 11
a2 => 10
b1 => 2
b2 => -10
c1 => 3
c2 => 9
Kalit kirit: d3
```

### Elementlarga murojaat.

Lugʻatlarda elementlarga kirishning yana ikkita usul mavjud (iteratorlar bilan ishlashdantashqari). Birinchi usul `at(key)` usulini qoʻllashdir (`bilsangizkerak`). Biroq, indeksni argument sifatida qabul qilmaydi, lekin kalitning qiymatini qabul qiladi.

Bu usul kalitga ekvivalent boʻlgan kalit bilan mos elementni qiymatiga mos yoʻzlar qaytaradi. Agar bu element mavjud boʻlmasa, `out_of_range` istisno funksiyasining qiymati qaytariladi. Boshqacha aytganda, massivdabunday kalit bor-yoʻqligini tekshiradi. Ikkinchi usul `overloaded[]` funksiyadan foydalanishga asoslangan.

Lekin quyidagi sabablarga koʻra ehtiyojkorlik bilan foydalanish zarur.

`at(key)` mavjud boʻlmasa, standart konstruktoryordamida kalit bilan yangi element qator qoʻshiladi.

Aksholdaelementga koʻrsatkich qaytariladi. Eslatib oʻtish uchun, indekslash funksiyasini assotsiativ konteynerlarning ikkita turiga (`map` va `unordered_map`) foydalanish mumkin.

Ushbu usullarni ishlab chiquvchi lartomonidan qoʻshilishining sabablari aniq : dastur juda qisqavachiroylib oʻladi.

`at()` funksiyasi istisno qaytargani uchun `try - catch` funksiyasi bilan quyidagicha boshqarish mumkin:

```
map<string, int> rat;
try {
    rat.at("U");
}
// istesnoni ushlab olish
catch (std::out_of_range) {
    // uni qayta ishlash
    rat["U"] = 15;
}
```

Dasturfragmentida "U" kalitlijuftliktopilmasa, uniyaratib, yangiqiyamatberadi.

rat["U"] = 15; (hiyla, ruschafishka)  
indekslashamaliufaqatasosiyqiyamatiniolishemas,  
balkibuqiyamatnio‘zgartirishmumkindeganma‘nonianglatadi,  
birqiyamatiniqaytaradi. Boshqachaqilibaytganda,  
ushbudasturfragmenidagiamallarini bajarishmumkin:

```
rat["b1"]=1980;  
cout << rat["b1"] << endl;  
rat["b1"] +=1;  
cout << rat["b1"] << endl;  
--rat["b1"];  
cout << rat["b1"] << endl;
```

map<K, T, C = less<K>, A = allocator<pair<const K, T>>> to‘plam  
pair<const K, T>tipidagiqiyamatnisaqlaydi. Bunda K kalit,  
tanlashimkoniniberadiva T saqlanadiganqiyamat.  
Shuninguchuniteratorandanfoydalangannimizda first kalitqiyamatniva  
second saqlanganqiyamatniqaytaradi, o‘zgartirishimkoniniberadi.  
Yuqoridaaytib o‘tilgandek operator[] asosiyamallardanhisoblanadi  
(vazifasinibilsangizkerak). Agar map<K, T> m bo‘lsava m[k] = t  
ma‘noji hatidan quyidagi fragment gateng:

```
m.insert(make_pair(k, T())).first->second = t
```

Bukabielementlarniqo‘shishsamarali hisoblanib, T  
ob’ektniyaratishgaimkonbermaydi (yuqoridagi fragmentda aniq T()  
konstruktorchaqirilgan).

Agar operator[] noqulayvasamarasizdebhisoblasangiz,  
yanaungaikkita alternativvariantlarbor. Birinchisi find  
funksiyasidanfoydalanib, kalitbo‘yicha (key, value)  
juftligamositeratorniolishyokikalitlug‘atdabo‘lmasa, end()  
funksiyadanfoydalanishmumkin. Ikkinchisi at() funksiyasidan,  
ya‘nikalitasosidaqiyamatniqaytaradi. Agarkalitbo‘lmasa,  
istisnoholatgao‘tadi.

Map sinfiningumumiyshabloniquyidagicha:

```
template <class Key, class T, class Compare = less<Key>,  
         template <class U> class Allocator = allocator>
```

Map sinfiningoperatorlari, xususiyatlari,  
funksiyalarivausullariniqo‘ribchiqamiz.

Typedef operatorlari :

```
typedef Key key_type;
```

```

typedef pair<const Key, T> value_type;
typedef Compare key_compare;
class value_compare
    : public binary_function<value_type, value_type, bool>
{
    friend class map;
protected:
    Compare comp;
    value_compare(Compare c) : comp(c) {}
public:
    bool operator()(const value_type& x, const value_type&
y) {
        return comp(x.first, y.first);
    }
};
typedef iterator;
typedef const_iterator;
typedef Allocator<value_type>::pointer pointer;
typedef Allocator<value_type>::reference reference;
typedef Allocator<value_type>::const_reference
const_reference;
typedef size_type;
typedef difference_type;
typedef reverse_iterator;
typedef const_reverse_iterator;

```

Xotiraniajratishvabo‘shatish (xolliqilish) operatorlari  
(allocation/deallocation):

```

map(const Compare& comp = Compare());
template <class InputIterator>
map(InputIterator first, InputIterator last,
    const Compare& comp = Compare());
map(const map<Key, T, Compare, Allocator>& x);
~map();
map<Key, T, Compare, Allocator>&
    operator=(const map<Key, T, Compare, Allocator>& x);
void swap(map<Key, T, Compare, Allocator>& x);

```

Ruxsatetishvositalari (accessors):

```

key_compare key_comp() const;
value_compare value_comp() const;

```

```

iterator begin()
const_iterator begin() const;
iterator end();
const_iterator end() const;
reverse_iterator rbegin();
const_reverse_iterator rbegin();
reverse_iterator rend();
const_reverse_iterator rend();
bool empty() const;
size_type size() const;
size_type max_size() const;
Allocator<T>::reference operator[](const key_type& x);

```

Qo'shishvao'chirishfunksiyalar (insert/erase):

```

pair<iterator, bool> insert(const value_type& x);
iterator insert(iterator position, const value_type& x);
template <class InputIterator>
void insert(InputIterator first, InputIterator last);
void erase(iterator position);
size_type erase(const key_type& x);
void erase(iterator first, iterator last);

```

Mapsinfusulari:

```

iterator find(const key_type& x);
const_iterator find(const key_type& x) const;
size_type count(const key_type& x) const;
iterator lower_bound(const key_type& x);
const_iterator lower_bound(const key_type& x) const;
iterator upper_bound(const key_type& x);
const_iterator upper_bound(const key_type& x) const;
pair<iterator, iterator> equal_range(const key_type& x);
pair<const_iterator, const_iterator> equal_range(const key_type&
x)const;

```

Mantiqiyoperatorlari:

```

template <class Key, class T, class Compare, class Allocator>
bool operator==(const map<Key, T, Compare, Allocator>& x,
                const map<Key, T, Compare, Allocator>& y);

template <class Key, class T, class Compare, class Allocator>
bool operator<(const mapr<Key, T, Compare, Allocator>& x,

```



```
const map<Key, T, Compare, Allocator>& y);
```

iterator- ikkitomonlama iterator bo'lib value\_type ishoraberadi. Aniqtipiamalgaoshirishbilanbog'liqva Allocator asosidabelgilanadi.

const\_iterator- doimiyikkitomonlama iterator bo'lib const value\_type ishoraberadi. Aniqtipiamalgaoshirishbilanbog'liqva Allocator asosidabelgilanadi. Bunda iterator va const\_iterator uchunkonstruktorbor, bukafolatlanadi.

size\_type – butunishorasiztip. Aniqtipiamalgaoshirishbilanbog'liqva Allocator asosidabelgilanadi.

difference\_type - butunishoralitip. Aniqtipiamalgaoshirishbilanbog'liqva Allocator asosidabelgilanadi.

Assotsiativkonteynerlaruchunstandartusullarto'plamidantashqari, lug'atAllocator::reference operator[](const key\_type&)amalinita'minlaydi. Masalan, m lug'atda k kalituyulan m[k] kuyidagifragmentbilanekvmvalent:

```
(*((m.insert(make_pair(k, T()))).first)).second
```

Assotsiativkonteynerlarasosiyxususiyatlari

Headers		<set>		<map>	
Members		<u>set</u>	<u>multiset</u>	<u>map</u>	<u>multimap</u>
	constructo r	<u>set</u>	<u>multiset</u>	<u>map</u>	<u>multimap</u>
	destructor	<u>~set</u>	<u>~multise t</u>	<u>~map</u>	<u>~multima p</u>
	assignment	<u>operator =</u>	<u>operator =</u>	<u>operator =</u>	<u>operator =</u>
iterator s	begin	<u>begin</u>	<u>begin</u>	<u>begin</u>	<u>begin</u>
	end	<u>end</u>	<u>end</u>	<u>end</u>	<u>end</u>
	rbegin	<u>rbegin</u>	<u>rbegin</u>	<u>rbegin</u>	<u>rbegin</u>
	rend	<u>rend</u>	<u>rend</u>	<u>rend</u>	<u>rend</u>
const iterator s	cbegin	<u>cbegin</u>	<u>cbegin</u>	<u>cbegin</u>	<u>cbegin</u>
	cend	<u>cend</u>	<u>cend</u>	<u>cend</u>	<u>cend</u>
	crbegin	<u>crbegin</u>	<u>crbegin</u>	<u>crbegin</u>	<u>crbegin</u>
	crend	<u>crend</u>	<u>crend</u>	<u>crend</u>	<u>crend</u>
capacity	size	<u>size</u>	<u>size</u>	<u>size</u>	<u>size</u>
	max_size	<u>max size</u>	<u>max size</u>	<u>max size</u>	<u>max size</u>
	empty	<u>empty</u>	<u>empty</u>	<u>empty</u>	

Tartiblangan unikalitga ega bo'lmagan map – multimap sinfi hisoblanadi.

Businfi operator[] standart operatorni qo'llab quvvatlamaydi, va multiset juftligini eslatadi. Bunda qidirish faqat birinchi maydon, ya'ni kalit maydon bilan amalga oshiriladi. multimap sinfini o'rganib chiqish uchun, misol sifatida matnli faylga yozilgan lug'atga murojaat mallari niko'rsatamiz.

Faraz qilaylik lug'at so'zlar juftligidan tashkil topgan va birinchi so'z kalit bo'lsin va takrorlanishi mumkin.

“so'z - tarjima” tipini aniqlash.

Bunda so'z juftligini o'lamiz va lug'at element deb qaraymiz.

```
using Entry = pair<string, string>;
```

Lug'at tipini aniqlash

```
using Dictionary = multimap<string, string>;
```

Standart kirish va chiqish operatorlari quyidagicha aniqlaymiz.

```
namespace std
{
    ostream& operator>>(ostream &is, Entry &en)
    {
        return is >> en.first >> en.second;
    }

    ostream& operator<<(ostream &os, const Entry &en)
    {
        return os << en.first << ' ' << en.second;
    }

    ostream& operator>>(ostream &is, Dictionary &d)
    {
        ostream_iterator<Entry> begin(is), end;
        d = Dictionary(begin, end);
        return is;
    }

    ostream& operator<<(ostream &os, const Dictionary &d)
    {
        ostream_iterator<Entry> out(os, "\n");
```

```
    copy(d.begin(), d.end(), out);  
return os;  
}  
}
```

Lugʻatgakirishuchunamal.  
Belgilanganlugʻatuchunyangilugʻatchiqaradi.

```
Dictionary reverse(const Dictionary &d)  
{  
    Dictionary result;  
    for (const auto &el : d)  
        result.emplace(el.second, el.first);  
return result;  
}
```

Lugʻatelementigamurojaatvaoʻqish.

```
void dictionary_reverse(istream &from, ostream &to)  
{  
    Dictionary read;  
    from >> read;  
    to << reverse(read);  
}
```

Std sinfidaaniqlanganistream\_iterator va ostream\_iterator tiplaridanfoydalanganlikuchun [<<] va [>>] operatorlarishufazoagamoslabyaratilgan.

Bufoydalanuvchigakirishvachiqishoperatorlarivastandardtiplardan foydalanishruhiyatigataʻsirqilmaydi.

*Multimarsinfningkonstruktori map sinfinikibilanbirxil.*

*Multimarsinfning map sinfinikidanfarfqiladiganoperatorlari:*

```
friend class multimap;  
protected:  
    Compare comp;  
    value_compare(Compare c) : comp(c) {}  
public:  
    bool operator()(const value_type& x, const value_type&  
y) {  
        return comp(x.first, y.first);  
    }  
};
```

Mantiqiyoperatorlari:

```
template <class Key, class T, class Compare, class Allocator>
```

```
bool operator==(const multimap<Key, T, Compare, Allocator>& x,  
                const multimap<Key, T, Compare, Allocator>& y);
```

```
template <class Key, class T, class Compare, class Allocator>  
bool operator<(const multimap<Key, T, Compare, Allocator>& x,  
              const multimap<Key, T, Compare, Allocator>& y);
```

iterator- ikkitomonlama iterator bo'lib value\_type ishoraberadi. Aniqtipiamalgaoshirishbilanbog'liqva Allocator asosilabelgilanadi.

const\_iterator- doimiyikkitomonlama iterator bo'lib const value\_type ishoraberadi. Aniqtipiamalgaoshirishbilanbog'liqva Allocator asosidabelgilanadi. Bunda iterator va const\_iterator uchunkonstruktorbor, bukafolatlanadi.

size\_type – butunishorasiztip. Aniqtipiamalgaoshirishbilanbog'liqva Allocator asosidabelgilanadi.

difference\_type - butunishoralitip. Aniqtipiamalgaoshirishbilanbog'liqva Allocator asosidabelgilanadi.

**Tartiblanmaganassosiativkonteynerlar<unordered\_set>va <unordered\_map>**.Tartiblanmagankonteynerlar<unordered\_set>, <unordered\_map>xesh – jadvalasosidaqurilgan (odatdabuxeshjadvallarekvivalentelementlarro'yxati - bloklar) vaxeshfunksiyalarga (joriyholatidastandarttiplaruchun<functional>sinfidan std::hash<T> bilananiqlanadi) vatenglikamaliga (joriyholatda [=]= operatori) tayanadi.

Ikkitaelementuchunxeshlaritengbo'lsavatenglikamalichin (true) qiymatqaytarsa, ularekvivalenthisoblanadi. Tartiblanmagankonteynerlaribirtomolamaiteratorlarbilanelementlarigam urojaatqilishkerak.

Tartiblanmaganstandartassosiativkonteynerlargaquyidagishablonsi nflarkiradi:

```
unordered_set<K, H = hash<K>, E = equal_to<K>, A =  
allocator<K>>,  
unordered_map<K, T, H = hash<K>, E = equal_to<K>, A =  
allocator<pair<const K, T>>>,  
unordered_multiset,  
unordered_multimap.
```

Bularningfunksionalligitartiblanganassosiativkonteynerlarga o'xshaydi.

Tartiblanmagankonteynerlardaxeshjadvalnishakllantirishningo‘zig axosligilower\_bound va upper\_bound funksiyalarigaegamasligidir. Ammo, berilgankalitgamosbirintervaldagielementlarniolishuchun equal\_range funksiyasiishlatiladi.

O‘rtachatariblanmagankonteynerlardaelemetnlarniqo‘shish, qidirishvao‘chirisho‘rtachadoimiyvaqttalabvatartiblanganlarnisanisbatanse zilarlidarajadatezroqbo‘lishimumkin. Biroq,

engyomonholatdahamvaqtgachiziqlierishishimumkin.

Samaradorligiheshfunksiyasiniamalgaoshirishsifatigabog‘liq, lekinbutunlaybundayvaqtajratishnioldiniolishimumkinemas.

Shuninguchun, interaktivdasturlarda (masalan, o‘yinlar), tartibgasolinmagankonteynerlardavriykechikishlartufaylitartiblanganlard anko‘rayomonroqbo‘lishimumkin.

Kattasondagielementlarqo‘shishuchun, keraklivaqtdabirqatorkiritishda "rehash"gaoshirishimumkin (ko‘pvaqtoladiganajratilgansaqlashvaqaytatarqatishelementlarhajminios hirish) rehash funksiyasinichaqiribto‘g‘rihisoblanadi (vektoruchun reserve funksiyasigao‘xshaydi).

Ayrimvazifalardasaralashuchunvaqtqo‘pajratilsaavtomatiksaralash nimuhimkamchilikdebhisoblashimumkin. Buholda, tartiblanganto‘plamvalug‘atlarsinflarinitartiblanmaganassotsiativkonteyn erlarsinflar (unordered\_set, unordered\_multiset, unordered\_map va unordered\_multimap) tomonidano‘zgartirilishimumkin.

Tartiblanmagankonteynerlardaqidirish, qo‘shishvao‘chirishamallario‘rtachadoimiyvaqtmurakkabligigaegavauO (1) gatengdir.

Tartiblanmagankonteynerlarxeshjadvallarsifatidaamalgaoshiriladi. Xesh-jadvaldaamal nibajarishxesh-funksiyada (xeshlash) kalitdanboshlanadi. Olinganxeshqiymati (shuningdek, xeshyokixeshkodi) massivdaindeksrolinibajaradi. Xeshjadvalioddiylug‘atgao‘xshaydi, undatomma‘nodaxeshqiymatidebhisoblashimumkin.

Tartiblanmagankonteynerlarningxeshjadvallaridayacheykalariborv aularcheksizko‘pelementlarnio‘zichigaolishimumkin. Buyacheeykani segment debataymiz. Aslida, segment (xeshbilanbog‘liq) ulanganro‘yxatdir.

To‘plamhajmiganisbatansaqlanadiganelementlarsonimoskelsa, (xeshfunksiyasiuchunimumkinbo‘lganelementlarsoni) xeshjadvalnito‘ldirish (load factor)

koefitsientidebyurituladivao‘rtachaamalijrovaqtbelgilaydiganmuhimparametrhisoblanadi.

`unordered_set` va `unordered_multiset` sinflaribilanishlashniboshlashuchun `unordered_set` sarlavhasini (ikkalasinfbilanbirgalikda) quyidagichaqo‘shishkerak:

```
#include <unordered_set>
```

`unordered_map` va `unordered_multimap` sinflarihamxuddiyuqorida gidekqo‘shiladi:

```
#include <unordered_map>
```

Tartiblanmagankonteynerlarningtiplariquyidagishablonparametrlar inio‘zichigaoladi:

- `const Key` – kalit tipi (`unordered_set`, `unordered_multiset`, `unordered_map`, `unordered_multimap`).

- `T` – qiymatning tipi (`unordered_map`, `unordered_multimap`).

- `alloc` – alloktorfunksiya (majburiyemas, erkin).

- `size_type` `bucket_count` – initsializatsiyaqilishdafoydalanishuchunminimallyacheykalarsoni (agarko‘rsatilmaganbo‘lsa, joriyholatbo‘yichaolinadi).

- `hash` – xesh-funksiya (erkin).

- `equal` – taqqoslashfueksiyasi, kalitlarnisolishtirishuchunishlatiladi (erkin).

Quyidagikonstruktorlarasosida`unordered_set`va `unordered_map`snifl ariningob’ektiniquradi:

```
unordered_set<Key> ar;  
unordered_set<Key, hash, equal> ar;  
unordered_map<Key, T> ar;  
unordered_map<Key, T, hash, equal> ar;
```

Agar `hash` xeshfunksiyayozilmaganbo‘lsa, joriyholatbo‘yicha `hash<key_type>`funksiyasiishlatiladi.

Agarequal taqqoslashfunksiyasiyozilmaganbo‘lsa, joriyholatbo‘yicha `equal_to<>` (operator==) funksiyasiishlatiladi.

Soddaligiuchun `Allocator ()` va `bucket_count` yozilmaydi.

Nusxalashkonstruktrlari:

```
unordered_set<Key> ar(other);  
unordered_map<Key, T> ar(other);
```

Iteratorlarasosidaqo‘shish:

```
unordered_set<Key> ar(first, last);  
unordered_set<Key, hash, equal> ar(first, last);  
unordered_map<Key, T> ar(first, last);
```

```
unordered_map<Key, T, hash, equal> ar(first, last);
```

**Ro'yxatbo'yichainitsializatsiyaqilish:**

```
unordered_set<Key> ar {init};
```

```
unordered_set<Key> ar(init);
```

```
unordered_set<Key, hash, equal> ar(init);
```

```
unordered_map<Key, T> ar {init};
```

```
unordered_map<Key, T> ar(init);
```

```
unordered_map<Key, T, hash, equal> ar(init);
```

**unordered\_set va unordered\_map**

**sinfob'ektlarinio'chirishuchundestruktorlar:**

```
ar.~unordered_set();
```

```
ar.~unordered_map();
```

**Tartiblanmagankonteynerlardaislatilmaydiganusullar:**

- [ $>$ ], [ $<$ ], [ $\geq$ ], [ $\leq$ ] taqqoslashamallari ( [ $==$ ] va [ $!=$ ] amallarniqo'llabquvvatlaydi)
- lower\_bound va upper\_bound
- Ikkitomonlamaiteratorlar: rbegin, crbegin, rend, crend

Tartiblanmagankonteynerlarningusullari:

**Xeshlashuslubi**

(Buusullartartibsizkonteynerlarningishlashiniboshqarishimkoniniberadi):

-load\_factor() – to'ldirishkoeffitsentiniqaytaradi (yuqorigaqarang)

-max\_load\_factor() –

Agarargumentsizfoydalanilsa, maksimalto'ldirishqiymatiuchunfloat tipiniqaytaradi, agarargumentsifatida float tipifoydalanilsa, maksimalto'ldirishqiymatiuchuno'rnatiladi.

Agarterminatoromiliushbuchegaradanoshsa,

konteyneravtomatikravishdasegmentlarsoninioshirishimumkin.

-rehash(size\_type count) – konteyneruchunqaytaxeshlasho'tkazadi, ya'niyacheykalarsonini (bucket\_count) bucket\_count  $\geq$  count va bucket\_count  $>$  size/max\_load\_factor bo'lishiuchun.

-reserve(size\_type count) – songayacheykalarsonnio'rnatadi, konteynerniqaytaxeshlasho'kazmaslikvamaksimalto'ldirishkoeffitsientiu chun, oxirgiyacheykaningcount hisoblanadi.

Maksimalsamaradorlikkaerishishuchunhardoimmaksimalyacheykalarsoninianiqlashkerak. Tavsiyaetilganqiymatlar 0.7 dan 0.8 gachaoraliqdabo'ladi.

Elementqidirishuchunstandartmaksimalyacheykalarsonio'rnatiladi ( 1.0 gateng) vadastuochilartomonidan 0.7 o'rnatiladi.

Xeshfunksiyasining o'z konteyner faoliyatini yaxshilash mumkin, ammo bu mavzudoi rasidan chiqib ketishga olib keladi.

Interfeys segmentlari.

Segment elementlari bilan bevosita ishlash uchun

(bog'langan ro'yxat sifatida) quyidagi usullardan foydalaniladi:

• `begin()`, `end()`, `cbegin()`, `cend()` – iteratorlar. E'tibor bering, bular segmentlar uchun, massiv uchun emas!

Konteynerlar uchun shularning analoglari yaratilgan.

• `size_type bucket_count()` – massivdagi segmentlar sonini qaytaradi.

• `size_type max_bucket_count()` –

muumkin bo'lgan maksimal elementlar soni

(tizing avajoriy qilishga bog'liq).

• `size_type bucket_size(size_type n)` –

ni indeksli segmentdagi elementlar soni.

• `size_type bucket(Key)` – kalitasosid segment sonini qaytaradi.

Bu usullar asosida xesh jadvalni vizual ko'rinishini tasavvur qilish mumkin.

### 3.6-dastur. Tartiblanmagan konteynerlardan foydalanish.

```
#include "stdafx.h"

#include <iostream>
#include <iomanip>
#include <unordered_set>
#include <random>
#include <ctime>
#include <chrono>
using namespace std;

int main() {
    default_random_engine rnd(time(0));
    uniform_int_distribution<unsigned> g(100, 999);
    unordered_multiset<int> unm;
    size_t n = 60;
    unm.reserve(n);
    while (n--) unm.emplace(g(rnd));
    cout << "Konteynerdagi segmentlar soni: "
```



```

<< unm.bucket_count()
<< "\n"
<< "Maksimal to'ldirishlar ko'ffitsenti: "
<< unm.max_load_factor()
<< endl;
    auto start = chrono::system_clock::now();
    auto res = unm.find(200);
    auto end = chrono::system_clock::now();
    auto elapsed = end - start;
    cout << "Natija => "
<< elapsed.count()
<< endl;
    // O'zimiz tomondan Maksimal to'ldirishlar
ko'ffitsenti o'rnatamiz
    unm.max_load_factor(0.7);
    unm.rehash(200);
    cout << "Konteynerdagi segmentlar soni: "
<< unm.bucket_count()
<< "\n"
<< "Maksimal to'ldirishlar ko'ffitsenti: "
<< unm.max_load_factor()
<< endl;
    start = chrono::system_clock::now();
    res = unm.find(200);
    end = chrono::system_clock::now();
    elapsed = end - start;
    cout << "Natija => "
<< elapsed.count()
<< endl;
    cout << " Hash tuzilmasi " << endl;
    for (size_t i = 0; i < unm.bucket_count(); i++)
    {
        cout << "Segment N:" << setw(3) << i << "
=> ";
        // Iteratorga lokal segmentlarni olish
        auto first = unm.cbegin(i);
        auto last = unm.cend(i);
        while (first != last) {

```

```

        cout << *first++ << " ";
    }
    cout << endl;
}
system("pause");
return 0;
}

```

3.6-dastur. Output

```

Konteynerdagi segmentlar soni: 32
Maksimal to'ldirishlar ko'ffitsenti: 1
Natija => 30023
Konteynerdagi segmentlar soni: 256
Maksimal to'ldirishlar ko'ffitsenti: 0.7
Natija => 10004
Segment N: 0 =>          Segment N: 1 =>
Segment N: 2 =>          Segment N: 3 =>
Segment N: 4 =>
Segment N: 5 =>          Segment N: 6 =>
Segment N: 7 =>          Segment N: 8 =>
Segment N: 9 => 646
Segment N: 10 =>         Segment N: 11 =>
Segment N: 12 =>         Segment N: 13 =>
Segment N: 14 =>
Segment N: 15 =>         Segment N: 16 =>
Segment N: 17 =>         Segment N: 18 =>
Segment N: 19 =>
Segment N: 20 =>         Segment N: 21 =>
Segment N: 22 =>         Segment N: 23 =>
Segment N: 24 => 940
Segment N: 25 =>         Segment N: 26 =>
Segment N: 27 =>         Segment N: 28 =>
Segment N: 29 =>
Segment N: 30 => 717    Segment N: 31 =>
Segment N: 32 =>         Segment N: 33 =>
Segment N: 34 =>
Segment N: 35 =>         Segment N: 36 =>
Segment N: 37 =>         Segment N: 38 =>
Segment N: 39 =>

```

Segment N: 40 =>	Segment N: 41 =>
Segment N: 42 =>	Segment N: 43 =>
Segment N: 44 =>	
Segment N: 45 => 841 738	
Segment N: 46 =>	Segment N: 47 =>
Segment N: 48 =>	Segment N: 49 =>
Segment N: 50 =>	Segment N: 51 =>
Segment N: 52 => 768	Segment N: 53 =>
Segment N: 54 =>	Segment N: 55 =>
Segment N: 56 =>	Segment N: 57 =>
Segment N: 58 =>	Segment N: 59 =>
Segment N: 60 => 631	Segment N: 61 =>
Segment N: 62 => 316	Segment N: 63 =>
Segment N: 64 =>	Segment N: 65 =>
Segment N: 66 =>	Segment N: 67 =>
Segment N: 68 =>	Segment N: 69 =>
Segment N: 70 =>	Segment N: 71 =>
Segment N: 72 =>	Segment N: 73 =>
Segment N: 74 =>	Segment N: 75 =>
Segment N: 76 =>	Segment N: 77 =>
Segment N: 78 =>	Segment N: 79 =>
Segment N: 80 =>	Segment N: 81 =>
Segment N: 82 =>	Segment N: 83 =>
Segment N: 84 =>	Segment N: 85 =>
Segment N: 86 =>	Segment N: 87 =>
Segment N: 88 =>	Segment N: 89 =>
Segment N: 90 =>	Segment N: 91 =>
Segment N: 92 => 498	Segment N: 93 =>
Segment N: 94 => 187 476	
Segment N: 95 =>	Segment N: 96 =>
Segment N: 97 =>	Segment N: 98 => 280
Segment N: 99 =>	Segment N:100 => 225
Segment N:101 =>	Segment N:102 =>
Segment N:103 =>	Segment N:104 =>
Segment N:105 => 973	Segment N:106 =>
Segment N:107 =>	Segment N:108 =>
Segment N:109 =>	Segment N:110 =>
Segment N:111 =>	Segment N:112 =>Segment

N:113 =>		Segment N:114 =>		Segment
N:115 =>		Segment N:116 =>	960	Segment
N:117 =>		Segment N:118 =>		Segment
N:119 =>		Segment N:120 =>		Segment
N:121 =>	630	Segment N:122 =>		Segment
N:123 =>		Segment N:124 =>	567	Segment
N:125 =>		Segment N:126 =>		Segment
N:127 =>		Segment N:128 =>		Segment
N:129 =>		Segment N:130 =>		Segment
N:131 =>		Segment N:132 =>		Segment
N:133 =>	351	Segment N:134 =>	227	Segment
N:135 =>		Segment N:136 =>		Segment
N:137 =>		Segment N:138 =>		Segment
N:139 =>		Segment N:140 =>		Segment
N:141 =>		Segment N:142 =>		Segment
N:143 =>		Segment N:144 =>		Segment
N:145 =>		Segment N:146 =>	761	Segment
N:147 =>		Segment N:148 =>	145	Segment
N:149 =>		Segment N:150 =>		Segment
N:151 =>		Segment N:152 =>		Segment
N:153 =>		Segment N:154 =>		Segment
N:155 =>		Segment N:156 =>		Segment
N:157 =>		Segment N:158 =>		Segment
N:159 =>		Segment N:160 =>		Segment
N:161 =>		Segment N:162 =>		Segment
N:163 =>		Segment N:164 =>		Segment
N:165 =>		Segment N:166 =>		Segment
N:167 =>	242	Segment N:168 =>		Segment
N:169 =>		Segment N:170 =>		Segment
N:171 =>		Segment N:172 =>		Segment
N:173 =>		Segment N:174 =>		Segment
N:175 =>		Segment N:176 =>		Segment
N:177 =>		Segment N:178 =>		Segment
N:179 =>		Segment N:180 =>	458	Segment
N:181 =>		Segment N:182 =>		Segment
N:183 =>		Segment N:184 =>		Segment
N:185 =>		Segment N:186 =>		Segment
N:187 =>		Segment N:188 =>		Segment N:189 =>

Segment N:190 =>	Segment N:191 =>
Segment N:192 => 852	Segment N:193 =>
Segment N:194 =>	Segment N:195 =>
Segment N:196 =>	Segment N:197 => 287
Segment N:198 =>	Segment N:199 =>
Segment N:200 =>	Segment N:201 =>
Segment N:202 =>	Segment N:203 =>
Segment N:204 =>	Segment N:205 =>
Segment N:206 =>	Segment N:207 =>
Segment N:208 =>	Segment N:209 => 483
Segment N:210 => 697	Segment N:211 =>
Segment N:212 =>	Segment N:213 =>
Segment N:214 =>	Segment N:215 => 967
Segment N:216 =>	Segment N:217 =>
Segment N:218 =>	Segment N:219 =>
Segment N:220 =>	Segment N:221 =>
Segment N:222 =>	Segment N:223 =>
Segment N:224 =>	Segment N:225 =>
Segment N:226 =>	Segment N:227 =>
Segment N:228 =>	Segment N:229 => 666
Segment N:230 => 484	Segment N:231 =>
Segment N:232 => 603	Segment N:233 =>
Segment N:234 =>	Segment N:235 => 254
Segment N:236 =>	Segment N:237 =>
Segment N:238 =>	Segment N:239 =>
Segment N:240 =>	Segment N:241 =>
Segment N:242 =>	Segment N:243 =>
Segment N:244 =>	Segment N:245 =>
Segment N:246 =>	Segment N:247 =>
Segment N:248 =>	Segment N:249 =>
Segment N:250 =>	Segment N:251 =>
Segment N:252 =>	Segment N:253 =>
Segment N:254 =>	Segment N:255 =>

Tartiblanmagankonteynerlarning sinflarimurakkabtuzilishgaega, ammostandartvositalar bilan ham tartiblangankonteynerlardankambo‘l magan mukammalishlashniko‘rsatadi.

Tartiblangankonteynerlaryokitartiblanmagankonteynerlarorasidagitanlov vazifagabog‘liqbo‘ladi.

Assotsiativ konteynerlarning asosiy usullari keng tarqalganligi sababli, dasturning bir bajarilishidan boshqasiga o'tish oson va testlarni o'tkazganingizdan so'ng tanlovingizni bir yoki bir nechta turlik konteynerda amalga oshirib, tanlov qilishingiz mumkin.

Tartiblanmagankonteynerlarning sinflari xususiyatlari.

Headers		<unordered_set>		<unordered_map>	
Members		unordered_set	unordered_multiset	unordered_map	unordered_multimap
	<i>constructor</i>	unordered_set	unordered_multiset	unordered_map	unordered_multimap
	<i>destructor</i>	~unordered_set	~unordered_multiset	~unordered_map	~unordered_multimap
	<i>assignment</i>	operator=	operator=	operator=	operator=
iterators	begin	<u>begin</u>	begin	begin	begin
	end	end	end	end	end
	rbegin				
	rend				
const iterators	cbegin	cbegin	cbegin	cbegin	cbegin
	cend	cend	cend	cend	cend
	crbegin				
	crend				
capacity	size	size	size	size	size
	max_size	max_size	max_size	max_size	max_size
	empty				

## NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1. Assotsiativ konteynerlarga qaysi sinflar kiradi?

2. C++

barcha assotsiativ konteynerlar qanday amallarni qo'llab-quvvatlaydi?

3. Key1 va key2 kalitlar o'zaro teng hisoblanadi,

agartaq qoslanadigan ob'ektlar uchun qaysi funksiyachin qiymat qaytaradi?

4. Qanday konteynerlar uchun aniqlanadigan tip va kalit tipini bir xil bo'ladi?

5. Assosiativkonteynerlarning iteratorlarine chatomonlama iteratorlarsini fikiradi?
6. Kalitasosida qidirishdaberilgan kichik bo'lmagan birinchi elementni qanday funktsiya va birinchi kattaelementni qanday funktsiya yordamida amalga oshiriladi?
7. Set  
to'plamda qiymatni o'zgartirish uchun avvalnima funktsiya ni bajarib, so'ng yangi qiymatni yoki o'zgartirilgan variantni qo'shiladi?
8. Xotirani ajratish va bo'shatish operatorlari set sinfi uchun sanab bering?
9. Ishoralib tutun tip, aniq tipni belgilash realizatsiya qilishga bog'liq va Allocator da aniqlanadi. Gap qaysi funktsiya haqida.
10. To'plam - assosiativ konteyner to'plam, teng qiymatli kalitlarni saqlaydi (mumkin qadar birkalit qiymatli elementlarni to'plamini saqlaydi) va kalitni qalitez qidirishimkonini beradi. Gap qaysi to'plam haqida bormoqda.
11. set va multiset sinflarining ob'ektlarining tipikali bilan yonma-yon bittashablonli parametrolishim mumkin. Bushablon qaysi funktsiya?
12. [=] (operator=) operatorni nima uchun ishlatiladi?
13. To'plam vektorning massivlarini tasodifiy sonlar bilan to'ldiradigan va belgilangan qiymatni qidiradigan dasturtuzing.
14. Pair qanday sinfni nima uchun kerak?
15. p.first - havolabo'yicha juftlikning nechanchi elementiga murojaat?
16. Lug'atda kalitlari unikal (takrorlanmaydigan) va qaysi sinfdan iborat nusxada kalitlari bo'ladi.
17. Oddiy map konreiner konstruktorlari: map<Key, T, comp> ar; va map<Key, T, comp> ar(Comp); ni ishlatish tartibini tushuntirib bering.
18. Ro'yxatni initsializatsiya qilishda har bir juftlik alohida qanday qavs ichiga olinishi kerak.
19. Lug'atlarda elementlarga kirishning yana ikkita usulimavjud (iteratorlar bilan ishlatishdantashqari). Birinchi va ikkinchi usullarni tushuntirib bering.
20. Indeksash funktsiyasini assosiativ konteyner sinflarining qaysi sinflariga foydalanishim mumkin.
21. at() funktsiyasi istisno qaytargani uchun try - catch funktsiyasi bilan boshqarishim mumkinmi?

22. operator[] noqulayvasamarasizdebhisoblasangiz, yanaungaikkitaalternativvariantlarbor. Buvariantlarniayting.
23. Qaysito‘plamdaqidirishfaqatbirinchimaydon, ya‘nikalitmaydonbilanamalgaoshiriladi.
24. Qandaykonteynerlarxesh – jadvalasosidaqurilganvaxeshfunksiyalargavatenglikamaligatayanadi.
25. Ayrimvazifalardasaralashuchunvaqtko‘pajratilsaavtomatiksaralashnimuhimkamchilikdebhisoblashmumkin. Bundadasturchiqandayyo‘ltutadi.
26. Tartiblanmagankonteynerlarningxeshjadvallaridanimaborvaularcheksizko‘pelementlarnio‘zichigaolishimumkin.
27. Tartiblanmagankonteynerlarningtiplariuchunshablonparametrlarinisanabbering.
28. Tartiblanmagankonteynerlardaagarequal taqqoslashfunksiyasiyozilmaganbo‘lsa, joriyholatbo‘yichaqaysifunksiyaishlatiladi.
29. Xeshlashuslubinimauchunkerak.
30. rehash(size\_type count) – konteyneruchunqaytaxeshlasho‘tkazadi, ya‘niyacheykalarinisoniniqaysishartdabo‘lishiuchun.
31. Segmentelementlaribilanbevositaishlashuchunqandayusullardanfoydalaniladi.
32. Tartiblangankonteynerlaryokitartiblanmagankonteynerlarorasidagitanlovnimagabog‘liqbo‘ladi.



**AMALIY KO‘NIKMA VA MALAKALARNI ANIQLASH  
HAMDA RIVOJLANTIRISH UCHUN ASSISMENT  
TOPSHIRIQLARI.**

<b>BIRINCHI ASSISMENT TOPSHIRIG‘I</b>	
	Assosiativkonteynerlarga (set, multiset) oidberilganquyidagidasturbo‘yichaberilgantopshiriqlarkeraklif ragmentlariasosidabajaring. Bundadasturdagiba‘zio‘zgartirishlarnitopishorqalitopshiriqlar bosqichma – bosqichamalganoshiriladi.
dastur	topshiriqlar



<pre>// Created by MBBahodir #include "stdafx.h" #include &lt;iostream&gt; #include &lt;set&gt; using namespace std;  void myPrint(set&lt;int&gt;&amp;my set){      for     (set&lt;int&gt;::iterator it=myset.begin(); it!=myset.end(); ++it)         cout &lt;&lt; ' ' &lt;&lt; *it;     cout &lt;&lt; '\n'; } void myPrintmulti(multis et&lt;int&gt;&amp;myset){      for     (multiset&lt;int&gt;::ite rator it=myset.begin(); it!=myset.end(); ++it)         cout &lt;&lt; ' ' &lt;&lt; *it;     cout &lt;&lt; '\n';</pre>	<p>1. Dasturdanecht global funksiyalar yaratilgan.</p> <hr/> <hr/> <hr/>
	<p>2. Dasturdanecht set to'plam mavjud.</p> <hr/> <hr/> <hr/>
	<p>3. Dasturdanecht multiset to'plam mavjud.</p> <hr/> <hr/> <hr/> <hr/>
	<p>4. Set va multiset to'plamlarning farqida sturning qay sif fragmentida ko'rsatilgan .</p> <hr/> <hr/>
	<p>5. Nima uchun oxirgi myPrint(myset); funksiyasi ishlamaydi.</p> <hr/> <hr/>
	<p>6. reverse_iterator ni o'rniga iterator ni o'rnatsa, nimabo'ladi. Xatobo'ladi. Xatoni tuzatish uchun nima qilish kerak.</p> <hr/> <hr/>

<pre> } void mySizeSets(set&lt;int&gt; &amp;myset){     cout &lt;&lt; int (myset.size()) &lt;&lt; '\n'; }  int main () {     set&lt;int&gt; myset;  set&lt;int&gt;::iterator it;      int myints[]={ 12,82,37,64,15,15 };      set&lt;int&gt; first (myints,myints+6);     multiset&lt;int&gt; mfirst (myints,myints+6);     set&lt;int&gt; second;     second = first;      myPrint(first);  myPrintmulti(mfirst );  mySizeSets(second);  mySizeSets(first); </pre>	<p>7. myset.swap(second);nio‘rnigasecond.swap( first); yozilsanimahodisasodirbo‘ladi.</p> <hr/> <hr/> <p>8. Set vamultisetningoperator=dasturningqaysiqis mlaridafoydalanilganvanimauchun.</p> <hr/> <hr/>
--	--

```

    for (int i=1;
i<=5; i++)
myset.insert(i*10);

    myPrint(myset);

it=myset.find(20);
    myset.erase (it);
    myset.erase
(myset.find(40));

myset.swap(second);

    for (
set<int>::reverse_i
terator
rit=myset.rbegin();
rit !=
myset.rend();
++rit)
        std::cout << '
' << *rit;
        cout << '\n';

    while
(!myset.empty())
    {
        cout << ' ' <<
*myset.begin();

myset.erase(myset.b
egin());
    }

    myPrint(myset);

```


<pre>system("pause"); return 0; }</pre>	
<p>9. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____</p> <p>10. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.</p>	

<b>IKKINCHI ASSISMENT TOPSHIRIG‘I</b>	
	<p>Assosiativkonteynerlarga (map, multimap) oidberilganquyidagidasturbo‘yichaberilgantopshiriqlar kerakli fragmentlari asosida bajarang.</p> <p>☝ Bundasturdagiba‘zior‘zgartirishlarnitopishorqalitopshiriqlar bosqichma — bosqich amalga oshiriladi.</p>
dastur	topshiriqlar
<pre>// Created by MBBahodir #include "stdafx.h"  #include &lt;iostream&gt; #include &lt;map&gt; #include &lt;string&gt;  using namespace std; void myPrint(map&lt;char,int&gt;&amp;mymap){     for (map&lt;char,int&gt;::iterator it=mymap.begin(); it!=mymap.end(); ++it)     cout &lt;&lt; it-&gt;first &lt;&lt; " =&gt; " &lt;&lt; it-&gt;second &lt;&lt; endl;     cout &lt;&lt; endl; } void myPrintmulti(multimap&lt;s</pre>	<p>1. Dasturdanecht global funksiyay aratilgan.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <p>2. Dasturdanecht map va multimap to‘plam mavjud.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>

<pre> tring,int&gt;&amp;myMultimap){     for (auto&amp; x : myMultimap)         cout &lt;&lt; " [" &lt;&lt; x.first &lt;&lt; ':' &lt;&lt; x.second &lt;&lt; ']';         cout &lt;&lt; endl; } int main () {     map&lt;char,int&gt; mymap, mymap_one;  map&lt;char,int&gt;::iterator it;  pair&lt;map&lt;char,int&gt;::ite rator,map&lt;char,int&gt;::it erator&gt; ret;      mymap['a']=10; mymap_one['x'] = 100;     mymap['b']=20; mymap_one['y'] = 200;     mymap['c']=30; mymap_one['z'] = 300;  multimap&lt;std::string,in t&gt; mymultimap;  mymultimap.emplace("Ali ",1.50); </pre>	<p>_____</p>
	<p>3. mapvavmultimap to‘plamlarningfarqidasturningqaysif ragmentidako‘rsatilgan .</p> <p>_____</p> <p>_____</p>
	<p>4. Qachonoxirgi myPrint(mymap);funksiyaishlamay di.</p> <p>_____</p> <p>_____</p> <p>_____</p>
	<p>5. mymultimapnio‘rnigamapnio‘r natsa, nimabo‘ladi. Xatobo‘ladi. Xatonituzatishuchunnimaqilishkera k.</p> <p>_____</p> <p>_____</p> <p>_____</p>
	<p>6.mymap.swap(mymap_one) nio‘rnigamymap_one.swap(mymap )yozilsanimahodisasodirbo‘ladi.</p> <p>_____</p> <p>_____</p> <p>_____</p>

<pre> mymultimap.emplace("Val i",2.71);  mymultimap.emplace("Sol i",1.40);  mymultimap.emplace("Sod iq",3.14);      it = mymap.find('b');     if (it != mymap.end()) mymap.erase (it);      myPrint(mymap);     myPrint(mymap_one);  /* while (!mymap.empty()) {     std::cout &lt;&lt; mymap.begin()-&gt;first &lt;&lt; " =&gt; " &lt;&lt; mymap.begin()-&gt;second &lt;&lt; '\n';  mymap.erase(mymap.begin ()); } */ </pre>	<p>7. map vamultimapningoperator=dasturning qaysiqismlaridafoydalanilganvanima uchun.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
<pre> mymap.swap(mymap_one);      myPrint(mymap_one);     myPrint(mymap); </pre>	<p>8. Dasturdagiizohgaolingandastur fragmentiolibtashlansa, dasturdaqandayo‘zgarishlarbo‘ladiva natijadachi.</p> <hr/> <hr/> <hr/> <hr/> <hr/>

<pre> ret = mymap.equal_range('b');  cout &lt;&lt; ret.first- &gt;first &lt;&lt; " =&gt; " &lt;&lt; ret.first-&gt;second &lt;&lt; endl; cout &lt;&lt; ret.second- &gt;first &lt;&lt; " =&gt; " &lt;&lt; ret.second-&gt;second &lt;&lt; endl;  myPrintmulti(mymultimap );  system("pause"); return 0; } </pre>	
<p>9. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____</p> <p>10. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.</p>	

<p align="center"><b>UCHINCHI ASSISMENT TOPSHIRIG‘I</b></p>	
	<p>Tartiblanmaganassosiativkonteynerlarga (unordered_set, unordered_multiset) oidberilganquyidagidasturbo‘yichaberilgantopshiriqlar keraklifragmentlariasosidabajaring.</p> <p>👉 Bundaydasturdagiba‘zio‘zgartirishlarnitopishorqalito pshiriqlarbosqichma – bosqichamalganoshriladi.</p>
<p align="center">dastur</p>	<p align="center">topshiriqlar</p>
<pre> // Created by MBBahodir #include "stdafx.h" #include &lt;iostream&gt; #include &lt;string&gt; </pre>	<p>1. Dasturdanechtaglobalfunksiyayaratilgan.</p> <hr/> <hr/> <hr/>

<pre> #include &lt;unordered_set&gt;  using namespace std;  template&lt;class T&gt; T cmerge (T a, T b) {     T t(a);     t.insert(b.begin(),b .end()); return t; } void myPrint(unordered_se t&lt;string&gt;&amp;myset){     for (const string&amp; x: myset) cout &lt;&lt; " " &lt;&lt; x;     cout &lt;&lt; endl; } void myPrintmultiset(unor dered_multiset&lt;strin g&gt;&amp;mymultiset){     for (const std::string&amp; x: mymultiset) std::cout &lt;&lt; " " &lt;&lt; x;     std::cout &lt;&lt; std::endl; } int main () { </pre>	<p>2. Dasturdanechta unordered_set to'plammavjud.</p> <hr/> <hr/> <hr/> <hr/>
	<p>3. Dasturdanechta unordered_multiset to'plammavjud.</p> <hr/> <hr/> <hr/> <hr/>
	<p>4. DasturdamyPrint(myset_one)funksiy asiyaqirilgandanatijaqandaybo'ladi vanim asababdan.</p> <hr/> <hr/>
	<p>5. To'plamdanberilganqiymatniizlashd asturfragmentinianiqlangvamymset_oneuch unhamizlashfunksiyasiniyarating.</p> <hr/> <hr/> <hr/> <hr/>
	<p>6. myset.rehash(12)nimavazifanibajara di.</p> <hr/> <hr/>



<pre>unordered_set&lt;string&gt; myset, myset_str;     unordered_set&lt;int&gt; myset_one;</pre>	<p>7. myset_str.swap(myset); nio'rni gamyset.swap(myset_str); yozilsanimahodisasodirbo'ladi.</p> <hr/> <hr/>
<pre>unordered_multiset&lt;string&gt; first, second, third;      myset.emplace("Laseti"); myset_one.insert(1);     myset.emplace("Tico"); myset_one.insert(15); ;</pre>	<p>8. operator= dasturningqaysiqismlaridafoydalanilganva nimauchun.</p> <hr/> <hr/> <hr/>
<pre>    myset.emplace("Damas"); myset_one.insert(45); ;</pre> <p>myPrint(myset);</p>	<p>9. myPrintmultiset(second); funksiyasie kranganimanatijanichiqaradi.</p> <hr/> <hr/> <hr/>
<pre>    for (const int&amp; x: myset_one) cout &lt;&lt; " " &lt;&lt; x;         cout &lt;&lt; endl;</pre> <p>string input;     cout &lt;&lt; "Mashina nomini kirit: ";     getline(cin, input);</p> <pre>unordered_set&lt;::string&gt;::const_iterator got =</pre>	<p>10. third = cmerge (first, second); funksiyasinimavazifanibajaradivanatijasiq anday.</p> <hr/> <hr/> <hr/>

```

myset.find(input);

    if ( got ==
myset.end() )
        cout <<
"to'plamda yo'q";
    else
        cout << *got <<
" nomli mashina
bor";
    cout << endl;

    myset.rehash(12);

myset.insert("mashin
a");

myset.insert("uy");

myset.insert("daraxt
");

myset.insert("eshik"
);

myset.insert("choyna
k");

    cout << "joriy
bucket_count: " <<
myset.bucket_count()
<< endl;

myset_str.swap(myset
);

```

```
    cout << "first
=>";
    myPrint(myset);

    cout << "second
=>";

myPrint(myset_str);

first.insert("www.uz
");
first.insert("tuit.u
z");
first.insert("edu.uz
");

second.insert("www.u
z");
second.insert("uzedu
.uz");
second.insert("edu.u
z");
    third = cmerge
(first, second);
    first = third;

    std::cout <<
"first =>";

myPrintmultiset(firs
t);

    second.clear();

second.insert("bed")
;
```

```

second.insert("bed")
;

second.insert("wardr
obe");

second.insert("night
stand");

    cout << "second
=>";

myPrintmultiset(seco
nd);

    string mystring =
"red";

    second.insert
(mystring);
    second.insert
(mystring+"dish");

    cout << "second
=>";

myPrintmultiset(seco
nd);

    system("pause");
return 0;
}

```

11. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. \_\_\_\_\_
12. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.

## TO‘RTINCHI ASSISMENT TOPSHIRIG‘I



Tartiblanmaganassosiativkonteynerlarga  
(unotdered\_map, unordered\_multimap)

	<p>oidberilgan quyidagi dastur bo'yicha berilgan topshiriqlar kerakli fragmentlari asosida bajarilgani.</p> <p>☝ Bundagi dasturda giba 'zio' zgartirishlarni topish orqali topshiriqlar bosqichma bosqich amalga oshiriladi.</p>
dastur	topshiriqlar
<pre>// Created by MBBahodir #include "stdafx.h" #include &lt;iostream&gt; #include &lt;string&gt; #include &lt;unordered_map&gt;  using namespace std;  template&lt;class T&gt; T cmerge (T a, T b) {     T t(a);     t.insert(b.begin(),b.en d()); return t; }  void myPrint(unordered_map&lt;i nt,string&gt;&amp;mymap){     for (auto&amp; x: mymap) cout &lt;&lt; "[" &lt;&lt; x.first &lt;&lt; "," &lt;&lt; x.second &lt;&lt; "]" ;     cout &lt;&lt; endl; }  void myPrintmultimap(unorder ed_multimap&lt;int,string&gt; &amp;mymultimap){     for (auto&amp; x: mymultimap) cout &lt;&lt; "["</pre>	<p>1. Dasturda nechta global funksiya yaratilgan.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <p>2. Dasturda nechta unordered_map to'plam mavjud.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <p>3. Dasturda nechta unordered_multimap to'plam mavjud.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>

<pre> &lt;&lt; x.first &lt;&lt; ", " &lt;&lt; x.second &lt;&lt; "]" ;     cout &lt;&lt; endl; } int main () {  unordered_map&lt;int, string&gt; mymap, mymap_str;  unordered_map&lt;int, int&gt; mymap_one;  unordered_multimap&lt;int, string&gt; first, second, third;      mymap.emplace (80686, "Laseti"); mymap_one.insert(make_p air&lt;int, int&gt;(1, 2));     mymap.emplace (10010, "Tico"); mymap_one.insert(make_p air&lt;int, int&gt;(15, 16));     mymap.emplace (70707, "Damas"); mymap_one.insert(make_p air&lt;int, int&gt;(45, 46));      myPrint(mymap);      for (auto x: mymap_one) cout &lt;&lt; "[" &lt;&lt; x.first &lt;&lt; ", " &lt;&lt; x.second &lt;&lt; "]" ; </pre>	<p>4. DasturdamyPrint(mymap_one) funksiyasi yaqirilgandanatijaqanday bo‘ladi vanimasababdan.</p> <hr/> <hr/> <hr/> <p>5. To‘plamdanberilganqiymatniiz lashdasturfragmentinianiqlangvamy map_oneuchunhamizlashfunksiyasi niyarating.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <p>6. mymap.rehash(20)nimavazifan ibajaradi.</p> <hr/> <hr/> <hr/> <p>7. mymap_str.swap(mymap); nio‘rnigamymap.swap(mymap_str); yozilsanimahodisasodirbo‘ladi.</p> <hr/> <hr/> <hr/>
--	--

<pre> cout &lt;&lt; endl;  int input; cout &lt;&lt; "Mashina raqamini kirit: "; cin &gt;&gt; input;  unordered_map&lt;int, string&gt;::const_iterator got = mymap.find(input);  if ( got == mymap.end() )     cout &lt;&lt; "to'plamda yo'q"; else     cout &lt;&lt; "[" &lt;&lt; got-&gt;first &lt;&lt; ", " &lt;&lt; got-&gt;second &lt;&lt; "]" &lt;&lt; " raqamli mashina bor"; cout &lt;&lt; endl;  mymap.rehash(20);  mymap[1] = "mashina"; mymap[10] = "uy"; mymap[15] = "daraxt"; mymap[20] = "eshik"; mymap[25] = "choynak";  cout &lt;&lt; "joriy bucket_count: " &lt;&lt; mymap.bucket_count() &lt;&lt; endl; </pre>	<p>8. operator= dasturning qaysiqismlarida foydalanilgan vanima uchun.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
<pre> if ( got == mymap.end() )     cout &lt;&lt; "to'plamda yo'q"; else     cout &lt;&lt; "[" &lt;&lt; got-&gt;first &lt;&lt; ", " &lt;&lt; got-&gt;second &lt;&lt; "]" &lt;&lt; " raqamli mashina bor"; cout &lt;&lt; endl;  mymap.rehash(20);  mymap[1] = "mashina"; mymap[10] = "uy"; mymap[15] = "daraxt"; mymap[20] = "eshik"; mymap[25] = "choynak";  cout &lt;&lt; "joriy bucket_count: " &lt;&lt; mymap.bucket_count() &lt;&lt; endl; </pre>	<p>9. myPrintmultimap(second); funksiyasi ekranga nimanat ijanichiqaradi .</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
<pre> cout &lt;&lt; "joriy bucket_count: " &lt;&lt; mymap.bucket_count() &lt;&lt; endl; </pre>	<p>10. third = cmerge (first, second); funksiyasini mavzifanibajaradiv anati ijasiqanday.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>

```

mymap_str.swap(mymap);

    cout << "first =>";
    myPrint(mymap);

    cout << "second =>";
    myPrint(mymap_str);

    pair<int,string>
mypair (1,"www.uz");
    pair<int,string>
mypair_one
(2,"tuit.uz");

    first.insert
(mypair);

second.insert(mypair_on
e);
    first.insert
(pair<int,string>(50,"G
00G"));

second.insert(pair<int,
string>(501,"GGOD"));
    first.insert
(second.begin(),
second.end());

    third = cmerge
(first, second);
    first = third;

    cout << "first =>";
myPrintmultimap(first);


    second.clear();

```



<pre> second.insert(pair&lt;int, string&gt;(80, "BUXORO"));  second.insert(pair&lt;int, string&gt;(01, "TOSHKENT")) ;  second.insert(pair&lt;int, string&gt;(70, "QARSHI"));  second.insert(make_pair (75, "TERMIZ"));      cout &lt;&lt; "second =&gt;";  myPrintmultimap(second) ;      system("pause"); return 0; } </pre>	
<p>11. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____</p> <p>12. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.</p>	

### 1.4. Konteynerlarning adapterlari.

 Adapterdarning muhim tushunchalari asosida tek, navbat, ustivornavbat, ikki tomonlama navbat sifatlar va ularning xususiyatlari, funksiyalari, dasturlashda o‘zlarini tutishlari, konteynerlar bilan ishlashga mo‘ljallangan asosiy algoritmlar va ularning dasturlashdagi ahamiyati, funkto funktsiyalarini yaratish va foydalanish usullari, amallari, talablari, vazifalarini va usullarini keltirilgan bo‘lib, dasturchining adapterlar, algoritmlar va funkto funktsiyalar bilan ishlashdagi tanlashimkoniyati bo‘yicha tajribaxulosalari, nazariy bilimlarni asoslash uchun dasturlar tuzib ko‘rsatilgan.

Bilimlarni mustahkamlash uchun

tanazariysavolvaamaliyko‘nikmavamalakalarnirivojlantrishuchun 5 taassimenttopshirig‘ivaharassimentda 10(9,7) tatopshiriq, jami 46tatopshiriqberilgan.

Butopshiriqlarnibajarishmavzunimustahkamlashuchunxizmatqiladi.

**✍Kalitso‘zlar.** Konteyner, adapter, shablon, sinf, moslashtirish, stack (stek), queue (navbat), priority\_queue (ustuvorbilannavbat), deque (ikkitomonlamanavbat), ma’lumotlartuzilmasi, push\_back, pop\_back, pop\_front, LIFO, FILO, LIFO, for\_each(), find(), funksionalob’ektlar, Operator(), Standart ML vafunktor, Haskell vafunktor.

**✔Bilishshartbo‘lgantushunchalar.** Tiptushunchasi, sinfvasinfob’ekti, to‘plam, statikvadinamikmassiv, elementgamurojaat, funksiyavako‘rsatkich, konteyner, ketma-ketkonteynerlar, xesh, tasodifiy, istisnoholat, ma’lumotlarnikirishvachiqishi, oqim, dasturlashgaoiddastlabkitushunchalarvaC++tiliniqo‘llabquvvatlovchimu hitdaishlashnibilishlozim.

**☞Bilibolasiz.** stack (stek), queue (navbat), priority\_queue (ustuvorbilannavbat), deque (ikkitomonlamanavbat) sinfob’ektlarivaularningxususiyatlari, funksiyalari, dasturlashdao‘zlarinitutishlari (LIFO, FILO, LIFO), konteynerlarbilanishlashgamo‘ljallanganasosiyalgoritmlar (for\_each(), find()) vaularningdasturlashdagiahamiyati, funktofunktsiyalarniyaratishvafoydalanishuslublari, amallari, talablari, vazifalarivausullarinio‘rganishingiz mumkin.

## REJA

1. Stack, queue, priority\_queue, deque.
2. Konteynerlarbilanishlashalgoritmlari.
3. Funktorlarningqo‘llanilishi.

## KIRISH

Adapterlarstandartishablonkutubxonasingalohidatoifasidir (vakilidir). Adapterlaryangitushunchalaryokiilovalaremas, balkimavjudkutubxonatushunchalariningo‘zigaxos, tez-tezishlatiladiganmaqсадlaruchunmoslashtirishlari. Bumoslashtirish (adaptatsiyalash) ko‘pinchaadapterningso‘rovlarnnigasosiytushunchasigaasoslanib, funksiyalarinicheklashorqaliamalgaoshiriladi. Kutubxonakonteynerlarningadapterlari, iteratorlarvafunksiyalarnio‘zichigaoladi.

**Konteynerlarning adapterlari.** Konteynerlarning adapterlari asosida adapterlarni hosil qilishning eng soʻy oʻlimi solorqalitushtirishdir. Bular stack (stek), queue (navbat), priority\_queue (ustuvor bilan navbat) misol boʻladi. Buning uchun quyidagilarni koʻr: adapterlar:

- juda keng vatez-tez ishlatiladigan maʼlumotlarni tuzilmalari;
- har qanday realizatsiya alohida bajarilishi kerak emas.

Ular funktsiyalarini taʼminlash uchun STL standart konteyneri har qanday moslashtirish datayanch sifatida push\_back, pop\_back, yoki pop\_front amallardan (adapter turiga qarab) foydalanish mumkin;

- adapter uchun datayanch konteynerning toʻplamidan qoʻshimcha amallar chiqarib tashlanishi kerak (vahimlari yaratmaslik uchun, masalan, indekslash amali uchun, konteyner vektor boʻlsa, stek moslashtirish ishlatiladi);

Konteynerlarning adapterlari uchun sintaktik taʼriflarni misollar orqali qisqartirib koʻrsatish mumkin, shuning uchun misollar bilan tushuntirib oʻtish.

Birinchi navbatda konteynerlarning adapterlari uchun `<stack>`, `<queue>` kabir flarni dasturga qoʻshish lozim.

Stek (stack, maʼlumotlari yigʻindisi): stekda uning elementlari gafaqat bir uchidan murojaat qilish mumkinligi bilan xarakterlanadi va stekning yuqori qismida bataladi. Bu LIFO (Last In — First Out, birinchi kirgan oxir chiqadi) tamoyili boʻyicha faoliyat koʻrsatuvchi maʼlumotlarni taʼminlaydigan.

Stekka oid deyarli barcha funktsiyalarini koʻrsatadigan dastur keltirilgan (4.1-dastur).

4.1-dastur. Stekka oid funktsiyalardan foydalanish.

```
// Created by MBBahodir
#include "stdafx.h"

#include <iostream>
#include <vector>
#include <stack>
#include <string>
#include <array>
using namespace std;

int main() {
```

```

stack<int> st, st_one, st_two;
stack<string,vector<string>> vst;

vst.emplace("str1");
vst.emplace("str2");
vst.push("eostr");

    for (int i=0; i<5; ++i) st.push(i);

    for (int i = 0; i < 15; i++)
st_one.emplace(i*10);

    st_two = st;
    st.swap(st_one);

    cout << st.size() << " " << st_one.size() << " "
<< st_two.size() << " " << vst.size() << endl;

    while (!vst.empty()) {
        cout << vst.top() << " : stack dagi o'rni " <<
vst.size() << endl;
        vst.pop();
    }

    int size = st.size();
    while (size--) {
        cout << st.top() << " : stack dagi o'rni " <<
st.size() << endl;
        st.pop();
    }

    while (st.size() != 0){
        cout << st_two.top() << " : stack dagi o'rni "
<< st_two.size() << endl;
        st_two.pop();
    }

```

```

while (!st.empty()) {
    cout << st.top() << " : stack dagi oʻrni " <<
st.size() << endl;
    st.pop();
}

cout << st.size() << " " << st_one.size() << " "
<< st_two.size() << " " << vst.size() << endl;

cout << " stack st " << (st.empty() ? "" : "not ")
<< " empty" << endl;
cout << " stack vst" << (vst.empty() ? "" : "not
") << " empty" << endl;
cout << " stack st_one " << (st_one.empty() ? "" :
"not ") << " empty" << endl;
system("pause");
return 0;
}

```

#### 4.1-dastur. Output.

```

15 5 5 3
eostr : stack dagi oʻrni 3
str2 : stack dagi oʻrni 2
str1 : stack dagi oʻrni 1
140 : stack dagi oʻrni 15
130 : stack dagi oʻrni 14
120 : stack dagi oʻrni 13
110 : stack dagi oʻrni 12
100 : stack dagi oʻrni 11
90 : stack dagi oʻrni 10
80 : stack dagi oʻrni 9
70 : stack dagi oʻrni 8
60 : stack dagi oʻrni 7
50 : stack dagi oʻrni 6
40 : stack dagi oʻrni 5
30 : stack dagi oʻrni 4
20 : stack dagi oʻrni 3
10 : stack dagi oʻrni 2
0 : stack dagi oʻrni 1

```

0 5 5 0

```
stack st empty
stack vst empty
stack st_one not empty
```

#### 4.1

dasturdakeltirilganstekadapteriningfunksiyalaribo'yichadasturnitahlilqili shorqaliba'zixulosalarkeltiramiz:

- Stack<string>elementlarisatrbo'lgano'zgaruvchinie'lonqiladi. String ob'ektlaro'zlari STL konteynerlarekanligigae'tiborbering. Shundayqilib, stekkonteynerlarningharqandaye elementlarinio'zichigaolgankonteynerlar niolishimumkin (buboshqa STL konteynerlariuchunxosdir).

- stack<string>e'lonqilishnimisollarningko'pchiligidastekkatavsifsifatidak o'rishmumkin.

Ko'pchilikdasturchibuniboshqabo'lishibilmasligivatassafurqilmasligimu mkin. Lekinbixilta'rifmavjud stack<string, vector<string>>satrlaribirsteki, bundatayanchsinflars ifatidafoydalanish. Ba'zanso'rashdi, nimauchun stack<vektor<string>>yozishmumkinemas (birinchi stringnio'chirganholda)? Chunki stack konstruktoriniamalgaoshirishdabuyo'ibelgilanganvabujudamumkin.

Ammo, butunlayboshqaturdagitavsifbo'ladi: string vektorlarito'plami stack<vektor<string>>umumanboshqastekbo'ladi (konteynerlarningtarkibiyjoylashtirishhaqidagieslatma).

- Judako'pholardasteknivektorko'rinishidainitsializatsiyaqilinda, ammobufaqatC++11variantidamavjud. 4.1-dasturdaemplace() funksiyasidanstekkaelementqo'shishuchunfoydalanilgan.

- Keyingifragmentlarda, deyarlibarchaamallarni (usullarini) ko'rishmumkin: stack:push() – stekkabittaelementqo'shish, top() – stekningengyuqoridaturganelementigako'rsatkicholish, pop() – yuqorielementinichiqaribtashlash, size() – stekningjoriyhajmi, empty() – stekningbo'shliginitekshirish, swap() – ikkistekningalmashtirish, operatop= stekniqiymatqilibboshqabirtekkaberish (o'zlashtirish).

- Dasturdanko'rishmumkinki, stekadapteritayanchkonteynergaxosusullarniyo'qotdi (at(), operator[]

vahokazo.), lekin push(), pop() funksiyalarniqaytaaniqlashorqaliorttirilganhisoblanadi.

Steknitushinibolib, o'xshashfunksiyalarninavbatgajoriyqilishqiyinmasalaemas. Stekdanfarqlio'laroq, navbat FIFO (First In — First Out birinchikirganbirinchichiqadi) tamoyilibo'yichafaoliyatyuritadiganma'lumotlarto'plamidir. (Buusulnibiruchigaoqibkirish, keyinboshqauchidanoqibchiqadiganquvurgao'xshatishmumkin).

4.1-

dasturdatilsemantikasitalabibo'yichatahrirlarniqilibnavbatuchundeyarlio'zgarishsizqoldiramizva 4.2 dasturniolamiz.

4.2-dastur. Navbatsinfiningfunksiyalari.

```
// Created by MBBahodir
#include "stdafx.h"

#include <iostream>
#include <list>
#include <queue>
#include <string>
#include <array>
using namespace std;

int main() {

    queue<int> st, st_one, st_two;
    queue<string, list<string>> vst;

    vst.emplace("str1");
    vst.emplace("str2");
    vst.push("eostr");

    for (int i=0; i<5; ++i) st.push(i);

    for (int i = 0; i < 15; i++)
st_one.emplace(i*10);

    st_two = st;
    st.swap(st_one);
```

```

    cout << st.size() << " " << st_one.size() << " "
<< st_two.size() << " " << vst.size() << endl;

    while (!vst.empty()) {
        cout << vst.front() << " : stack dagi o'rni "
<< vst.size() << endl;
        vst.pop();
    }

    int size = st.size();
    while (size--) {
        cout << st.front() << " : stack dagi o'rni "
<< st.size() << endl;
        st.pop();
    }

    while (st.size() != 0){
        cout << st_two.front() << " : stack dagi o'rni
" << st_two.size() << endl;
        st_two.pop();
    }

    while (!st.empty()) {
        cout << st.front() << " : stack dagi o'rni "
<< st.size() << endl;
        st.pop();
    }

    cout << st.size() << " " << st_one.size() << " "
<< st_two.size() << " " << vst.size() << endl;

    cout << " stack st " << (st.empty() ? "" : "not ")
<< " empty" << endl;
    cout << " stack vst" << (vst.empty() ? "" : "not
") << " empty" << endl;
    cout << " stack st_one " << (st_one.empty() ? "" :
"not ") << " empty" << endl;

```



```

system("pause");
return 0;
}

```

#### 4.2-dastur.Output

```

15 5 5 3
str1 : stack dagi o'ni 3
str2 : stack dagi o'ni 2
eostr : stack dagi o'ni 1
0 : stack dagi o'ni 15
10 : stack dagi o'ni 14
20 : stack dagi o'ni 13
30 : stack dagi o'ni 12
40 : stack dagi o'ni 11
50 : stack dagi o'ni 10
60 : stack dagi o'ni 9
70 : stack dagi o'ni 8
80 : stack dagi o'ni 7
90 : stack dagi o'ni 6
100 : stack dagi o'ni 5
110 : stack dagi o'ni 4
120 : stack dagi o'ni 3
130 : stack dagi o'ni 2
140 : stack dagi o'ni 1
0 5 5 0
stack st empty
stack vst empty
stack st_one not empty

```

4.1-dasturda o'zgartirish talab qilinganlarini o'zgartirib, quyidagixulosalarga olib keladi:

- navbatda `pop_front()` usuli vektor konteyner ustiga qurib bo'lmaydi, lekin uni ro'yxat konteyner uchun qurish mumkin.

- Deque sinfi uchun `front()`, `push_back()`, `pop_front()`, usullari har qanday konteynerlarda ishlatiladi (faqat `pop_front()` usuliborlarida).

- shuning uchun navbat uchun yasala digan ob'ektda, ya'ni deque sinfida `top()` usuli o'rniga `front()` usuli ishlatiladi.

Ma'lum birtartibda qayta ishlanish kerak ma'lumotlarni uchun umumiy ma'lumotlarni tuzilishiga asoslangan o'plamibu – stek navbatidir (stack,

deque). Masalan, biror funksiya o'z navbatida uchinchifunksiyanicha qiradigan boshqa funksiya anicha qirsa, uchinchifunksiya birinchifunksiyanimas, ikkinchifunksiyaga qaytish muhimdir.

Bu ma'lumotlarni qayta ishlashtartibini amalga oshirish uchun dasturchi o'zining navbat funksiyasini tashkil qilish lozim. Stekka qo'shilgan oxirgi qiymat birinchi bajariladi va aksincha, stekka qo'shilgan birinchi qiymat oxirgi bajariladi. Shunday qilib, ma'lumotlarni tuzilmasining o'zicha qirishlarning to'g'ri belgilangan tartibdab erilishini ta'minlaydi.

Konseptual, ma'lumotlarni tuzilishi - stek juda oddiy: u ma'lumotlarni kiritish yoki chiqarish uchun ma'lum birtartibni belgilaydi. Har safar birelement qo'shiladi, buni stekka yuqori element bo'lib tugaydi. Stekdan olib tashlanadigan yagona element stekning yuqori qismida joylashgan elementdir. Shunday qilib, stekka "birinchi kirish, oxirgi chiqish — FILO" yoki "oxirgi kirish, birinchi chiqish — LIFO" deb ham aytiladi. Demak, stekka qo'shilgan birinchi element undan oxirgi chiqariladi.

Xo'sh, buni ma'lumotlarni tuzilishi Nima uchun stek kerak? Yuqorida aytilganidek, funksiyalarni (qiymatlarni) chaqirishni tashkil qilish uchun qulay yo'lidir. Aslida, stek chaqirish (call stack) tez-tez ishlaydigan yoki boshqa funksiyalarning qaytish qiymatini kutayotgan funksiyalar o'z xatiga murojaat qilish uchun ishlatiladigan atamadir.

Shuningdek, stek informatika fanining fundamental tilining bir qismidir. Agar birinchi kelib, oxirish ketish tamoyili asosida fikrlasangiz, unda stek terminini ishlatish lozim.

Bundantashqari, bunday (stek kabi) navbatlarni o'zaro bajarishda ishlatiladi, nazariy informatika dasturida, bunday push-down funksiyalarivajudaham ko'p bag usullarida.

Stek assosiativ usullarga:

Push – stekka element qo'shish..

Pop – stekdan element o'chirish

Top – elementni ko'rish.

LIFO - stek xarakati.

FILO - stek xarakati (ekvivalent LIFO ga).

Stack

(stek)

Ma'lumotlarni tuzilmasini amalga oshirish masalani qaraymiz.

Umumlashgan dasturlash usuliga mos keladigan umumiy stack, ya'ni Stack sinfi uchun shablondan yaratish degani.

Agar shablonlar bilan tanish bo'lmasangiz, funksiyalar shablonlariga oid nazariy ma'lumotlarga qarang, shablonlar bilan ishlash mexanizmi batafsil bilish lozim.

Agar shablonlar bilan tanish, lekin sinf shablonlar bilan ishlashni unutmagan bo'lsangiz, sinf shablonlari haqidagi nazariy materiallarni o'qib o'rganing.

Bu stack shablonlar bilan amalga oshirilgan kutubxonaning deyarli har qanday ma'lum otlar tipi uchun foydalanish mumkin. Bundan tashqari, stack hajmi dastur ijrosida vomidajadal belgilanadi. Stackga qo'shimcha funksiyalar ham qo'shildi peek(), bu stackning yuqori qismidan n- elementini qaytaradi.

#### 4.3-dastur. Stack shablonsini

```
#ifndef STACK_H
#define STACK_H
// Created by MBBahodir
#include <cassert>
#include <iostream>

#include <iomanip>

template <typename T>
class Stack{
private:
    T *stackPtr; // stackka
    ko'satkich
    const int size; // stackdagi
    maksimal elementlar soni
    int top; // stackning
    joriy elementi
public:
    Stack(int = 10); // stackning
    joriy holatda elementlar soni 10 ta
    Stack(const Stack<T>&); // nusxalsh
    konstruktori
    ~Stack(); // destruktori

    inline void push(const T & ); // elementni
    stackning yuqori qismiga joylashtirish
```

```

    inline T pop(); // stack
ichidan eng yuqoridagi elementni o'chirish
    inline void printStack(); // stackni
ekranga chiqarish
    inline const T &Peek(int ) const; // stackning
yuqorisidagidam keyingi n-element
    inline int getStackSize() const; // stack
o'lchamini olish
    inline T *getPtr() const; // stackka
ko'rsatkich olish
    inline int getTop() const; // joriy
elementni olish
};

// Stack sinfi shablon funksiyalarini realizatsiyz
qilish

// konstruktorSteka
template <typename T>
Stack<T>::Stack(int maxSize): size(maxSize) //
o'zgaraslarni initsalizatsiyalash
{
    stackPtr = new T[size]; // xotira ajratish
    top = 0; // joriy elementno 0 bilan toldirish;
}

// nusxalsh konstruktori
template <typename T>
Stack<T>::Stack(const Stack<T>& otherStack) :
    size(otherStack.getStackSize()) //
o'zgaraslarni initsalizatsiyalash
{
    stackPtr = new T[size]; // yangi xotira
ajratish
    top = otherStack.getTop();

    for(int ix = 0; ix < top; ix++)
        stackPtr[ix] = otherStack.getPtr()[ix];
}

```

```

}

// destruktör
template <typename T>
Stack<T>::~~Stack()
{
    delete [] stackPtr; // ochirish
}

// element qo'shish
template <typename T>
inline void Stack<T>::push(const T &value)
{
    // o'lchamni o'zgartirish
    assert(top < size); // joriy element soni
    o'lchamdan kichik bo'lishi kerak

    stackPtr[top++] = value; // elementni
    joylashtirish
}

// elementni o'chirish
template <typename T>
inline T Stack<T>::pop()
{
    // o'lchamni tekshirish
    assert(top > 0); // joriy element 0 katta
    bo'lishi kerak

    stackPtr[--top]; // elementni o'chirish
    return 0;
}

// yuqori elementdan n-elementni qaytaradi
template <class T>
inline const T &Stack<T>::Peek(int nom) const
{
    //

```

```

    assert(nom <= top);

    return stackPtr[top - nom]; // n-elementni
qaytarish
}

// ekranga chiqarish
template <typename T>
inline void Stack<T>::printStack()
{
    for (int ix = top - 1; ix >= 0; ix--)
        cout << "|" << setw(4) << stackPtr[ix] <<
endl;
}

// o'lchamni qaytarish
template <typename T>
inline int Stack<T>::getStackSize() const
{
    return size;
}

// ko'rsatkich qaytarish (nusxalash konstruktori
uchun)
template <typename T>
inline T *Stack<T>::getPtr() const
{
    return stackPtr;
}

// o'lchamni qaytarish
template <typename T>
inline int Stack<T>::getTop() const
{
    return top;
}

#endif // STACK_H

```

Stack shablonsinfi \*.h (header) fayldayaratildi. Shablonsinfinterfeysivaamalgaoshirishhambirxilfayldabo'lishikerak, bo'lmasa, shungao'xshashmazzmunbilanxatoberishimumkin.

```
error undefined reference to "method of a class template»
```

Ya'ni, xatolik, shablonsinfusullarihavolalardaaniqsizlik.

Shablonsinfinterfeysiningfragmentlarida 9 -dan 28gachayangilandi.

Sinfningbarchausullariizohlarnio'zichigaoladivameningfikrimcha, ularningfunktionalariinalohidata'riflashmantiqiyto'g'riemas. Stack shablonsinfningbarchausullari inline funksiyalarisifatidae'lonqilinadi. Businfshinitezlashtirishmaqsadidaamalgaoshiriladi.

Sinfningajralmasfunktisylaritashqifunktisylargaqaragandatezroqishlaydi.

Shabloninterfeysidanso'ng, stack sinfusullariamalgaoshiriladi, kutubxonafragmentlarida 32-117gachayangilandi. Agar stack sinf, shablonlarvasinflarusullariniamalgaoshirishhaqidabilsangizshablonsinfni yaratishdamurakkabhechnarsayo'q. Sinfdaikkitaonstrktorbor,

birinchisifragmentning 32-33 qatorlarida - bustandartkonstruktorhisoblanadi. Fragmentning 41-50 qatordaginusxakonstruktorhisoblanadi.

Birob'ektniboshqasiganusxalashuchunkerak. Peek usuli (funktisylasi), fragmentning 80-88 fatorlarida stack elementlarniko'rishimkoniniberadi. Faqatelementraqaminikiritishingizkerak, hisoblash stack yuqorisidagielementdanboshlanadivakeraklisiniqaytaradi.

Boshqafunktisylarixizmatchifunktisylarhisoblanadi, ulartashqisinfnafoydalanishuchunmo'ljallangan, ya'ni prinstack() funktisylasidantashqari, bufunktisyl stack elementlariniekrangachiqaradi.

Endi stack shablonsinfuchundasturtuzaylikvahardoimgidek, stack sinfshablonsininovbo'ladihamdaasosiyfunktisylaridanfoydalanishniko'r amiz.

4.4-dastur. Shablonsinfndanfoydalanish.

```
// Created by MBBahodir
include "stdafx.h"

#include <iostream>

using namespace std;

#include "stack.h"
```

```

int main()
{
    Stack<char> stackSymbol(5);
    int ct = 0;
    char ch;

    while (ct++ < 5){
        cin >> ch;
        stackSymbol.push(ch); // element qo'shish
    }

    cout << endl;

    stackSymbol.printStack(); // ekranga chiqarish

    cout << "\n\n Element o'chirish\n";
    stackSymbol.pop();

    stackSymbol.printStack(); // ekranga chiqarish
    Stack<char> newStack(stackSymbol);
    cout << "\n\nNusxalash konstruktori!\n";
    newStack.printStack();
    cout << "2 tartibdagi element: "<<
newStack.Peek(2) << endl;
    system("pause");
    return 0;
}

```

#### 4.4-dastur. Output.

```

b
a
h
o
q

|   q
|   o
|   h

```



```

|   a
|   b

Element o'chirish
|   o
|   h
|   a
|   b

Nusxalash konstruktori!
|   o
|   h
|   a
|   b
2 tartibdagi element: h

```

4.4-dasturda Bir stack ob'ektini yaratilgan, stack hajmi 5 gateng va stack ko'pio'zichiga olishim mumkin bo'lgan elementlar soni 5 gateng. stackni while takrorlanish operatori bilan to'ldiriladi. Ekranda stack elementlarini chiqariladi, keyin stackdan bira element o'chiriladi va yana ekranda stack elementlarini chiqariladi. Natija asosida u o'zgaradi, aniq bira element o'chirildi.

Shuningdek, dasturda nusxa konstruktor ishlatiladi. Peek() funksiyasi stackning ikkinchi elementini qaytardi.

Stack shablonsini foydalanishni yaratishni davlat o'g'ri ishlatmoqda. uni, masalan, int ma'lumotlar turidasi nabo'rcangiz bo'ladi. Ishonchim komilki, hammanarsa to'g'ri ishlaydi.

Navbat (queue). Navbat ma'lumotlar tuzilmasi bo'lib (yuqorida aytib o'tilganidek) bu LIFO tamoyilini qo'llab quvvatlaydi (last in-last out: oxirgi kirgan oxirgi chiqadi). C++ allaqachon tayyor STL konteyner-navbat (queue) mavjud.

Navbatda, birinchi element kiritilgan bo'lsa, u ham birinchi chiqariladi. 4 taelementni qo'shsangiz, birinchi qo'shilgan element birinchi, ikkinchi qo'shilgan element ikkinchi chiqadi.

Navbat qanday ishlatishni tushunish uchun do'kon navbatini tasavvur qilishingiz mumkin. Sizesauning o'rtasidaturibsiz, shundakassaning oldidaturibsiz,

avvaloldingizdagibarchaodamlargaxizmatqilinishi.

Lekin oxirginavbatdao'tgankishi uchun,

o'zitash qaribarchaodamlargaxizmatqilish kerak emas.

1	9	8	0	1	2	8
---	---	---	---	---	---	---

Buyuqoridakeltirilgan ro'yxatga qarang.

Bulardan birortasini chiqarish uchun shuketma-ketlik asosida chiqarib, kerakli sigat o'xtaladi. Masalan, 0 sonni chiqarish, uchun avval 3 tason chiqariladi so'ng kerakli son.

Stack shablonsini fidabir peek()  
(bu indeks bo'yicha elementni qaytaradi),  
bu navbat shablonidamu ayya elementni ko'rish mumkin emas.

Agar barcha navbat elementlarga kirish kerak bo'lsa, lekin,  
navbatdabuni amalga oshirish mumkin.

Keyin ro'qbuqanday amalga oshirishni ko'rsatib o'tamiz.

C++ dan navbat yaratish. Agar C++  
dan navbat shablonini ishlatmoqchibo'lsangiz,  
avval <queue> kutubxonasi ni ulashingiz kerak.

Keyin navbatni e'lon qilish uchun quyidagi sintaktikni ishlatishingiz kerak:

```
queue<type><name>;
```

Birinchi queue so'zini yozishimiz kerak.

Type ganavbatni o'ldirish uchun kerakli tipni ko'rsatishimiz lozim.

Navbat ob'ektining nomi ni ko'rsatishimiz kerak.

Masalan,

```
queue<int>q;
```

Navbat usullari va funksiyalari:

Navbatda funksiya bubi xil funksiyadir, lekin u faqat STL konteynerlar bilan ishlaydi. Bu funksiyalarga

push(), pop(), front(), back(), empty() funksiyalari kiradi.

push() – navbatga yangi element qo'shish.

pop() – navbatdan birinchi elementni o'chirish, ikkinchi element birinchi element o'rniga suriladi.

front() – navbatdagibirinchi elementni gamurojaat

back() - navbatdagibirinchi elementni gamurojaat

empty() – navbat bo'shyoki bo'shmasligini tekshiradi.

Agar bo'sh bo'lsa true, aks holda false qiymat qaytaradi.

Bu keltirilgan funksiyalarga doir 4.5-dasturini keltiramiz.

4.5-dastur. Navbat funksiyalaridan foydalanish.

```

// Created by MBBahodir
#include "stdafx.h"
#include <iostream>
#include <queue>
using namespace std;

int main() {
    queue <int> q;

    for (int h = 0; h < 7; h++) {
        q.push(rand()%100);
    }

    cout << "Navbatdigi birinchi element " <<
q.front() << endl;
    cout << "Navbatdigi oxirgi element " << q.back()
<< endl;
    q.pop();

    cout << "Navbatdigi birinchi element
(o'chirishdan so'ng): " << q.front() << endl;

    cout << "Navbat bo'sh " << (q.empty()?"":"emas")
<< endl;

system("pause");
    return 0;
}

```

#### 4.5-dastur.Output.

```

Navbatdigi birinchi element 41
Navbatdigi oxirgi element 78
Navbatdigi birinchi element (o'chirishdan so'ng):
67
Navbat bo'sh emas

```

**Massivyordamidanaavbatlarniyaratish.** Yuqoridaaytganimizdek, navbatnimassivorqaliamalgaoshirishmumkin. Odatda, agardasturchibundaynavbatniyarasa, massivnavbatdebataladi. Shuningdek, bunimassivdebhamaytishmumkin, ammomassivso‘ziC++dabor. Shuninguchun, navbatshablondabaytishhammumkinvabirxiltarzdaunichaqiriladi.

Buninguchun 2 tako‘shimchao‘zgaruvchiyaratishkerak. Birinchisi start bo‘lsinvanavbatningbirinchielementinibildirsin. Ikkinchisi ends bo‘lsinvanavbatningoxirgielementinibildirsin.

```
queue[ends] -
ifodasiniishlatibnavbatningoxirgielementiniqiymatiniqaytaradi.
```

```
queue[start] -
ifodasiniishlatibnavbatningoxirgielementiniqiymatiniqaytaradi.
```

Navbatdagibirinchielementnio‘chirishuchunstartnibirgakamaytirish kerakxolos, navbatnibo‘shliginitekshirishuchunstart=ends mantiqiyshartnitekshirishningo‘zikifoyadir. Agarmantiqiyshart true qaytarsanavbatbo‘sh, akasholdabo‘shemas.

4.6-dastur. Massivyordamidanaavbatlardanfoydalanish.

```
// Created by MBBahodir
#include "stdafx.h"
#include <iostream>
#include <queue>
using namespace std;

int main() {
    //queue <int> q;
    int q[3];
    int start = 0, ends = 0;
    for (int h = 0; h < 3; h++) {
        int a; cin >> a;
        q[start++] = a;
    }

    cout << "Navbatdagi birinchi element " << q[start
- 1] << endl;

    start--; // bir elementni o‘chirish

    cout << " Navbatdagi birinchi element
```

```

(o'chirilgandan so'ng) " << q[start - 1] << endl;

    cout << " Navbatdagi oxirgi element " << q[ends]
<< endl;

    cout << "Navbat bo'sh " << ((start == ends)?"": "
emas ") << endl;

system("pause");
    return 0;
}

```

#### 4.6-dastur.Output

```

9
7
5
Navbatdagi birinchi element 5
Navbatdagi birinchi element (o'chirilgandan so'ng)
7
Navbatdagi oxirgi element 9
Navbat bo'sh emas

```

**Ustuvornavbat (Priority\_queue).** Ustuvornavbat (Priority\_queue) bilannavbat (queue) - oddiynavbatdek, lekinustuvornavbatgayangielementqo'shilasa, shundanavbatkamayishtartibidasaralanadi. Shutariqaustuvornavbatdaengkattaelementbirinchi o'ringachiqadi. Ustuvornavbatshablonidanquyidasintaktikorqalifoydalanishkerak:

```
priority_queue<type><name>;
```

Birinchipriority\_queueso'ziniyozishimizkerak.

Type ganavbatnito'ldirishuchunkeraklitipniko'rsatishimizlozim.

Navbatob'ektiningnominiko'rsatishimizkerak.

Ustivornavbatdayangielementqo'shishuchun push() funksiyasi, birinchi o'ringidagielementigamurojaatqilishuchun top() funksiyasi ishlatiladi. Front() – funksiyasi ishlatilmaydi. Shuningdek, back() – oxirgielementgamurojaat funksiyasi hamishlatilmaydi.

4.7-dastur. Ustuvornavbatni (Priority\_queue) yaratish va usullaridan foydalanish.

```
// Created by MBBahodir
```

```

#include "stdafx.h"
#include <iostream>
#include <queue>
using namespace std;

int main() {
    priority_queue <int> priority_q,
priority_q_one;

    for (int h = 0; h < 7; h++) {
        priority_q.push(rand()%100);
    }

    priority_q_one = priority_q;

    cout << "priority_q navbatdigi birinchi element "
<< priority_q.top() << endl;
    cout << " priority_q_one navbatdigi birinchi
element " << priority_q_one.top() << endl;
    priority_q.pop();

    cout << "priority_q navbatdigi birinchi element
(o'chirishdan so'ng): " << priority_q.top() <<
endl;

    cout << "priority_q navbat bo'sh " <<
(priority_q.empty()?"":"emas") << endl;

    while (!priority_q_one.empty())
    {
        cout << "priority_q_one navbatdigi element :
" << priority_q_one.top() << endl;
        priority_q_one.pop();
    }
    cout << " priority_q_one navbat bo'sh " <<
(priority_q_one.empty()?"":"emas") << endl;

system("pause");

```

```
return 0;
}
```

```
4.7-dastur. Output
priority_q navbatdigi birinchi element 78
priority_q_one avbatdigi birinchi element 78
priority_q navbatdigi birinchi element (o‘chirishdan so‘ng): 69
priority_q navbat bo‘sh emas
priority_q_one navbatdigi element : 78
priority_q_one navbatdigi element : 69
priority_q_one navbatdigi element : 67
priority_q_one navbatdigi element : 41
priority_q_one navbatdigi element : 34
priority_q_one navbatdigi element : 24
priority_q_one navbatdigi element : 0
priority_q_one navbat bo‘sh
```

Navbatlarniyaratishni ikki usulini kishribchiqdik. Queue  
sinfshabloni va massiv asosida. Aslida ko‘proq queue  
sinfshablonidan foydalanmaqul.  
Chunki massivga nisbat tanzimlash qulayroq.

**Ikkita monolamanavbat (deque, double-ended queue).** Deque –  
shunday ma'lumotlarni tuzilmasiga ayladiki,  
undama'lumotlarni qo‘shish va o‘chirish holdidan va orqadan amalga oshiriladi.  
Deque ham xuddi queue dekkotiradasiqlanadi. Dek (Deque)  
bushunday dinamik massivki,  
u ikki tomondan kengayish va boshqanavbatlar va to‘plamlarga qaraganda tez  
roq.

Bundantashqari,  
u har qanday holatda o‘z elementlarini boshida yoki oxirida kiritish uchun ruqsat  
atberadi,  
lekin o‘z boshimchalik holatda elementlarni tez kiritish imkoniyatini saqlab qolish  
nimasligi mumkin. Vektor o‘xshash amallari kam bo‘lishi mumkin.

Dek konteyner bilan tanisharekanmiz unimasibilan yaxshi,  
boshqa konteynerlardan fandan farq qiladidegansavollar paydobo‘lishi tabiiy  
yalbatta. Dekning umumiy tavsifi dan boshlaymiz. STL da bu deko‘zini ma?  
Tushuntirish uchun, uzoqqa bormaymiz, jismoniy tarbiya darsini olamiz.  
Ko‘pchiligimiz "bo‘ybo‘yicha saflanish" nimaligini bilamiz.  
Shunday qilib, haqiqiy Dek shunday qurilish mexanizmi ga.

Undashundaybirqatorhosilbo‘ladikivabuqatoruningoxiridavauningboshid  
anto‘ldiriladi, zarurbo‘lsa, bundaketma-  
ketelementlarnio‘rnialmashtirishihammumkin.

Qanchatezroqturishkerakbo‘lsa, shunchayaxshivastek,  
navbatma‘lumotlartuzilmasigaqaraganda? Albatta,  
yoboshidayokiqatorningoxiridavaqatorningo‘rtasidatxminansizningbo‘  
ybalandligigaqarashvakeyinsolishtirishvao‘rinlarnio‘zgartirishkerak.  
Biroq, birqatoro‘rtasidajoylashtirilganbo‘lsa, butunelementlarsuriladi.  
Buhayotdandekkaqiyosimisoledi.

Shakllantirilayotganto‘planningboshidayokioxiridaelementlarniqo‘  
shishkerakbo‘lganda,  
ba‘zielementlarnio‘rtagaqo‘yishkerakbo‘ladidekniishlatishmantiqanto‘g‘  
rikeladi (mosravishdashutipdagidektanlashkerak). Dekdaketma-  
ketliko‘rtasidaqo‘shishvao‘chirishqo‘llab-quvvatlashbo‘lsa-da,  
buamallarchiziqilivaqtidaamalgaoshiriladi.  
Dasturdabukabiamallarko‘pbajarishkerakbo‘lsa,  
ro‘yxatlarengyaxshitanlovbo‘ladi.

Aslida, Dek-  
buboshidayokioxiridaelementlarningtezroqqo‘shilishiimkonibo‘lganvekt  
ordir. Shubilanbirga,  
Dekningboshqaamallarivektorlarningtegishliamallaribilanbixilishlaydig  
anvaqtga (amallarko‘naysasekinroq) ega.

Dek ( navbat queue) bilanvavektor (vector) o‘rtasidagifarqlar:

1. Deklarningboshigavaoxirigahaelementlarniqo‘shishtezkoramald  
ir, ammovektorlaruchunfaqatoxirigaqo‘shishtezkoramalhisoblanadi.  
Buqo‘shishdoimiyvaqttalabetadi.

2. Dekningfunksiyalarvaelementlarniqo‘shishganisbattanvektordagis  
hungao‘xshashfunksiyalarodatda, birozsekin.

3. Dekningiteratorlarimaxsustipdagiaqlliko‘rsatkichlargaegabo‘lishi  
kerak,  
chunkidoimiyko‘rsatkichlarularturliqismlario‘rtasidabiridanikkinchisiga  
o‘tishkerak.

4. Xotirabloklarihajmigachegaralaribortizimlaridaikkitomonlamanav  
batdekxotiraniortiqg‘ibilanajratadi,  
chunkivektorolishimumkinbo‘lganma‘lumotlarganisbattanko‘p.

Shundayqilib, max\_size ()  
funksiyasiningnatijasivektorlarganisbatanikkitomonlamanavbatlarga  
kattabo‘lishimumkin



5. Ikki tomonlamanavbatlarimkoniyatlarniboshqarishnita 'minlamayd ivaxotiraqaytaajratilgandaungajoyaniqlanmaydi. Xususan, boshidayokioxiridaelementlarniqo 'shishboshqaharqandayqo 'shishyokio 'chirishkabiikki tomonlamanavbatdaelementlarbilanbog 'liqbarchako 'rsatk ichlarni, murojaatlarniiteratorlarbekorqilmaydi. Biroq, xotiraningtaqmislanihivektorlargaqaragandayaxshiroqishlashimumkin, chunkiularodatdaichkituzilishigako 'ra, ikki tomonlamanavbatlarxotiraniqaytajoylashtirishdabarchaelementlarini ngnusxalashshartemas.

6. Deklardaishlatilmaydiganxotirabloklaribandqilinmaydivarealo 'lch amdanso 'ngxotiraningbo 'shjoylariozodqilinadi.

BuesaDekningxotirahajmikamaytirishgaimkonberadi (ammo, qachonvaqandayqilibbusodirbo 'lshiumamalgaoshirishgabog 'liq).

Dekkata 'luqlibo 'lganvektorlarningafzalligi.

1. To 'planningo 'rtasidaelementlarniqo 'shishvao 'chirishsekinroqam alhisoblanadi.

Chunkibiro 'zgarishuchunbarchaelementlarningo 'rinlarinialmashtirishker akbo 'ladi.

2. Vektorlarvadeklarningiteratorlarierkinruxsatturigamansub.

3. Deksinfi vektorsinfiningfunksiyalaribilanbirxilfunksiyalargaega.

Dekningkamchiliklariga

1. Tez-

tezqo 'shishyokio 'chirishdekningboshidayokioxiridasodirbo 'lishi.

2. Konteynerelementlarigahavolaqilaolmaysiz.

3. Konteynerishlatilmayqolgandaxotirani xoliqilishamalgaoshirilishi gaC++qafolatbermaydi (amalgaoshirishtamoyiligaqarab, xolibbo 'lshivabo 'lmasligiham mumkin)

Inobatgaolishinikerakbo 'lganholatlar:

1. Dekningat()bo 'qabirorbirfunksiyasiiteratorvaindeksnitekshiraol maydi.

2. Qo 'shishvao 'chirishqo 'shimchaxotiratalabqilishmumkin.

Shuninguchunqo 'shishvao 'chirishgadoiramallar bajarilgandadekningbos hqaelementlariuchunko 'rsatkichlar,

havolalarvaiteratorlarnianiqko 'rsataolmaydi. Bungaistisnobo 'lib,

Dekningboshigayokioxirigaelementqo 'shisholinadi.

Boshigayokioxirigaqo 'shilgandako 'rsatkichvahavolalaraniqbo 'ladi, ammoiteratoryo 'q.

3. Dekvavektorninginterfeyslarideyarlibirxil. Farqishundaki, ba 'zibirfunksiyalashyaratilganbo 'lishimumkin. Masalan, deksinfi

capacity, reserve kabifunksiyalarga ega emas.  
Buning sababidek vektorlarga o'xshabishlashini yaxshilash kerak emas.

4. Deklaratsiya qilganda,  
ayniq sama'lumotlarni qo'shishda ko'rsatkich va iteratorlarga bo'liq dastur fr  
agmenini yozishdan ehtiyoj bo'lishi va mustaqil yozish kerak emas.

Umuman olganda, shaxsiy deklaratsiya vektorlardir,  
ular bilan ishlash ham vektorlarni kabiamalga oshiriladi. Ammo qachon dekni,  
qachon vektorni tanlashni bilish kerak.

Ma'lumotlarni tuzilmasining boshiga va oxiriga qo'shish uchun optimizatsiya  
iyalangan (yaxshilangan, ixcham qilingan) vektor – bu dekdir.  
Bunda qo'shishni qo'zgarmas vaqt da bajariladi va nusxalash konstruktorini  
birmartacha qiradi.

Agar elementdekning o'rtasiga qo'shilsa,  
eng yomon iboatmal vaqt talab qiladi,  
chunki qo'shish nuqtadan boshigacha va oxirigacha bo'lgan masofalar orasid  
akichikaniqlash lozim.

insert, push\_front i push\_back  
funksiyalaridek daga ham iteratorlarni bekor qilishim mumkin.  
Shuningdekdekning o'rtasiga ma'lumot qo'shilganda havo larni xambekor  
qiladi.

Berilgan navbatdagi elementlarni guruhlash masalasini qaraylik.  
Guruhlashni amalga oshirish uchun guruhlash shartini tanlaymiz va shu asosid  
anavbatichidagi guruhlaymiz, shartga tegishli bo'lganlarni navbat boshidan,  
bo'lmaganlar oxiridan joylatiriladi.

4.8-dastur. Navbatdagi elementlarni guruhlash.

```
// Created by MBBahodir
#include "stdafx.h"
#include <iostream>
#include <deque>
#include <algorithm>
using namespace std;

bool mypred(const int x){
    return x <= 51; // guruhlash uchun shart
}

int main(){

    int Arr[]={1,78,89,23,51,49,100,18,50};
```

```

deque<int> d(&Arr[0],&Arr[9]);

cout << "Joriy deque: " << endl;
for (deque<int>::iterator
it=d.begin();it!=d.end();it++) std::cout<<*it<<" | ";
cout << endl;

stable_partition(d.begin(),d.end(),mypred);

cout << "Natija: " << endl;
for (deque<int>::iterator
it=d.begin();it!=d.end();it++) std::cout<<*it<<" | ";
cout << endl;
system("pause");
return 0;
}

```

#### 4.8-dastur.Output

```

Joriy deque:
1 | 78 | 89 | 23 | 51 | 49 | 100 | 18 | 50 |
Natija:
1 | 23 | 51 | 49 | 18 | 50 | 78 | 89 | 100 |

```

4.7-dasturdagifragmentasosiyguruhl意思shartihisoblanadi.

```

bool mypred(const int x){
    return x <= 51;    // guruhl意思 uchun shart
}

```

Bundaagarx <= 51 bo'lsa true, aks holda false qaytaradi. Bunga ekvivalent sifatida quyidagicha fragment ham yozish mumkin.

```

bool mypred(const int x){
    if (x <= 51) return true;    // guruhl意思 uchun
shart
    return false;
}

```

Dekunchunkeltirilgan materiallardastlabki o'rganuvchilar uchun yetarlidi hisoblaymiz.

Masalaning shartiga qarab deki yoki vektor nitanlash dasturchining mahoratiga bog'liq.

Dekningfunksiyalariga quyidagilarkiradi:

4.1-jadval. Dekningfunksiyalari

push_front	Boshidanyangielementqo'shish
push_back	Oxiridanyangielementqo'shish
pop_front	Birinchielementniolish
pop_back	Oxirgielementniolish
front	Birinchielementqiymatiniko'rish
back	Oxirgielementqiymatiniko'rish
size	Elementlarsoniniko'rish
clear	Barchaelementlarnio'chirish

**Konteynerlar bilan ishlash algoritmlari.** Barcha konteynerlar umumiy yondashuv tufayli, amalda qiziq tirgan asosiy algoritmlar konteynerlarning har qanday turi uchun qo'llanilishi mumkin bo'lgan umumiy shaklda amalga oshirilishi mumkinligi tufayli STL konteynerlaruzoqamaliy foydalanish uchun birarzigulik ixtiro bo'ldi.

Algoritmlar kutubxonaning eng kattava ommabop qismidir. Judako'palgoritmlar mavjudki, ularning hammasini batafsil bayon qilish uchun kattakitob kerak. Quyidagi guruhlarga ajratib, ularning nomlarini bilan ataymiz (ularning hammasi emas) va ular dan faqat ayrimlarini misol tariqasida keltiramiz.

for\_each() algoritmi. for\_each() eng ko'p ishlatiladigan algoritmdir. Konteyner elementlar guruhi (ehtimol, barchaelementlari) uchun ketma-ket harakatni amalga oshiradi. U har qanday STL konteynerda foydalanish mumkin. Quyidagi 4.9-dasturda for\_each() algoritmi massiv vektor uchun ishlatilgan.

4.9-dastur. for\_each() algoritmidan foydalanish.

```
// Created by MBBahodir
#include "stdafx.h"
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

inline ostream& operator <<( ostream& out, const
vector< unsigned >& obj ) {
```

```

    cout << "< ";
    for( auto& p: obj )
        cout << p << " ";
    return out << ">";
}

void pow2( unsigned& i ) { i *= i; }

int main( void ) {
    const int examples = 4;
    for( int i = 0; i < examples; i++ ) {
        unsigned ai[] = { 1, 2, 3, 4 , 5, 6, 7, 8, 9
},
            ni = sizeof( ai ) / sizeof( ai[ 0 ] );
        vector< unsigned > vi( ai, ai + ni );
        cout << vi;
        switch( i ) {
            case 0:
                for_each( vi.begin(), vi.end(), pow2 );
                cout << " => " << vi << endl;
                break;
            case 1:
                for_each( ai, ai + ni, pow2 );
                cout << " => " << vector< unsigned >(
ai, ai + ni ) << endl;
                break;
            case 2:
                for( auto& i : ai ) pow2( i );
                cout << " => " << vector< unsigned >(
ai, ai + ni ) << endl;
                break;
            case 3:
                for_each( vi.begin() + 2, vi.end() - 2,
pow2 );
                cout << " => " << vi << endl;
                break;
        }
    }
}

```

```

    system("pause");
return 0;
}

```

#### 4.9-dastur.Output.

```

< 1 2 3 4 5 6 7 8 9 > =>< 1 4 9 16 25 36 49 64 81 >
< 1 2 3 4 5 6 7 8 9 > =>< 1 4 9 16 25 36 49 64 81 >
< 1 2 3 4 5 6 7 8 9 > =>< 1 4 9 16 25 36 49 64 81 >
< 1 2 3 4 5 6 7 8 9 > =>< 1 2 9 16 25 36 49 8 9 >

```

Dasturning birinchi global (auto&x : ...) takrorlanish fragmenti ishlatilgan (C++11 standartitomonidan joriy qilingan).

Bu kabir fragmentni massivlar va konteynerlarga ham qo'llanilishi mumkin (bu vektorning oqimga chiqarish operator variant funksiyasida ko'rsatilgan).

Bu takrorlanish fragmentida STL kutubxonayoki algoritmlarni qismi emas, lekinu for\_each() algoritmi bilan bir xil ta'sir ko'rsatadi, ya'ni to'plam barcha elementlar uchun izchil qo'llash mumkin.

Buyuqoridagi dastur barcha algoritmlarni berilgan intervalda sosidatash kiliyasosiy mantiq qismini ko'rsatadi (shart emas, barcha konteynerlar). Boshiva oxiri bilan cheklangan iteratorga (ko'pincha, birinchi berilgan 2 ta parametrlar) funksiya, funkto'rvapredikat (bu funksiyalarga - qaysidi relementlaribo'yicha saralashimkonibo'lsavamantiqiy qiymat qayta radigan funksiyalar kiradi) navbat bilan qo'llaniladi.

Find(). Navbatdagi algoritmi - bu find(). Bu nomidan ma'lumki, to'plamdagi elementlarni qidirish funksiyasidir. E'tibor bersak,

judako'p konteynerlar bunday find() - funksiyasiga egavau ob'ekt uchun obj.find(...) sietaksida ishlatiladi.

**Shu vaqtda algoritmi quyidagicha funksiyani find( obj:iteator, ... )** ishlatish uradi. Aslida, bu bitta algoritmi emas,

balki ularning butun birkatta guruhimaveud,

ular to'plam elementlariniba'zi atribut,

shart bilan tanlashlaris sosidabirlashtirilishi mumkin.

Bu guruh predikatlariga **find(), find\_if(), find\_if\_not(), find\_first\_of(), find\_end(), adjacent\_find()** funksiyalar kiradi.

Shuningdek bu guruhdagi aytilgan algoritmlar ham tegishli dirvabularga

**count(), count\_if(), search(), binary\_search(), min(), max(), minmax\_element(), min\_element(), max\_element(), equal()**

**kabi birkatta algoritmlar kiradi.**

To'plamga ishlov berish (aralashtirish) birshartli guruhalgoritmlar mavjud va ular joylarda elementlarni qaytatartibg asoluvchi va qiymatlarni o'zgartiruvchi algoritmlardir: **fill()**, **replace\_copy()**, **reverse()**, **rotate()**, **rotate\_copy()**, **shuffle()**, **random\_shuffle()**, **transform()**, **replace()**, **replace\_if()** **vaboshqa funksiyalar.**

Ikkito'plamlar ustida bajariladigan algoritmlar guruhi ham mavjud. Bular asosan nusxalash, tarkibi ni ko'chirib o'tish (xartipdag konteynerlar uchun bo'lishi mumkin. Masalan, **vector<>** ni **set<>ga**) funksiyalarini bajaradi. Bular **copy()**, **copy\_if()**, **move()**, **swap\_ranges()**, **remove\_copy()**, **remove\_copy\_if()**, **merge()**, **set\_intersection()**, **set\_difference()** **vaboshqa algoritmlar.**

Nihoyat, juda maxsus guruhalgoritmlari, to'plam ichida elementlarni tartibga solish bilan bog'liq funksiyalar **sort()**, **stable\_sort()**, **is\_sorted()**, **is\_sorted\_until()** **vaboshqa algoritmlar.** Ushbu qiziqarli guruhi ni keyinroq, alohida batafsil ko'rib chiqish kerak.

Kutubxonada vaqto'lish bilan ortib borayotgan algoritmlarning ko'pligi va ularning aksariyati adabiyotda hech qanday ta'riflanmasligiga qaramasdan, tabiiy savol tug'iladi: bu xilma-xillikni qanday tushunish kerak? Bu qiziqchiliklar qanday eng o'lim mumkin:

- Barcha STL ob'ektlar (konteynerlar, algoritmlar) shablonsintaktik bo'yicha tasvirlangan. Shuning uchun, ularning tavsiflari, ularning header fayllarida dastur fragmentlari sifatida tuzilgan bo'lishi kerak.

- Kerakli algoritmlar uchun header fayllarini standart </usr/include/c++> katalogiga o'ting va **stl\_algo\*** kabilar lavhafayllarini toping - u da barcha algoritm funksiyasi prototiplarini (o'xshashlarini) topasiz. Bundan tashqari, har bir prototipial algoritm maqsadini tushuntirib va parametrlarini tushuntirishning batafsil izohi dindanyozib ko'yilgan.

- Bir necha asosiy STL algoritmlarini foydalanish gadoirmisollarni ko'rib chiqing - kutubxonalar tizimida ularning ko'pchiligini bor. O'xshashlik bilan boshqabarcha algoritmlarni yaratish mumkin.

Shablonsinfi kutubxonalar shablon jihatidan **template** termin bilan belgilangan, chunki o'sha, kompilyatsiya sintaktik xat bo'yicha xabarlarini chiqaradi. Birinchi siyozmaxabarlar fragmentlari bo'yicha va ikkinchi xatolarni izlashu

chun. Bushablonkabikuchlimexanizmitomonidanqayytaishlanidi (tekshiriladi) vadasturchibuuchuntayyorbo‘lishikerak.

Yuqoridaaytibotilganidek, misollarnio‘rganishjudako‘psavollargajavobberadi, shuninguchundasturkeltiramiz.

Endio‘rganganalgoritmlarnidiqqatbilaneslang (harbirinisharhlab, kutubxonasinflaridantashqaridabo‘lganjudako‘p STL algoritmlarimavjud, ammoularningbarchasibir-birigao‘xshabishlaydi):

4.10-dastur. Standartalgoritmlardanfoydalanish.

```
#include "stdafx.h"
#include <iostream>
#include <cstring>
#include <vector>
#include <map>
#include <set>
#include <algorithm>
using namespace std;

inline ostream& operator <<( ostream& out, const
vector<char>& obj ) {
    for( auto p: obj ) cout << p;
    return out;
}

int main( void ) {
    char s[] = "Muhammad al-Xorazmiy nomidagi
Toshkent Axborot Texnologiyalari Universiteti"; //
nazvanieknigi
    vector<string> vs;
    vs.push_back("Muminov");
    vs.push_back("Bahodir");
    // copy & find :
    vector<char> v1( strlen( s ) );
    copy( s, s + v1.size(), v1.begin() );
    int nb = 0;
    for( auto is = find( v1.begin(), v1.end(), ' '
); is != v1.end();
        is = find( ++is, v1.end(), ' ' ) )
```



```

nb++;
    cout << "bo'sh joylar soni: " << nb << " (" <<
nb + 1
<< " so'z)" << endl;
    // min & max :
    auto mm = minmax_element( v1.begin(), v1.end()
);
    cout << "belgilar uzunligi: '" << *mm.first <<
"' ... '"
<< *mm.second << "'" << endl;
    // fill & reverse & rotate & shuffle :
    vector<char> suv( vs[ 0 ].size() );
    copy( vs[ 0 ].begin(), vs[ 0 ].end(),
suv.begin() );
    cout << suv << endl;
    random_shuffle( suv.begin(), suv.end() );
    cout << suv << endl;
    reverse( suv.begin(), suv.end() );
    cout << suv << endl;
    rotate( suv.begin(), suv.begin() + suv.size() /
2, suv.end() );
    cout << suv << endl;
    // set_intersection & set_difference
    set< char > sus, pns;
    for( char s: vector<char>( vs[ 0 ].begin(), vs[
0 ].end() ) )
        sus.insert( s );
    for( char s: vector<char>( vs[ 1 ].begin(), vs[
1 ].end() ) )
        pns.insert( s );
    vector<char> outi( 100 ), outd( 100 );
    auto ret = set_intersection( sus.begin(),
sus.end(), pns.begin(),
                                pns.end(),
outi.begin() );
    cout << "umumiy belgi " << ( ret - outi.begin()
) << " : "
<< outi << endl;

```

```

    ret = set_difference( sus.begin(), sus.end(),
pns.begin(),
                        pns.end(), outd.begin() );
    cout << "unikal belgi " << ( ret - outd.begin()
) << " : "
<< outd << endl;
    system("pause");
    return 0;
}

```

#### 4.10-dastur. Output

```

bo'sh joylar soni: 6 (7 so'z)
belgilar uzunligi: ' ' ... 'z'
Muminov
nuvmMoi
ioMmvun
mvunioM
umumiy belgi 2 : io
unikal belgi 5 : Mmnuv

```

Dasturda char uchunkonteynerlardan foydalanilgan. Deyarli barcha berilgan konteynerlar uchun turli xil algoritmlar amalga oshirilgan. (ixcham, ammo zerikarli fragmentlar).

**Funktorlarning ko‘llanilishi.** Funktor so‘zi C++ da “funktional ob’ektlar” ning qisqartmasidan olingan. C++ da funktional ob’ekt operator() bilan aniqlangan sinfning nusxasi hisoblanadi. Agar C++ sinfi operator() aniqlasangiz, u funksiyakabi ishlaydi, ammo holat nisaqlay olmaydi. Masalan, 4.11-dasturga qarang.

4.11-dastur. Operator() niqayta ishlatish.

```

#include "stdafx.h"

#include <iostream>
#include <string>

using namespace std;

class SimpleFunctor {
    string name_;

```

```

public:
    SimpleFunctor(const char *name) : name_(name)
    {}
    void operator()() { cout << "Salom, " << name_
<< " ishlar zo'rmi!" << endl; }
};

int main() {
    SimpleFunctor sf("Akbar");
    sf();
    system("pause");
    return 0;
}

```

4.11-dastur. output

Salom, Akbar ishlar zo'rmi!

Dasturdasfob'ektbo'lishidanqa'tiynazarsf() funksiyasinichaqiramiz. Chunki SimpleFunctop sinfida operator() aniqlangan.

Ko'phollarda STL kutubxonasi ningtaqqoslashvaxatolikfunksiyasiningalgoritmlaridafunktor larC++dapredikatlarsifatidaishlatiladi. Farazqilaymiz, sonlarketmaketligiberilgan.

Buketmaketlikdagibarchajuftsonlarvatoqsonlaryig'indisinihisoblashkera kbo'lsin. Bumasalaniyechishuchunfunktorva for\_each algoritmidanfoydalanishmumkin.

4.12-dastur. Funktorva for\_each algoritmidanfoydalanish.

```

#include "stdafx.h"
#include <algorithm>
#include <iostream>
#include <list>
using namespace std;
class EvenOddFunctor {
    int even_;
    int odd_;
public:
    EvenOddFunctor() : even_(0), odd_(0) {}
    void operator()(int x) {

```

```

        if (x%2 == 0) even_ += x;
        else odd_ += x;
    }
    int even_sum() const { return even_; }
    int odd_sum() const { return odd_; }
};

int main() {
    EvenOddFunctor evenodd;

    int my_list[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
};
    evenodd = for_each(my_list,
my_list+sizeof(my_list)/sizeof(my_list[0]),
                    evenodd);

    cout << " Juftlar yig'indisi: " <<
evenodd.even_sum() << endl;
    cout << " Toqtlar yig'indisi: " <<
evenodd.odd_sum() << endl;
    system("pause");
    return 0;
}

```

#### 4.12-dastur. output

```

Juftlar yig'indisi: 30
Toqtlar yig'indisi: 25

```

Dasturda `for_each` da `EvenOddFunctor` sinfob'ektiberilgan. `for_each` `my_list` bo'yichaketmaketo'tgandafunktorfunksiyanichaqirmoqda. Hartakrorlanishda `evenodd` funktornusxasiniqaytaradi. `Evenodd` funktoresajuftyokitoqelmentlarniyig'indisini saqlaydi.

**Standart MLd funktorlar.** ML dagifunksiyainterfeyslarningumumiyyatdabiqlarimavjudvaOYDjihatidansh akllantirishqiyinmasallardanhisoblanadi. ML standartjihatidanfunktortar ML modultiziminingbirqismidirvaulartuzilmalarniyaratishgaimkonberadi.

Masalan, plagintiziminiyozishmasalasiniolaylik. Aytaylikbarchaplaginlarsoddaroqbo‘lishiuchun, faqatperformfunksiyasinio‘zichigaoladiganzarurinterfeysiniamalgaoshiri shkerakbo‘lsin. MLdaavvalpluginlaruchunmaxsustasdiqlovchi (imzolovchini)niyaratishzarur.

```
Signature Plugin =
sig
  val perform: unit -> unit
end;
```

Pluginlaruchuntasdiqlovchiinterfeysnianiqladik, ikkitaplaginni, aytaylik, Loudplugin va Silentpluginnituzilmalarorqaliamalgaoshirishimizmumkin. LoudPluginniamalgaoshirish:

```
structure LoudPlugin :> Plugin =
struct
  fun perform() = print "KATTA HARFLAR!\n"
end;
```

Silentpluginniamalgaoshirish:

```
structure SilentPlugin :> Plugin =
struct
  fun perform () = print "kichik harlar\n"
end;
```

Endifunktorlargaqaytarishimiz. ML dagifunktorlartuzilmalarniargumentsifatidaqabulqilishadi, shuninguchun**Plugin**argumentsifatidaberilgan.

```
functor Performer(P : Plugin) =
struct
  fun job () = P.perform ()
end;
```

BufunktorPlugin ni P argumentsifatidaqabulqiladiva P plaginni perform funksiyasinichaqiradigan job funksiyasiuchunishlatadi.

Performer funktorniishlatamiz (Funktortuzilmaqaytarishinibilamiz).

```
structure LoudPerformer = Performer(LoudPlugin);
structure SilentPerformer = Performer(SilentPlugin);

LoudPerformer.job ();
SilentPerformer.job ();
```

Standard ML uchun koddiymi soledi.  
Ammo chuqurroq o'rganish uchun Stili ni ham bilish talab qilinadi.

### Haskell da funkto'rlar.

Haskell da funkto'rlar haqiqiy funkto'rlar bo'lishi bilan farqlanadi.  
Haskell da funkto'rlar kate'goriya nazariyasidan matematik funkto'rlarga juda o'xshaydi (Kate'goriya nazariyasi - matematika ning o'g'ektlarining ichki tuzilishi ga bog'liq bo'lmagan matematika kob'ektlar o'rtasida gimunosabatlarning xususiyatlarini o'rganadigan bo'limi).

Kate'goriya nazariyasida funkto'rlar kate'goriyalar o'rtasida shunday taqsimlanishi kerakki,

kate'goriya strukturasi saqlanib qolsin yoki boshqa qacholibayt ganda, u ikki kate'goriya orasida gimomorfi zmb o'lsin.

Buta'rif Haskell da oddiy tip dagi sinfsifatida amalga oshiriladi.

```
class Functor f where
  fmap :: (a -> b) -> f a -> f b
```

ML ga qarab, masalan, Haskell tipi plagi sinfiga o'xshaydi, joriy sinfn ing nusxasi bo'lish uchun qaysi amallarni qanday qilib yaratishni aniqlaydi. Bu holatda, birgina Functor tiplari bo'yicha aniqlanmagan, balki f konstruktorasosida aniqlangan. Bundan shuni bilish mumkin ki Functor - fmap funksiyasini yaratib beradi. fmap funksiyasi tipini qabul qiladiv a b tipini qaytaradi. fa (f konstruktoriasosida qurilgan tip, a ga qo'llaniladi) qabul qilinadigan tip va fb qaytariladigan tip.

Unit ushuni sh uchun, ba'zi konreynerdaha bir element uchun amal qiladigan funksiyasi sifatida fmap funksiyasini o'ylab ko'ring. Funkto'rlarning oddiy misoli muntazam ro'yxatlar va map funksiyasi bo'lib, u ro'yxatdagi har bir element uchun funksiyani qo'llaydi.

```
Prelude> map (+1) [1,2,3,4,5]
[2,3,4,5,6]
```

Bu oddiy misolda fmap funksiyasi faqat map va tipidagi konstruktor [] - list tipidagi konstruktor dir. Shuning uchun, ro'yxatlar uchun, masalan, Functor sifatida belgilangan.

```
instance Functor [] where
  fmap = map
```

Bu map o'rniga fmap yordamida fragmentni shakllantirib olsak, albatta, to'g'ri bo'ladi.

```
Prelude> fmap (+1) [1,2,3,4,5]
```

[2,3,4,5,6]

Lekin funktor ta'rif tuzilishini asrab-avaylash haqida hech narsa demaydi, unutmang! Shuning uchun har qanday normal funktorlarning matematik funksiyalarga ta'rifiga kiruvchi funktorlar qonunlarini qanoatlantirish kerak. Fmapning ikki qoidalaribor:

```
fmap id = id
fmap (g. h) = fmap g. fmap h
```

Birinchi qoida konteynerdagi har bir element uchun bir xil funksiya ni xiralash hec qanday ta'siri yo'q deb aytilgan. Ikkinchi qoida gako'ra, konteynerdagi har bir element ustida ikki ta'rif funksiyani tarkibida birinchi funksiyani ko'rsatish bilan, so'ngra ikkinchi siniko'rsatish bilan bir xil bo'ladi.

Ularni yaqqol ko'rsatish uchun funktorlarga yanabir misol daraxtlar ustida amallar hisoblanadi. Bir konteyner sifatida daraxt, shajara (ierarxiya, ichma ich joylashgan ob'ektlar) berilgan bo'lsin. Daraxt tuzilishini saqlab qolish uchun, daraxt xususiyatlariga fmap funksiyasini qo'llaymiz. Buning uchun birinchi daraxt ni belgilab boshlaymiz.

```
data Tree a = Node (Tree a) (Tree a)
              | Leaf a
              deriving Show
```

Bu fragmentda daraxt turi ikki daraxtning (chap va o'ng tugunlariga) tuguni (node) yokibarg (leaf) ekanligi aytiladi. **deriving Show** ifoda show funksiyasi orqali ko'rishim ko'ni beradi. Endi Tree daraxti ustida funktor ni aniqlayolamiz.

```
instance Functor Tree where
  fmap g (Leaf v) = Leaf (g v)
  fmap g (Node l r) = Node (fmap g l) (fmap g r)
```

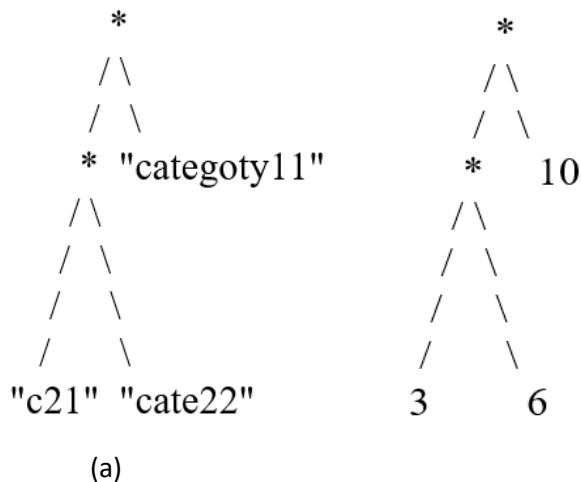
Bu fragmentda fmap tipidagi funksiyasi argument sifatida Leaf tipli qiymatni qabul qiladiv Leaf tipidagi gni v ga qiymat qilib beradi. Ikkinchi qoida da fmap tipiga oid g funksiyasi Node tipidagi l va r qiymatlarini qabul qiladiv natijada Node funksiyasi fmap tipidagi gl va gr argumentlarini qiymat qilib beradi.

Fmap funksiyasini tree ob'ektlari bilan ishlash uchun tayyorlaymiz. Tree sinfi sifatli (string) ro'yxat bilan quramiz va har bir ro'yxat uzunligini length() funksiyasi bilan aniqlaymiz.

```
Prelude> let tree = (Node (Node (Leaf "c21") (Leaf "cate22")) (Leaf "category11"))
```

```
Prelude> fmap length tree
Node (Node (Leaf 3) (Leaf 6)) (Leaf 10)
```

Quyidagichatree quriladi:



4.1-rasm. (a) tree ko‘rinishi, (b) length() bo‘yicha ko‘rinishi.

Aslida, Haskell da funktorlar fundamental funktorlarini hisoblanadi va applicative functors vastrelkalar barchasibular ham funktorlaridir. Haskell funksiyalar murakkab hisoblanadi, agar bu funktorlarni haqiqiy xayot qonuniyatlariga ‘labo‘rganmoqchibo‘lsa ngi, alohida fan vama balarni o‘rganish lozim.

Prolog funktorlari. Bu hammasidan oson funktorlardir. Ikki tamisol qaraymiz. Birinchisi atom - butuzilmaning birinchisidir. Ifodani qaraymiz:

```
?- likes(olma, nok)
```

Birinchi atom funktordir – likes

Ikkinchisi - functor deb nomlangan ajralmas predmet.

Bu argumentlar va funktor tuzilmasini qaytaradi. masalan,

```
?- functor(likes(olma, nok), Functor, Arity).
```

Functor = likes

Arity = 2

Bu Prolog funktorlarga misol edi.

Funktorlarni yaratish dasturchilarga xos masal batta.

Chunki bugungi kundatayyor foydalanish uchun ko‘plab funktorlarni yaratilgan.

Ularni asosan instrument yaratuvchilar, ya‘ni dasturchilar uchun dastur tuzuvchilarni yaratadi. Apm mantiqiy dasturlash funktorlarining o‘rnini muhim ahamiyat kasbetadi.



## NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1. Adapterlarnimavaqaysisinflarkiradi?
2. Konteynerlarningadapterlariqaysisinflarkiradi.
3. Judakengvatez-tezishlatiladiganma'lumotlartuzilmalaribu – nima?
4. Konteyneradapterlariningfunksiyalarinita'minlashuchun STL standartkonteyneriharqandaymoslashtirishdatayanchsifatidaqaysiamallardan (adapterturigaqarab) foydalanishmumkin.
5. Birinchinavbvtidakonteynerlarningadapterlariuchunqandaysinflarnidasturgaqo'shishlozim.
6. Steknimavaunimasibilanharakterlanadi.
7. Birinchikirganoxirichiqaditamoyilibo'yichafaoliyatko'rsatuvchima'lumotlarto'plamibunima.
8. Navbvtnimavaunimasibilanharakterlanadi.
9. BirinchikirganBirinchichiqaditamoyilibo'yichafaoliyatko'rsatuvchima'lumotlarto'plamibunima.
10. Navbatningqaysiusulivektorkonteynerustigaquribbo'lmaydi, lekinuniro'yxatkonteyneruchunqurishmumkin.
11. Deque sinfiuchun*front()*, *push\_back()*, *pop\_front()*, usullariharqandaykonteynerlardaqachonishlatiladi?
12. Deque sinfidakaysiusulio'rniga *front()* usuliishlatiladi?
13. Ma'lumbirtartibdaqaytaishlanishikerakma'lumotlarniuchunumumiyma'lumotlartuzilishigaasoslanganto'plamibu – qandayto'plamlar?
14. Ma'lumotlartuzilishi - stekjudaoddiy: uqandaytartibnibelgilaydi?
15. Stekningassosiativusullarinisanabbering?
16. Navbatda, birinchiementkiritilganbo'lsa, uqachonchiqariladi?
17. Agar C++ danavbatshabloniniishlatmoqchibo'lsangiz, avvalqandaykutubxonasiniulashingizkerak.
18. *empty()* – navbatuchunnimavazifanibajaradi.
19. Massivyordamidnavbatlarniyaratishmumkinmi?, mumkinbo'lsaizohlabbering.
20. Ustuvornavbat (*Priority\_queue*) bilannavbat (*queue*) ob'ektlariningfarqinimada.

21. Ustivornavbvtdayangielementqo‘shishuchun push() funksiyasi, birinchio‘rindagielementigamurojaatqilishuchunesaqaysifunksiyasi ishlatiladi.
22. Ikkitomonlamanavbatvaodiiynavbatningfarqinimada?
23. Dekvavektorningo‘xshashtomonlarivafarqlari.
24. Ikkitomonlamanavbatningavzalliklarinisanabbering.
25. Nimauchunikkitomonlamanavbatdusturchigakerak. 3 taasoskeltiring.
26. Dekningmaummolarinibilasizmi? Deknavbatqachondasturchigapandberishimumkin.
27. Konteynerelementlarguruhi (ehtimol, barchaelementlari) uchunketmaketharakatniamalgaoshiradiganalgoritmninganiqlang.
28. To‘plamdagielementlarniqidirishfunksiyasi find()gata‘rifbering.
29. To‘plamgaishlovberish (aralastirish)uchunbirshartliguruhalgoritmlarmavjudvaularjoylardaelementlarniqaytatartibgasoluvchivaqiyamatlarnio‘zgartiruvchialgoritmlardir. Ularnisanabbering.
30. Ikkito‘plamlarustidabajariladiganalgoritmlarguruhiqaysialgoritmlarmisolbo‘laoladi.
31. Funktorso‘ziC++daqaysiso‘zningqisqartmasidanolingan.
32. C++dafunksionalob‘ekt — qaysioperatorbilananiqlangansinfningnusxasihisoblanadi.
33. Ko‘phollarda STL kutubxonasingtaqqoslashvaxatolikfunksiyasiningalgoritmlaridafunktorlarC++danimalsifatidaishlatiladi.
34. ML standartjihatidanfunktorlar ML modultiziminingbirqismidirvaularnimalariniyaratishgaimkonberadi.
35. Haskelldagifunktorlarkategoriyanazariyasidanqaysifunktorlar gajudao‘xshaydi.



**AMALIY KO‘NIKMA VA MALAKALARNI ANIQLASH  
HAMDA RIVOJLANTIRISH UCHUN ASSISMENT  
TOPSHIRIQLARI.**

<b>BIRINCHI ASSISMENT TOPSHIRIG‘I</b>	
	Stack — stekkonteynerlaradarterigaoidberilganquyidagidasturbo‘yichabe

	<p>rilgantopshiriqlarkeraklifragmentlariasosidabajarang.  👉 Bundasturdagiba‘zio‘zgartirishlarnitopishorqalitopshiriqlarb  osqichma – bosqichamalganoshiriladi.</p>
dastur	topshiriqlar
<pre>// Created by MBBahodir #include "stdafx.h"  #include &lt;iostream&gt; #include &lt;vector&gt; #include &lt;stack&gt; #include &lt;string&gt; #include &lt;array&gt; #include &lt;string&gt; using namespace std;  int main() {     stack&lt;int&gt; st, st_one, st_two;     stack&lt;string, vector&lt;string&gt;&gt; vst;      vst.emplace</pre>	<p>1. Dasturdanechtaglobalfunksiyayaratilgan.</p> <hr/> <hr/> <hr/>
	<p>2. Dasturdanechtastack to‘plammavjud.</p> <hr/> <hr/> <hr/>
	<p>3. Dasturdavst.push("belgi - " +  to_string(int(i)) +  to_string(i));qandayvazifanibajaradi.</p> <hr/> <hr/> <hr/> <hr/>
	<p>4. Ikkixiltipdagistekto‘plamelementlarinialma  shtirishdasturfragmentiniyozing.</p> <hr/> <hr/>
	<p>5. (!vst.empty())  qachonchinvaqachonyolg‘onqiymatlarniqaytara  di.</p> <hr/> <hr/>

<pre> ("str1");     vst.emplace ("str2");     vst.push("e ostr");     for (int i = 0; i &lt; 255; i++)     {          vst.push("b elgi - " + to_string(int(i )) + to_string(i));     }      for (int i=0; i&lt;50; ++i) st.push(i*15);      for (int i = 0; i &lt; 105; i++) st_one.emplace( i*10);      st_two = st;     st.swap(st_ one);      cout &lt;&lt; st.size() &lt;&lt; " " &lt;&lt; st_one.size() &lt;&lt; " " &lt;&lt; st_two.size() </pre>	<p>6. Stekdanelementlarni ishlash funksiyasini tuzing.</p> <hr/> <hr/> <p>7. while (st.size() != 0){  cout &lt;&lt; st_two.top() &lt;&lt; " :  stack dagi o'rni " &lt;&lt; st_two.size()  &lt;&lt; endl;  st_two.pop();  }  Dastur fragmentini stek elementlarini saqlab qolgan xolda ishlatish funksiyasini tuzing.</p> <hr/> <hr/> <p>8. Stek vektor bilan farqlarini qlovchi dastur tuzing.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
---	--

```

<< " " <<
vst.size() <<
endl;

    while
(!vst.empty())
{
        cout <<
vst.top() << "
: stack dagi
o'rni " <<
vst.size() <<
endl;

        vst.pop();
    }

    int size =
st.size();
    while
(size--) {
        cout <<
st.top() << " :
stack dagi
o'rni " <<
st.size() <<
endl;

        st.pop();
    }

    while
(st.size() !=
0){
        cout <<
st_two.top() <<
" : stack dagi
o'rni " <<

```

```

st_two.size()
<< endl;

    st_two.pop(
);
    }

    while
(!st.empty()) {
        cout <<
st.top() << " :
stack dagi
o'rni " <<
st.size() <<
endl;


        st.pop();
    }

    cout <<
st.size() << "
" <<
st_one.size()
<< " " <<
st_two.size()
<< " " <<
vst.size() <<
endl;

    cout << "
stack st " <<
(st.empty() ?
"" : "not ") <<
" empty" <<
endl;
    cout << "
stack vst" <<
(vst.empty() ?

```

<pre> "" : "not ") &lt;&lt; " empty" &lt;&lt; endl;     cout &lt;&lt; " stack st_one " &lt;&lt; (st_one.empty() ? "" : "not ") &lt;&lt; " empty" &lt;&lt; endl;     system("pau se");     return 0; } </pre>	
<p>9. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____</p> <p>10. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.</p>	

<b>IKKINCHI ASSISMENT TOPSHIRIG‘I</b>	
	<p>queue —  navbatkonteyneradapterigaoidberilganquyidagida  sturbo‘yichaberilgantopshiriqlarkeraklifragmentl  ariasosidabajaring.  👉 Bundadasturdagiba‘zio‘zgartirishlarnitopishorq  alitopshiriqlarbosqichma —  bosqichamalganoshiriladi.</p>
dastur	topshiriqlar
<pre> // Created by MBBahodir #include "stdafx.h"  #include &lt;iostream&gt; #include &lt;list&gt; #include &lt;queue&gt; #include &lt;string&gt; #include &lt;array&gt; using namespace std; </pre>	<p>1. Dasturdanechtaglobalfunksiyay  aratilgan.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>

<pre> int main() {      queue&lt;int&gt; st, st_one, st_two;     queue&lt;string, list&lt;s tring&gt;&gt; vst;      vst.emplace("str1"); vst.emplace("str2"); vst.push("eostr");      for (int i=0; i&lt;5; ++i) st.push(i);      for (int i = 0; i &lt; 15; i++) st_one.emplace(i*10);      st_two = st; st.swap(st_one);      cout &lt;&lt; st.size() &lt;&lt; " " &lt;&lt; st_one.size() &lt;&lt; " " &lt;&lt; st_two.size() &lt;&lt; " " &lt;&lt; vst.size() &lt;&lt; endl;      while (!vst.empty()) {         cout &lt;&lt; vst.front() &lt;&lt; " : stack dagi o'rni " &lt;&lt; vst.size() &lt;&lt; endl;         vst.pop();     } </pre>	<p>2. Dasturdanechtaqueue sinfining ob'ektining to'plam mavjud.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
	<p>3. Dasturdagi ortiqchakutubxonalar ni o'chirib tashlang va sabablarini ayting.</p> <hr/> <hr/> <hr/>
	<p>4. Navbtda qachon oxirgi element qoladi. Dastur fragmenti asosida ko'rsatib bering.</p> <hr/> <hr/> <hr/>
	<p>5. (st.size() != 0) shartni mantiqantakrorlanish gashartqi libqo'yishto'g'rimivanimuchun. Anologini yozing.</p> <hr/> <hr/> <hr/> <hr/>



<pre> int size = st.size(); while (size--) {     cout &lt;&lt; st.front() &lt;&lt; " : stack dagi o'rni " &lt;&lt; st.size() &lt;&lt; endl;     st.pop(); } </pre>	<p>6.st.swap(st_one); nio‘miga st_one.swap(st); yozilsanimaho disasodirbo‘ladi.</p> <hr/> <hr/> <hr/> <hr/>
<pre> while (st.size() != 0){     cout &lt;&lt; st_two.front() &lt;&lt; " : stack dagi o'rni " &lt;&lt; st_two.size() &lt;&lt; endl;     st_two.pop(); } </pre>	<p>7. Navbatning dasturda ishlatimagan funksiyasini aniqlang.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
<pre> /* while (!st.empty()) {     cout &lt;&lt; st.front() &lt;&lt; " : stack dagi o'rni " &lt;&lt; st.size() &lt;&lt; endl;     st.pop(); } */  cout &lt;&lt; st.size() &lt;&lt; " " &lt;&lt; st_one.size() &lt;&lt; " " &lt;&lt; st_two.size() &lt;&lt; " " &lt;&lt; vst.size() &lt;&lt; endl;  cout &lt;&lt; " stack st " &lt;&lt; (st.empty() ? "" : "not ") &lt;&lt; " empty" &lt;&lt; endl; cout &lt;&lt; " stack vst" &lt;&lt; (vst.empty() ? "" : </pre>	<p>8. Dasturdagi izohga olingan dastur fragmenti olib tashlansa, dasturda qandayo‘zgarishlar bo‘ladi va natijasichi.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>

```

"not ") << " empty" <<
endl;
cout << " stack st_one
" << (st_one.empty() ?
"" : "not ") << "
empty" << endl;
system("pause");
return 0;

```

9. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. \_\_\_\_\_  
10. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.

### UCHINCHI ASSISMENT TOPSHIRIG‘I

	<p>Ustuvornavbat (Priority_queue)  vaikkitomonlamanavbat (deque)  oidberilganquyidagidasturbo‘yichaberilgantop  shiriqlarkeraklifragmentlariasosidabajarining.  👉 Bundaydasturdagiba‘zio‘zgartirishlarnitopish  orqalitopshiriqlarbosqichma —  bosqichamalغانoshiriladi.</p>
--	--

dastur	topshiriqlar
<pre> // Created by MBBahodir #include "stdafx.h" #include &lt;iostream&gt; #include &lt;queue&gt;  using namespace std; bool mypred(const int x){ return x &lt;= 51; // guruhlash uchun shart }  int main() { priority_queue &lt;int&gt; priority_q, priority_q_one; </pre>	<p>1. Dasturdanechtatpdaginavbatob’  ektlariyaratilgan.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <p>2. Dasturdapriority_q.push(rand()  %100);nio‘rnigaboshqaoperatorniy  ozing.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>

<pre> for (int h = 0; h &lt; 7; h++) {  priority_q.push(rand()% 100); } </pre>	<hr/> <hr/> <hr/> <hr/>
<pre> priority_q_one = priority_q;  cout &lt;&lt; "priority_q navbatdigi birinchi element " &lt;&lt; priority_q.top() &lt;&lt; endl; cout &lt;&lt; " priority_q_one navbatdigi birinchi element " &lt;&lt; priority_q_one.top() &lt;&lt; endl; priority_q.pop();  cout &lt;&lt; "priority_q navbatdigi birinchi element (o'chirishdan so'ng): " &lt;&lt; priority_q.top() &lt;&lt; endl;  cout &lt;&lt; "priority_q navbat bo'sh " &lt;&lt;(priority_q.empty()?" ":"emas") &lt;&lt; endl;  while </pre>	<p>3. Dasturdanechtapriority_queueet o'plammavjud.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
<pre> priority_q.pop();  cout &lt;&lt; "priority_q navbatdigi birinchi element (o'chirishdan so'ng): " &lt;&lt; priority_q.top() &lt;&lt; endl;  cout &lt;&lt; "priority_q navbat bo'sh " &lt;&lt;(priority_q.empty()?" ":"emas") &lt;&lt; endl;  while </pre>	<p>4. Dasturdadeque&lt;int&gt;d(&amp;Arr[0],&amp;Arr[9]);funksiyasi qandayvazifanibajaradi.</p> <hr/> <hr/> <hr/> <hr/>
<pre> priority_q.pop();  cout &lt;&lt; "priority_q navbatdigi birinchi element (o'chirishdan so'ng): " &lt;&lt; priority_q.top() &lt;&lt; endl;  cout &lt;&lt; "priority_q navbat bo'sh " &lt;&lt;(priority_q.empty()?" ":"emas") &lt;&lt; endl;  while </pre>	<p>5. To'plamdanberilganqiymatniiz lashdasturfragmentiniyozing.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>

<pre> (!priority_q_one.empty( )) {     cout &lt;&lt; "priority_q_one navbatdigi element : " &lt;&lt; priority_q_one.top() &lt;&lt; endl;  priority_q_one.pop(); } cout &lt;&lt; " priority_q_one navbat bo'sh " &lt;&lt; (priority_q_one.empty() ?"": "emas") &lt;&lt; endl;  int Arr[]={1,78,89,23,51,49 ,100,18,50}; deque&lt;int&gt; d(&amp;Arr[0],&amp;Arr[9]);  cout &lt;&lt; "Joriy deque: " &lt;&lt; endl; for (deque&lt;int&gt;::iterator it=d.begin();it!=d.end( );it++) std::cout&lt;&lt;*it&lt;&lt;"   "; cout &lt;&lt; endl;  stable_partition(d.begi n(),d.end(),mypred);  cout &lt;&lt; "Natija: " &lt;&lt; endl; </pre>	<p>6. Ustivornavbatshartkiritibikkita yanginavbatyaratuvchidasturtuzing.</p> <hr/> <hr/> <hr/> <p>7.Dasturda it qandaytip, nimauchunaynandeknavbatgafoydalanilgan.</p> <hr/> <hr/> <hr/> <p>8. operator= dasturningqaysiqismlaridafoydalanilganvanimauchun.</p> <hr/> <hr/> <hr/>
--	---

```
    for
    (deque<int>::iterator
    it=d.begin();it!=d.end(
    );it++)
    std::cout<<*it<<" | ";
    cout << endl;

    system("pause");
    return 0;
}
```

9. Dasturda jamibo'lib, nechao'zgartirish kiritildi. \_\_\_\_\_
10. Shu dasturning analogini yaratish sizga mustaqil vazifa dir.

## TO‘RTINCHI ASSISMENT TOPSHIRIG‘I

	<p>Konteynerlar bilan ishlash algoritmlariga oid berilgan quyidagi dastur bo‘yicha berilgan topshiriqlar kerakli fragmentlari asosida bajarilgan.</p> <p>☝ Bundagi dasturda giba ‘zio‘ zgartirishlari topish orqali topshiriqlar bosqichma bosqich amalga oshiriladi.</p>
dastur	topshiriqlar
<pre>// Created by MBBahodir #include "stdafx.h" #include &lt;vector&gt; #include &lt;algorithm&gt;  using namespace std;  inline ostream&amp; operator &lt;&lt;( ostream&amp; out, const vector&lt; unsigned &gt;&amp; obj ) {     cout &lt;&lt; "&lt; ";     for( auto&amp; p: obj )         cout &lt;&lt; p &lt;&lt; " ";     return out &lt;&lt; "&gt;"; }  void pow2( unsigned&amp; i ) { i *= i; }  int main( void ) {     const int examples = 4;     for( int i = 0; i &lt; examples; i++ ) {         unsigned ai[] = { 1, 2, 3, 4 , 5, 6, 7, 8, 9 },         ni = sizeof(</pre>	<p style="text-align: center;">1. Dastur dasturda global funksiyay aratilgan.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <p style="text-align: center;">2. Dastur dasturda global funksiyay aratilgan.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <p style="text-align: center;">3. for( auto&amp; p: obj ) fragmentni tushuntirib berilgan va anoli giniyozing.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>

```

ai ) / sizeof( ai[ 0 ]
);
    vector< unsigned
> vi( ai, ai + ni );
    cout << vi;
    switch( i ) {
        case 0:
            for_each(
vi.begin(), vi.end(),
pow2 );
                cout << "
=> " << vi << endl;
                break;
        case 1:
            for_each(
ai, ai + ni, pow2 );
                cout << "
=> " << vector<
unsigned >( ai, ai + ni
) << endl;
                break;
        case 2:
            for( auto&
i : ai ) pow2( i );
                cout << "
=> " << vector<
unsigned >( ai, ai + ni
) << endl;
                break;
        case 3:
            for_each(
vi.begin() + 2,
vi.end() - 2, pow2 );
                cout << "
=> " << vi << endl;
                break;
    }
}

```

---



---



---

4. Dasturningqaysifragmentlarini FIND() funksiyasibilanalmashtirishmumkin.

---



---



---

5. To‘plamdanberilganqiymatniiz lashdasturfragmentinianiqlangvao‘z izlashfunksiyasingizniyarating.

---



---



---



---



---



---



---



---



---



---



---



---



---



---



---



---



---



---



---



---



---




---

<pre>system("pause"); return 0; }</pre>	
<p>6. Dasturda jamiboʻlib, nechaoʻzgartirish kiritildi. _____</p> <p>7. Shudasturning analogini yaratishsiz gamustaqil vazifadir.</p>	



## BESHINCHI ASSISMENT TOPSHIRIG'I

	<p>Funktorlarning ko‘llanilishiga oid berilgan quyidagi dastur bo‘yicha berilgan topshiriqlar kerakli fragmentlari asosida bajarilgan.</p> <p>☝ Bundasturda gibabozlikni o‘zgartirishlarni topish orqali topshiriqlar bosqichma bosqich amalga oshiriladi.</p>
---	--

dastur	topshiriqlar
<pre>// ConsoleApplication1.cpp : main project file.  #include "stdafx.h" #include &lt;iostream&gt; #include &lt;string&gt; using namespace std; class myFunctorClass { private:     int _x;     char _ch; public:     myFunctorClass (int x) : _x( x ) {}     myFunctorClass (char ch): _ch(ch){}     int operator() (int y) { return _x + y; }     int operator() () {return _x*_x; }     void operator()(char ch, int a, int b) {     switch (ch)     {     case '+' : cout &lt;&lt;</pre>	<p>1. Dasturda nechta konstruktor funksiyalar yaratilgan.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <p>2. Dasturda xatolarni toping.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <p>3. operator() fragmentini tushuntirib bering va anoligini yozing.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>

<pre> a + b; break;     case '-' : cout &lt;&lt; a - b; break;     case '*' : cout &lt;&lt; a * b; break;     case '/' : cout &lt;&lt; ((b&gt;0) ? to_string(a/b) : "maxraj 0"); break;     case '%' : cout &lt;&lt; ((b&gt;0) ? to_string(a%b) : "maxraj 0"); break;     case '^': {         int ret = a;         while( b-- &gt; 1 ) ret *= a;         cout &lt;&lt; ret;     } break;     default: cout &lt;&lt; "amal to'g'rimas !!!";         break;     } } </pre>	<p>4. Massivning juft vatoqelem entlariyig'indisiniid f sonini hisoblovchi funkto'ryozing.</p> <hr/> <hr/> <hr/>
<pre> int operator()( int a, int b) {     switch (_ch)     {         case '+' : return a + b; break;         case '-' : return a - b; break;         case '*' : return a * b; break; </pre>	<p>5. To'plamdan berilgan qiymatni izlash funkto'ri funksiyasini g'zni yarating.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
<pre>         case '/' : return ((b&gt;0) ? a/b : 0); break;         case '%' : return ((b&gt;0) ? a%b : 0); break;         case '^': { </pre>	<p>6. Taqqoslash uchun funkto'ri funksiyasini yarating.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <p>7. Funkto'ri funksiyalarning g'ayisiuslubisiz g'ayoqdi. Dasturni to'liq shu uslubga almashiring.</p> <hr/> <hr/>

```

        int ret = a;
        while( b-- > 1
) ret *= a;
        return ret;
        } break;
        default: {cout <<
"amal to'g'rimas !!!";
return 0; }
        break;
    }
}
};

int main()
{
    myFunctorClass add( 5
);
    cout << add( 6 );
    cout << endl;
    myFunctorClass F(5);
    F('+',3,4);
    cout << endl << F() <<
endl;


    cout <<
myFunctorClass('^')(2,10);
    system("pause");
    return 0;
}

```

8. Dasturdajamibo'lib, nechao'zgartirishkiritildi. \_\_\_\_\_
9. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.

## 2-BOB. DASTURLASH TILINING TAKOMILLASHTIRILGAN IMKONIYATLARI

### 2.1. Standartalgoritmlarvaiteratorlar.

 C++ tilining standartalgoritmlar kutubxonasi, iteratorlar, xotirani taqsimlash talablari, dinamik xotira ajratish va foydalanishga asosida qidirish, saralash, taqqoslash funksiyalari, dasturlash da o'zlarini tutishlari, iteratorlar, o'z garmas iteratorlar, teskari iteratorlar bilan ishlashga mo'ljallangan asosiy funksiyalar va ularning dasturlashdagi ahamiyati, xotirani taqsimlovchi sinflarni yaratish va talablari, turli tipli massivlar uchun dinamik xotira ajratish va foydalanish operatorlar uslublari, amallari, talablari, vazifalar va usullarini keltirilgan bo'lib, nazariy bilimlarni asoslash uchun dasturlar tuzib ko'rsatilgan.

Bilimlarni mustahkamlash uchun 30  
tanazariy savol va amaliy ko'nikma va malakalarni rivojlantirish uchun 3  
ta assistent topshiriq va har assistentda 10(9,7) ta topshiriq, jami  
26 ta topshiriq berilgan.

Yuqoridagi topshiriqlarni bajarish mavzuni mustahkamlash uchun xizmat qiladi.

***✍️ Kalit so'zlar.*** Algoritm, kutubxona, iterator, o'z garmas iterator, teskari iterator, qidirish, saralash, o'rinalmashtirish, shablon, sinf, find(), sort(), swap(), begin(), end(), rbegin(), rend(), crbegin(), crend(), xotira, pul, xotira bloki, new, delete, for\_each(), funksional ob'ektlar.

***✔️ Bilish shart bo'lgan tushunchalar.*** Tiptushunchasi, sinf va sinf ob'ekti, meros xo'rsinfi, to'plam, statik va dinamik massiv, element gamurojaat, funksiyaviy ko'rsatkich, konteyner, xesh, istisno holat, ma'lumotlarni kirish va chiqishi, dasturlashga oid dastlabki tushunchalar va C++ tilini qo'llab quvvatlovchi muhitda ishlashni bilish lozim.

***👉 Bilib olinishi.*** Inspektorlar, mutatorlar, koordinatorlar, find() algoritmi, find\_if() algoritmi, count() va count\_if() algoritmlari, sort() algoritmi, <functional> kutubxonasi, for\_each() algoritmi, algoritmlarning bajarilish tartibi, iterator amallari, iteratorlarning qo'shima amallari, xotirani taqsimlovchilar, new() operatori, delete() operatori, xotirani ishlashni tezlashtirish usullari, pul ob'ektlari, konteyner va uni xotirada qo'llanilishi, standart bo'yicha taqsimlovchilar usullarini o'rganishingiz mumkin.

### REJA

#### 1. Standartalgoritmlar.

2. Iteratorlar va ularning qoʻllanilishi.
3. Xotira ni taqsimlovchilar va ularga qoʻyilgan talablar.
4. Standart boʻyicha taqsimlovchi.

### **KIRISH**

Bugungi kundasturlash juda sodda hisob bormoqda, baʼzi birmutaxassislar dasturlash uchun matematik kerakmas deb aytsa, baʼzilar matematika dasturlashning asoside baytadi.

Shuning negizidan imayotibdi. Albatta algoritmlar, iteratrlar va xotira ni taqsimlovchilar va ularning talablari yotadi.

Algoritmlar tayyob oʻlsa,

faqat bu algoritmlarini joy joyiga qoʻyish bu juda osondir.

Ammo algoritmlar ni yaratish va uni dasturlash joriy qilish oʻtamushkul va mas haqqatlimasadir.

Shuning uchun bugungi kundajuda koʻplab algoritmlar ni saqlovchi standart va an standart kutubxonalar mavjud.

**Standart algoritmlar.** Yangi dasturlashni boshlanuvchilar odatda saralash, qidirish yoki qator elementlarini sanash kabinisbatan oddiy vazifalarni bajarish uchun maxsus takrorlanishga asoslangan usullar ni yozish uchun juda koʻp vaqt sarflashadi. Bu usullar dasturda qoʻllanilish qanchalik osonligi jihatidan ham, umumiy ishonchlilik va mavjudlik nuqtainazaridan ham muammoli boʻlishi mumkin, chunki bu usullar tushunish qiyin boʻlishi mumkin.

Qidiruv,

sanash va saralash dasturlashda juda keng tarqalgan amallar boʻlganligi sababli, C++

standart kutubxonasi dastlab bu vazifalarni bir necha qator kod bajaradigan katta funksiyalar toʻplamini oʻz ichiga olgan. Bundantashqari,

buxususiyatlar oldindan sinovdan oʻtgan,

samarali va turlilik konteynerlarni qoʻllab-quvvatlaydi.

Buxususiyatlarni ham paralel qoʻllab

quvvatlashtirish uchun bir xil masala uchun bir necha MP

(markaziy protsessor) oqimlarini ajratish qobiliyatiga.

Algoritmlar kutubxonasi tomonidan taqdim etiladigan funksiyalar odatda uchtoifadan birigakiradi:

**Inspektorlar** (nazoratchilar) - konteynerdagima'lumotlarni (masalan, qidirish yoki elementlarni isonihisoblash amallari) koʻrish uchun ishlatiladi.

## Mutatorlar

konteynerdagima'lumotlarnio'zgartirishuchunishlatiladi (masalan, saralashyokielementlarniqaytatartibgasolishamallari).

## Fasilitatorlar

## (koordinatorlar)

elementlariningqiymatlariasosidanatijahosilqilishuchunishlatiladi (masalan,

qiymatlarniko'paytiruvchiob'ektlaryokielementlarningqaysitartibjuftlikl aridatartiblanishikerakliginianiqllovchiob'ektlar).

Bualgoritmlaralgoritmkutubxonasida

(`<algorithm>`sarlavhafaylida)

joylashgan.

Oldingibiladiganvaengko'ptarqalganalgoritmlardanba'zilariniimkoniyatl arinichuquroqko'ribchiqamizvabualgoritmlarningbarchasiiteratorlardanf oydalanadi.

`find()`

algoritmi

berilganqiymatbo'yichaelementlarniqidirishalgoritmi.

Berilganqiymatnibirinchitopgunichaishlaydi. Argumentsifatida 3 taparamertnioladi.

ketma-ketlikdaboshlang'ichelementuchuniterator;

ketma-ketlikdaoxirgielementuchuniterator;

qidirishningqiymati.

Qidirishnatijadaqidirilayotganqiymatlielementga (agarutopilsa) yokiiteratorningoxiriga (agarbundayelementtopilmasa)

ishoraqiluvchiiteratorqaytadi.

5.1-dastur. Find () funksiyasidanfoydalanish.

```
// Created by MBBahodir
#include "stdafx.h"

#include <algorithm>
#include <array>
#include <iostream>

using namespace std;
int main(){
    array<int,10> arr = {1, 3, 5, 7, 9, 11, 13, 15,
17, 19 };

    cout << "Qidirish va o'zgartirish uchun son
kirit: ";
    int search;
```

```

int replace;
cin >> search >> replace;

// array<int,10>::iterator found;

    auto found = find(arr.begin(), arr.end(),
search);

    if (found == arr.end())
    {
        cout << search << " topilmadi. " << endl;
    }
    else
    {
        *found = replace;
    }

    for (int i : arr)
    {
        cout << i << ' ';
    }

    cout << endl;
    system("pause");
    return 0;
}

```

#### 5.1-dastur.Output

##### Birinчисinov

Qidirish va o'zgartirish uchun son kirit: 19  
21  
1 3 5 7 9 11 13 15 17 21

##### Ikkinчисinov

Qidirish va o'zgartirish uchun son kirit: 10  
25  
10 topilmadi.  
1 3 5 7 9 11 13 15 17 19

find\_if() algoritmi - shartorqalielementniizlash. Judako'phollardakonteynerdamavjudelementlarorasidama'lumbirqiyamat saqlaydiganelementbormideganmasaladuchkelamiz. Masalan, konteynerdasatrlielementlarberilganbo'lsa, birorbirsatrqismibormikandeganmasalabo'lsin.

Manashundaymasalalarniyechishdafind\_if() algoritmiengmosvaqulayhi soblanadi. Bualgorithm find()

algoritmibilananalogsifatidaishlaydiammo, qidirishuchunqiymato'rnigafunksiyako'rsatgichlarigamoslanganelementqiyamatidao'xshashlikborbo'lganob'ektchaqiriladi.

find\_if() funksiyasibuninguchunharbirelementnichaqiradivatoshunday elementnitopilmagunchayokikonteynerdaelementlartugaguncha.

Ma'lumbirberilgansatrbo'yicha, berilgankonteynerdashusatrnio'zidaqismsatrqilibsaqlovchielemeniborkiy o'qliginianiqlovchidasturnituzishkerakbo'lsin.

5.2-dastur. find\_if() funksiyasidanfoydalanish.

```
// Created by MBBahodir
#include "stdafx.h"
#include <algorithm>
#include <array>
#include <iostream>
#include <string>

using namespace std;

bool containsNut(string str)
{
    return (str.find("oro") != string::npos);
}

int main()
{
    array<string, 4> arr = { "Toshkent", "Buxoro",
        "Samarqand", "Navoiy" };

    auto found = find_if(arr.begin(), arr.end(),
        containsNut);

    if (found == arr.end())
```



```

    {
        cout << " Topilmadi" << endl;
    }
    else
    {
        cout << " Topildi: " << *found << endl;
    }
    system("pause");
    return 0;
}

```

Agarodatdagistandardtarzdagidekmuammonihalqilshganimizda, kamidaikkitakrorlanishkerakedi (birinchitakrorlanishelementlarnisaralashuchunvaikkinchisisatrqisminisolishtirishuchun). Standart C++ kutubxonasi ningvazifalaribirnechtafragmentsatrlaridahamshundayqilishimkoniniberadi.

count() va count\_if() algoritmlari – konteynerdanberilganelementlarinisana shvazifasinibajaradi. Berilganmezon (shart) bo'yichabarchaelementlarnitekshirib, sanaydi. Berilgankonteynerdanechtaelementberilgansatrqismiborliginisana shdasturini keltiramiz.

5.3-dastur. count() va count\_if() algoritmlaridanfoydalanish.

```

// Created by MBBahodir
#include "stdafx.h"
#include <algorithm>
#include <array>
#include <iostream>
#include <string>
#include <vector>
using namespace std;

bool containsNut(string str)
{
    return (str.find("daryo") != string::npos);
}
bool counter(string str){
    return true;
}

```

```

int main()
{
    array<string, 6> arr = { "Sirdaryo",
    "Amudaryo", "Qashadaryo", "Suxondaryo",
    "Zarafshon", "Qashadaryo" };

    auto countWithIf = count_if(arr.begin(),
arr.end(), containsNut);
    int countofArr = count(arr.begin(), arr.end(),
"Qashadaryo");
    cout << "daryo satr qismi borlarining soni: "
<< countWithIf << endl;
    cout << " elementlar soni: " << countofArr <<
endl;

    system("pause");
    return 0;
}

```

5.3-dastur. Output

```

daryo satr qismi borlarining soni: 5
elementlar soni: 2

```

sort()

algoritmi

foydalanuvchining saralash funksiyasini qurish uchun ishlatiladi.

Oldinlarisort() funksiyasibilan massivlarni saralagan bo'lishingiz mumkin, ammosort() funksiyabubilan chegaralanib qolmaydi.

Bu funksiya yordamchi bir funksiya o'ladi va bu argumentasosida foydalanuvchi o'zining saralash uslubini amalga oshirishim mumkin.

Funksiya esataqqoslash uchun paramertikka parametroladi.

Agar birinchi element ikkinchisidan oldin bo'lishi kerak bo'lsa, ikkinchi paramert true qiymat qaytarishi kerak. Odatda sort() funksiyasio'z tartibidasaralaydi.

Berilgan massivni kamayish tartibidasaralash dasturini tuzaylik.

5.4-dastur. sort() funksiyasidan foydalanish.

```

#include "stdafx.h"
#include <algorithm>
#include <array>

```

```

#include <iostream>

using namespace std;
bool greater(int a, int b)
{
    return (a > b);
}
bool d_greater(int a, int b)
{
    return (b > a);
}

int main()
{
    array<int,5> arr = { 28, 18, 16, 25, 25 },
    arr_one;

    arr_one = arr;

    sort(arr.begin(), arr.end(), greater);
    sort(arr_one.begin(), arr_one.end(),d_greater );
    for (int i : arr) {
        cout << i << ' ';
    }
    cout << endl;
    for (int i : arr_one ) {
        cout << i << ' ';
    }
    cout << endl;

    system("pause");
    return 0;
}

```

#### 5.4-dastur. Output

```

28 25 25 18 16
16 18 25 25 28

```

Dasturdanko‘rinadiki,  
o‘zimizsaralashalgoritminiboshidanyozishimizshartemasekan,  
buninguchunbirqatorfragmentyetarlibo‘ldi. AmmoC++tili,  
judako‘psaralashningkamayishtartibdafoydalaniganligiuchungreater{ } ti  
plifunksiyasibo‘lishimumkin (<functional>kutubxonasidabor).  
Bizo‘zimizninggreaterfunksiyasidanfoydalanishuchunsort(arr.begin(),ar  
r.end(),greater);dasturfragmentiniyozdik,  
ammokutubxonanikinichaqirishuchunsort(arr.begin(),arr.end(),std::great  
er{ });kabifragmentyozishimizkerak. E‘tiborbering,  
greater{ } funksiyasifiguraliqavsgamuxtoj,  
shuninguchununifunksiyadebatalmaydi,  
ma‘lumottipidebyuritiladivaundanfoydalanishuchunnusxasiniyaratishi  
mizkerak.

Figuraliqavslaranonimob’ektlarniyaratishuchunxizmatqiladi.

for\_each() algortmi – konteynerningbarchaelementlaribo‘tish.  
Bufor\_each()

funksiyasikiruvchima‘lumotsifatidato‘plamniqabulqiladivafoydalanuvch  
iningfunksiyasinijoriyto‘plamdagiharbirelementuchunqabulqiladi.

Buningqulayliktomonishundaki,

agarbiramalnito‘plamningbarchaelementlaribo‘yichabajarishkerakbo‘lsa,  
for\_each() funksiyasiasqotadi.

Berilganmassivningbarchaelementlarinichiqarishuchun for\_each()  
funksiyasidanfoydalanishnidasturini keltiraylik.

5.5-dastur. for\_each() funksiyasidanfoydalanish.

```
#include "stdafx.h"

#include <algorithm>
#include <array>
#include <iostream>

using namespace std;

void doubleNumber(int &i){
    i = i;
}

int main(){
    array<int,4> arr = { 1, 2, 3, 4 };
}
```

```

    for_each(arr.begin(), arr.end(), doubleNumber);

    for (int i : arr){
        cout << i << ' ';
    }

    cout << endl;
    system("pause");
    return 0;
}

```

Yosh dasturchilar uchun buta takrorlanish jarayonini oddiy for operatoriga aniq parametrlarni qo'yib ishlatish maquldir. for\_each() funksiyasining yutuqlaridan iboratki, takrorlanish tanasidan qayta foydalanish va paralellashtirish, kattaproektlardakattama'lumotlar, turli tiplari har xil ma'lumotlarni tuzilmasi bilan ishlangandan eng yaxshi ishlash natijasi bo'lib xizmat qiladi.

**Algoritmning bajarilish tartibi**

.Algoritm kutubxonasida eng ko'p algoritm lar ma'lum bir tartibda kiritilgan bo'lib, ularni bajarish tartibini o'zgartirish mumkin.

Bunday algoritm lar uchun har qanday belgilangan funksiyalar bajarilish tartibini o'zgartirish mumkin. Quyidagi algoritm lar aniq bajariladi:

```

for_each()
copy()
copy_backward()
move()
move_backward()

```

Agar parametrlar sifatida biror funksiya berilmagan bo'lsa, aniq qilingan holatda. Buning uchun birinchi algoritmda arr.begin() va arr.end() aniq berilishi kerak. Ammo C++20 standartida faqat massiv berilsa o'z qilinganadi.

Algoritm kutubxonasining algoritm larini o'zgartirish:

<b>1</b>	<b>O'zgartirilmaydigan ketma-ketlik amallari:</b>	
1.	all_of	to'plamdagibarchaelementlar uchun holat shartini tekshirish
1.	any_of	to'plamdagiharqandayelement uchun holat shartini bajarish

2		hinitekshirish
1. 3	none_of	to‘plamdagihechbirelementuchunholatshartgajavobbermasliginitekshirish
1. 4	for_each	funksiyanito‘plamgaqo‘llash
1. 5	find	to‘plamdaqiymatinitopish
1. 6	find_if	to‘plamdashartnibajaruvchielementnitopish
1. 7	find_if_no t	to‘plamdashartnibajarmaydiganelementnitopish
1. 8	find_end	to‘plamdagioxirgielementnitopish
1. 9	find_first_of	to‘plamdagielementlardanbirinchisinitopish
1. 10	adjacent_find	to‘plamdatengqo‘shnielementlarnitopish
1. 11	count	to‘plamdagielementlarsoninihisoblash
1. 12	count_if	shartgajavobberadiganto‘plamdagielementlarsoniniqaytaradi
1. 13	mismatch	ikkifarqliintervallarnibirinchio‘rniniqaytaradi
1. 14	equal	to‘plamdagiIkki elementlartengliginitekshirish
1. 15	is_permutation	to‘plamdaalmashtirishboryokiyo‘qliginitekshirish
1. 16	search	ketmk-ketlikdanqidirish
1. 17	search_n	intervalndanqidirish

<b>2</b>	<b>Ketma-ketlikamallarini o'zgartirish:</b>	
2. 1	copy	to'plam elementlardan nusxa olish
2. 2.	copy_n	elementlarni ko'paytirish
2. 3	copy_if	shart asosida diapazon elementlarini nusxalash
2. 4	copy_backward	elementlarni to'plamdan orqaga nusxa ko'chirish
2. 5	move	elementlar qatorini ko'chirish
2. 6	move_backward	elementlarni to'plamdan orqaga ko'chirish
2. 7	swap	ikki ob'ektning qiymatlarini almashtirish
2. 8	swap_ranges	ikki intervalning qiymatlarini almashtirish
2. 9	iter_swap	ikki iteratorlarning ko'rsatikich ob'ektlari qiymatlarini almashtirish
2. 10	transform	to'plam ob'ektlarini aylantirish
2. 11	replace	to'plamning qiymatlarini almashtirish
2. 12	replace_if	Qiymatlarni shart asosida to'plamda almashtirish
2. 13	replace_copy	almashtirish qiymatlarini to'plamni nusxalash
2. 14	replace_copy_if	almashtirish qiymatlarini to'plamni shart asosida nusxalash
2. 15	fill	to'plamni qiymat bilan to'ldirish
2. 16	fill_n	ketma-ketligini qiymat bilan to'ldirish

6		
2. 1 7	generate	funksiyabilanto‘plamuchunqiymatlarniyaratish
2. 1 8	generate_n	funksiyabilanketma-ketlikuchunqiymatlarnihosilqilish
2. 1 9	remove	to‘plamqiymatinio‘chirish
2. 2 0	remove_if	to‘plamma’lumotlarshartasisidao‘uchirish
2. 2 1	remove_cop y	qiymatnio‘chirishdiapazonininusxalash
2. 2 2	remove_cop y_if	qiymatnio‘chirishdiapazoninishartasosidanusxalash
2. 2 3	unique	to‘plamdagitakrorlanganelementlarnio‘chirish
2. 2 4	unique_cop y	to‘plamdagitakrorlanganelementlarninusxalash
2. 2 5	reverse	teskarito‘plam
2. 2 6	reverse_cop y	teskarito‘plamninusxalash
2. 2 7	rotate	to‘plmdaelementlarningchapgaaylantirish
2. 2 8	rotate_copy	to‘plmdaelementlarningchapgaaylantirishninusxalash
2. 2	random_shuffle	to‘plmdagielementlarningtasodifiyqo‘yish



9		
2. 3 0	shuffle	to‘plamdagi elementlarning generator yordamida tasodifiy qo‘yish
<b>3</b>	<b>Bo‘laklar (qismlar) bilan amallari</b>	
3. 1	is_partitioned	to‘plamning bo‘linganligini tekshirish
3. 2	partition	ikkigabo‘lish orlig‘i
3. 3	stable_partition	qonuniyatasosida ikkigabo‘lish orlig‘i
3. 4	partition_copy	bir to‘plamni ikkigabo‘lish
3. 5	partition_point	bo‘lish nuqtasini olish
<b>4</b>	<b>Saralash amallari:</b>	
4. 1	sort	to‘plamda elementlarni saralash
4. 2	stable_sort	ekvivalent tartibini saqlab qolgan holda elementlarni saralash
4. 3	partial_sort	to‘plamda elementlarni qisman saralash
4. 4	partial_sort_copy	to‘plamda elementlarni qisman saralash va nusxalash
4. 5	is_sorted	to‘plamning tartiblanganligini tekshirish
4. 6	is_sorted_until	to‘plamda ajratilmagan elementni topish
4. 7	nth_element	to‘plamda elementni saralash
<b>5</b>	<b>Binarqidiruv (bo‘lingan / tartiblangan intervallar bilan ishlash):</b>	
5. 1	lower_bound	iteratori ni quyichegarasini qaytarish
5. 2	upper_bound	iteratori ni yuqorichegarasini qaytarishga qaytarish
5. 3	equal_range	teng elementlardan to‘plamni olish
5.	binary_search	qiymatning tartiblangan ketma-

4	arch	ketlikdamavjudlignitekshiring
<b>6</b>	<b>Birlashtirish (tartiblanganintervallar bilan ishlash):</b>	
6.1	merge	tartiblangan to'plamlarni birlashtirish
6.2	inplace_merge	ketma-kettartiblangan intervallarni birlashtirish
6.3	includes	tartiblangan to'plamlarni boshqatartiblangan to'plamni o'z ichiga olish yoki o'lmasligini tekshirish
6.4	set_union	ikkitartiblangan to'plamni birlashtirish
6.5	set_intersection	ikkitartiblangan intervallarning kesishishi
6.6	set_difference	ikkitartiblangan intervallar orasidagi farq
6.7	set_symmetric_difference	ikkitartiblangan intervallar orasidagi simmetrik farq
<b>7</b>	<b>Elementlari yig'indisi amallari:</b>	
7.1	push_heap	elementlari yig'indisini qatoriga surish
7.2	pop_heap	elementlari yig'indidan elementni o'chirish
7.3	make_heap	elementlari yig'indisini yasash
7.4	sort_heap	elementlari yig'indisini tartiblash
7.5	is_heap	elementlari yig'indisini tekshirish
7.6	is_heap_until	elementlari yig'indidan birinchi sinitopish
<b>8</b>	<b>Kichik va katta amallari:</b>	
8.1	min	eng kichik elementni qaytaradi
8.2	max	eng katta elementni qaytaradi
8.3	minmax	eng kichik va eng katta elementni qaytaradi
8.4	min_element	to'plamdagi eng kichik elementni qaytaradi

8.5	max_element	to'plamdagi eng katta elementni qaytaradi
8.6	minmax_element	to'plamdagi eng kichik va eng katta elementni qaytaradi
<b>9</b>	<b>Boshqayordamchiamallar:</b>	
9.1	lexicographical_compare	leksikografik jihatdan qiyoslash
9.2	next_permutation	to'plam aylantirish uchun keyingiideks
9.3	prev_permutation	to'plam aylantirish uchun oldingiideks

Buyerda 85 ta algoritmlar keltirilgan, ammobularning hammasi shablon funksiyalar bo'lganligi uchun C++ dasturlash versiyalarida farq qilib o'lish mumkin. Shuningdek, eslab ko'ring, bu algoritmlarning ko'plaridan oldin mavzularimizda foydalanganmiz.

**Iteratorlar va ularning qo'llanilishi.** Iteratorlar konteynerlarning elementlariga murojaat qilish uchun foydalaniladi.

Iteratorlar bilan elementlar bilan ishlash juda qulay hisoblanadi. Iterator iterator tipida yoziladi.

Har qanday konteyner uchun iteratorlarning tiplari farq qiladi. Masalan, `list<int>` tipidagi konteyner uchun `list<int>::iterator` tipi, `vector<int>` tipidagi konteyner uchun esa `vector<int>::iterator` tipi ishlatiladi.

C++ tilidagi konteynerlardan iteratorlarni ajratib olish uchun `begin()` va `end()` funksiyalaridan foydalaniladi.

`begin()` funksiyasi konteynerdagi birinchi element ko'rsatuvchi iteratorni qaytaradi (agarkonteynerda elementlar bo'lsa).

`end()` funksiyasi konteynerdagi oxirgi elementdan keyingi pozitsiyani ko'rsatuvchi iteratorni (ya'ni konteyner oxirini) qaytaradi.

Agarkonteyner bo'sh bo'lsa,

`begin()` va `end()` funksiyalaribir xil qiymat qaytaradi. Agar

`begin()` va `end()` funksiyalar o'zaroteng bo'lmasa, ularning orasidankamida bitta element bor.

Bu funksiyalarani qanday konteyner uchun iterator qaytaradi.

Iterator yaratish misoli:

```
std::vector<int> v = { 1, 2, 3, 4 };
std::vector<int>::iterator iter = v.begin();
```

Bu misolda tipiga mansub bo'lgan vektor tipidagi konteyner vektor yaratilgan. Konteyner doimiy qiymatlar bilan to'ldirilgan. `begin()`

funksiyasi (usuli) bilan vektor elementini olish uchun iterator keltirilgan. Bu iterator vektor konteynerning birinchi elementini ko'rsatadi.

Iterator mallari:

- **\*iter** – iterator ko'rsatadigan elementni olish;

- **++iter** -

keyingi elementga murojlat qilish uchun iteratorni harakatlantirish

- **--iter**

oldingi elementga murojlat qilish uchun iteratorni harakatlantirish.

forward\_list konteyner iteratorlaridekrement amalini qo'llab quvvatlamaydi.

- **iter1 == iter2** - ikki iterator teng, agarular bir xil iteratorni aniqlagan bo'lsa.

- **iter1 != iter2** ikki iterator teng emas, agarular bir xil iteratorni aniqlagan bo'lsa.

5.6-dastur. Iterator mallaridan foydalanish.

```
#include "stdafx.h"
#include <iostream>
#include <vector>
using namespace std;
int main()
{
    vector<int> myvector;
    for (int i = 0; i < 15; i++)
    {
        myvector.push_back(rand() % 100);
    }
    // vector<int>::iterator iter =
myvector.begin();
    auto iter = myvector.begin();
    while(iter != myvector.end())
    {
        cout << *iter << " | ";
        ++iter;
    }
    cout << endl;
    system("pause");
    return 0;
}
```

Konteynerlar bilan ishlaganda, konteynerdagi elementlarni qo‘shish yoki o‘chirish hukmi konteyner uchun baʼzi hujjatdagi iteratorlarni, shuningdek, uning elementlariga havolavako‘rsatgichlarni bekor qilishga olib kelishi mumkin. Iteratorlarni faqat elementlarni o‘lish, balki ularni o‘zgartirishimkin beradi.

5.7-dastur.

Iterator yordamida konteynerni elementlari qiymatini o‘zgartirish.

```
#include "stdafx.h"

#include <iostream>
#include <vector>
using namespace std;
int main()
{
    vector<int> myvector;
    for (int i = 0; i < 15; i++)
    {
        myvector.push_back(rand() % 10);
    }
    vector<int>::iterator iter = myvector.begin();
    //auto iter = myvektor.begin();
    while(iter!=myvector.end())
    {
        *iter = (*iter) * (*iter);
        ++iter;
    }

    for(iter = myvector.begin();
iter!=myvector.end(); ++iter)
    {
        cout << *iter << " | ";
    }
    cout << endl;
    system("pause");
    return 0;
}
```

## 5.7-dastur. Output

```
1 | 49 | 16 | 0 | 81 | 16 | 64 | 64 | 4 | 16 | 25 | 25 | 1 | 49 | 1 |
```

Dasturda `while` takrorlanishoperatorikonteynerningelementlariniiteratorogaolib, o'zini'zigako'paytirib, yanashuiteratorogayozadi. Shuninguchunkrangakonteynerdagisonlarningkvadratlarichiqadi.

**O'zgarmasiteratorlar.** Agarkonteynerdao'zgarmasqiymatlielementlarbo'lsa, buholdakonteynererelementlarigamurojaatqilishuchuno'zgarmasiteratorlar danfoydalanishshkerak. Buninguchun `const_iterator` tipiishlatiladi. Buiteratorlarfaqatelementlarnisanashimkoniniberadi. O'zgartirishmumkinemas.

```
vector<int>::const_iterator iter ;

    for(iter = myvector.begin();
iter!=myvector.end(); ++iter)
    {
        cout << *iter << " | ";
        /*iter = (*iter) * (*iter);
    }
}
```

O'zgarmasiteratorningqiymatlariniolishuchuncbegin() va cend() funksiyalarishamishlatiladi.

Agariteratoro'zgarmasdebolinmaganbo'lsaham, bufunksiyalarunio'zgarmasqilibberadi. o'zgartirishmumkinemasbo'ladi.

Shungdek,

## 5.8-dastur. O'zgarmasiteratorndanfoydalanish.

```
// ConsoleApplication1.cpp : main project file.

#include "stdafx.h"
#include <iostream>
#include <vector>
using namespace std;

int main(){
    int myints[] = {16,2,77,29,28};
    const vector<int> myvector (myints, myints +
sizeof(myints) / sizeof(myints[0]) );
    vector<int> myvector_one;
    for (int i = 0; i < 15; i++)
```

```

        myvector_one.push_back(rand() % 10);

        vector<int>::const_iterator it;
        for (it = myvector.begin(); it !=
myvector.end(); ++it)
            cout << ' ' << *it;
        cout << endl;
        for (it = myvector_one.cbegin(); it !=
myvector_one.cend(); ++it)
            cout << ' ' << *it;
        cout << endl;
        system("pause");
        return 0;
}

```

5.8-dastur.Output

```

16 2 77 29 28
1 7 4 0 9 4 8 8 2 4 5 5 1 7 1

```

**Teskari iteratorlar.** Bu iteratorlar konteynerning elementlarini teskari olishni amalga oshiradi. Buning uchun `reverse_iterator` tipi ishlatiladi. Bu tipdagi iteratorlarga konteynerning elementlarini olish uchun `rbegin()` va `rend()` funksiyalaridan foydalaniladi.

```

#include "stdafx.h"
#include <iostream>
#include <vector>
using namespace std;
int main()
{
    int myints[] = { 1, 2, 3, 4, 5 };
    vector<int> v (myints, myints+
sizeof(myints)/sizeof(myints[0]));
    for (vector<int>::reverse_iterator iter =
v.rbegin(); iter != v.rend(); ++iter)
        std::cout << *iter << " | ";

    cout << endl;
    system("pause");
}

```

```
return 0;
}
```

Agarkonteynernio‘zgartirishdanhimoyayaqilishvateskarisikerakbo‘lsa, const\_reverse\_iterator tipidagiiteratordanvauningcrbegin() va crend()funksiyalaridanfoydalanishmaqsadgamuvofiqdir. Yuqoridagidasturgaquyida gifragmentniyozishyetarlibo‘ladi.

```
for (vector<int>::const_reverse_iterator iter = v.crbegin(); iter !=
v.crend(); ++iter)
std::cout << *iter << " | ";
```

Shuningdekiteratorlariqo‘shimaamallargahamega (list va forward\_list konteynerlaridantashqari):

- **iter + n** - n tapozitsiya keyingielementniko‘rsatuvchi iteratorniqaytaradi.
- **iter - n** - n tapozitsiya oldingielementniko‘rsatuvchi iteratorniqaytaradi.
- **iter += n** - iteratorni n - chipozitsiyagao‘tkazadi
- **iter -= n** - iteratorni n - chipozitsiyagao‘tkazadi
- **iter1 - iter2** - iter1 va iter2 larorasidagipozitsiyalar soniniqaytaradi
- [**>**], [**>=**], [**<**], [**<=**] -taqqoslashamallari.

Agar iterator oxirigayaqinelementniko‘rsatsaboshqasidankatta.

5.9-dastur. Iteratorning qo‘shimcha amallaridan foydalanish.

```
#include "stdafx.h"
#include <iostream>
#include <vector>
using namespace std;
int main()
{
    int myints[] = { 1, 2, 3, 4, 5 };
    vector<int> v (myints, myints+
sizeof(myints)/sizeof(myints[0]));
    for (auto i = v.begin(); i < v.end(); i++)
    {
        cout << *i << " | ";
    }
    cout << endl;
    auto it1 = v.begin();
    auto it2 = it1 + 2;
    cout << "it1 + 2 => " << *it2 << endl;
```



```

    auto it3 = v.end() - 3;
    cout << "it1 - 3 => " << *it3 << endl;
    it2 += 2 ;
    cout << "it2 + 2 => " << *it2 << endl;
    it3 -= 2;
    cout << "it3 - 2 => " << *it3 << endl;
    cout << "it2 - it1 => " << it2 - it1 << endl;
    cout << "(it1 < it2) => " << (it1 < it2) <<
endl;
    cout << "(it1 >= it2) => " << !(it1 < it2) <<
endl;
    cout << "(it1 != it2) => " << !(it1 == it2) <<
endl;
    cout << "(it1 == it2) => " << (it1 == it2) <<
endl;
    cout << "(it1 != it2) => " << (it1 != it2) <<
endl;
    cout << endl;
    system("pause");
    return 0;
}

```

#### 5.9-dastur.Output

```

1 | 2 | 3 | 4 | 5 |
it1 + 2 => 3
it1 - 3 => 3
it2 + 2 => 5
it3 - 2 => 1
it2 - it1 => 4
(it1 < it2) => 1
(it1 >= it2) => 0
(it1 != it2) => 1
(it1 == it2) => 0
(it1 != it2) => 1

```

Taqqoslashamallari uchun ko'chirib olingan iteratorlar va bazaviy iteratorlar uchun ba'zi C++ dasturlashdagi farqlar bor.

5.1-jadval. Taqqoslashamallari yozish usullari.

Ko'chirib olingan iteratorlar uchun.	Ekvivalent bazaviy iteratorlar uchun.
--------------------------------------	---------------------------------------

move_ita == move_itb	basic_ita == basic_itb
move_ita != move_itb	!(basic_ita == basic_itb)
move_ita < move_itb	basic_ita < basic_itb
move_ita <= move_itb	!(basic_itb < basic_ita)
move_ita > move_itb	basic_itb < basic_ita
move_ita >= move_itb	!(basic_ita < basic_itb)

**Xotirani taqsimlovchilar va ularga qo‘yilgan talablar.** Dinamik xoti rabilanishlashda maxsus tuzilishlarni, fragmentlardan foydalanish bo‘lmasa, ko‘piga algoritmlardan foydalanish samarasiz bo‘lishi mumkin.

Bunga misol sifatida ikki holatni ko‘rib chiqamiz. `new` va `delete` operatorlarining qayta aniqlanib yuklanishida sintaktik konstruktorlarni kichik roq bo‘ladi va dastur lokalizatsiya qilish o‘ddiy bo‘ladi.

Shuningdek protsessordagi amallarni tizimida ham qayta aniqlash sodir bo‘ladi.

Avvalo, xotirani ishlatishni tezlashtirishda aqlli allocator qanchalik foydasiz bo‘lgini aniqlash lozim.

Buning uchun o‘ddiy test misollarini (C++ va S#) tillarida ko‘rib chiqaylik ( bu xotirani ishlatish uchun yaxshi menejer hisoblanadi va ob’ektlarni avlodlarga ajratadi, turli o‘lchamdagi ob’ektlar uchun turli pullardan (joylar) foydalanadi).

```
C++ dagidastur
class Node {
public:
    Node* next;
};
// ...
for (int i = 0; i < 10000000; i++) {
    Node* v = new Node();
}
```

```
S# dagidastur
class Node
{
    public Node next;
}
// ...
for (int l = 0; l < 10000000; l++)
```

```
{
    var v = new Node();
}
```

Dasturlarsharsimonvakumusuliniqo‘llabhamvaqtbo‘yichasolishtiril sa 10 barovar (62 ms va 650 ms) farqbo‘lmoqda. Shuningdek C#ishinitugutdi, ammoC++daesayaxshiginixotiraaajratilan, buhalio‘chirishkerak.

**Pulob’ektlari.** Pulob’ektlariuchunaniqechim - OSdankattaxotirablokiniolishvaunitengbloklilo‘lchamlargabo‘lishdir sizeof(node).

Xotiraniajratishdabloknipuldanolinadivaunibo‘shatishdaunipulgaqaytaril adi. Pulnitashkilqilishningengosonusuli - bittaaloqastekni (stack) ishlatishdir.

Maqsaddasturdaminimalqaytaaniqlashlarbo‘lganiuchun, BlockAlloc sinfnimerosxo‘risifatida Node sinfinitanlashdir.

```
class Node : public BlockAlloc<Node>
```

Avvalo, OSyoki C-runtime dankattabloklaruchunbirpulolishkerak. Buni malloc va free funksiyalaribilantashkilqilishmumkin, lekinkattasamaradorligiuchun (ortiqchaabstraksiyabosqichinio‘tkazishuchun), Virtualloc/Virtuallfreye funksiyalaridanfoydalanishmaqsadgamuvofiq. Bu vazifalar 4 karralibo‘lganbloklargaxotiraaajratadiva 64 karralibo‘lganbloklarjarayonmanziliuchunoraliqsaqlabturadi. Birvaqtningo‘zidacommit va reserve imkoniyatlariko‘rsatilganholdayanabirdarajadako‘tarilib, manziloralinqnirezervlashvabittachaqirishbilanxotirabloklariniajratamiz. PagePoolsinfidasturfragmenti.

```
inline size_t align(size_t x, size_t a) { return
((x-1) | (a-1)) + 1; }
//#define align(x, a) (((x)-1) | ((a)-1)) + 1)

template<size_t PageSize = 65536>
class PagePool
{
public:
    void* GetPage() {
        void* page = VirtualAlloc(NULL,
        PageSize, MEM_COMMIT | MEM_RESERVE,
```

```

PAGE_READWRITE);
        pages.push_back(page);
        return page;
    }

    ~PagePool() {
        for (vector<void*>::iterator i =
pages.begin(); i != pages.end(); ++i) {
            VirtualFree(*i, 0,
MEM_RELEASE);
        }
    }
private:
vector<void*> pages;
};

```

Berilgano'lchamgamos, pulbloklarnihosilqilishuchunBlockPool sinfiniyaratamiz.

BlockPoolsinfiningdasturfragmentlari:

```

template<class T, size_t PageSize = 65536, size_t
Alignment = 8 /* sizeof(void*) */>
class BlockPool : PagePool<PageSize>
{
public:
    BlockPool() : head(NULL) {
        BlockSize = align(sizeof(T),
Alignment);
        count = PageSize / BlockSize;
    }

    void* AllocBlock() {
        // todo: lock(this)
        if (!head) FormatNewPage();
        void* tmp = head;
        head = *(void**)head;
        return tmp;
    }
}

```

```

        void FreeBlock(void* tmp) {
            // todo: lock(this)
            *(void**)tmp = head;
            head = tmp;
        }
private:
    void* head;
    size_t BlockSize;
    size_t count;

    void FormatNewPage() {
        void* tmp = GetPage();
        head = tmp;
        for(size_t i = 0; i < count-1; i++)
        {
            void* next = (char*)tmp +
BlockSize;
            *(void**)tmp = next;
            tmp = next;
        }
        *(void**)tmp = NULL;
    }
};

```

Potoklarorasi dagisinxronzatsiyanitashkilqilishuchunberilganmanzil **todo: lock(this)** izohgaolibqo'yilgan. BuninguchunEnterCriticalSection yoki boost::mutex danfoydalanishhammumkin.

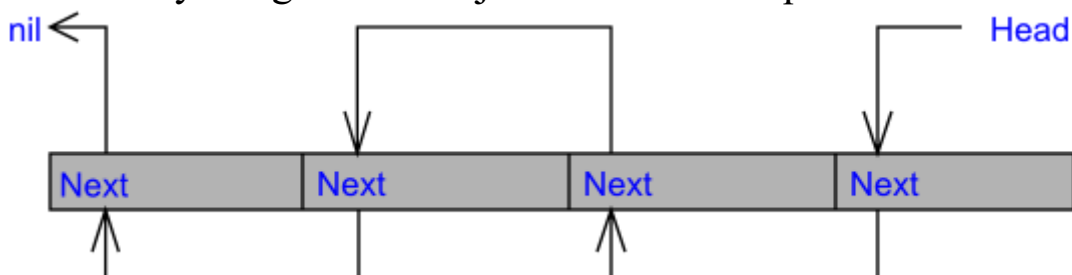
Agarbiornarsayozilganbo'lsa, bloklarmajmuiniformatlashdaabstrakt FreeBlock pulgablokqo'shishishlatilmaydi. O'ylabko'ring.

```

for (size_t i = 0; i < PageSize; i += BlockSize)
FreeBlock((char*)tmp+i);

```

FIFOtamoyilidagibloklarmajmuiteskari bo'libqoladi.



5.1-rasm. FIFOtamoyilidagibloklarmajmui.

Puldanketma-ketso‘ralganbirnechablokmanzillarkamayibketadi. Shuningdekprocessororqagaqaytishniyoqtirmaydi, chunkiuningPrefetchamaliibuziladi (UPD: zamonaviyprocessorlaruchuntegishliemas). Agartakrorlanishbilanalmashtirilsachi?

```
for (size_t i = PageSize-(BlockSize-(PageSize%BlockSize)); i != 0; i -= BlockSize) FreeBlock
```

Takrorlanishdaalmashtirishlarmanzillarbo‘yichaorqagabajariladi. Tayyorgalikko‘rilganidanso‘ng, sinfob‘ektniyozishmumkin:

```
template<class T>
class BlockAlloc{
public:
    static void* operator new(size_t s) {
        if (s != sizeof(T)) {
            return ::operator new(s);
        }
        return pool.AllocBlock();
    }
    static void operator delete(void* m, size_t s) {
        if (s != sizeof(T)) {
            ::operator delete(m);
        } else if (m != NULL) {
            pool.FreeBlock(m);
        }
    }
    static void* operator new(size_t, void* m) {
        return m;
    }
    static void operator delete(void*, void*) {
    }

private:
    static BlockPool<T> pool;
};
template<class T> BlockPool<T> BlockAlloc<T>::pool;
```

Sinfdaularishlagandanimauchunif (s != sizeof(T)) kerak. Tbazaviysinfdanmo‘rosxo‘rsinfyاراتilganvao‘chirilganbo‘lsa? degansavolbo‘lishimumkin.

Merosxo‘rlarodatdaoddiynew/delete operatorlaribilanishlatiladi, shuningdek BlockAlloc nihamjoylatirishmumkin. Buyo‘l bilanosonvaxavfsizdasturdabirornarsabuzibqo‘rqmasdan, pullarfoydalanishgaqaratilgansinflaraniqlashmumkin. Birnechtamerosxo‘rlarhambubilanyaxshiishlaydi.

BlockAlloc danmo‘rosxo‘rsifatida Node niolamizvayanasinovo‘tkazamiz.

Sinovnatijasigaqarasak, 5 barovartezva 120 msdabajarilmoqda. Ammo, c#allokatorhalihamyaxshi. Mumkinuyerdaoddiybo‘lmaganbiraloloqaliro‘yxatdanfoydalanilgandir. Agar, new operatoridanso‘ngzudlikbilan delete operatoriishlatilsa, xotirakamroqishlatiladivakeshdama’lumotlarhamkamayadivanatijani 62 msdaolishmumkin. Qiziqarlitomonishundaki, . NET CLR da GC nikutmasdanpulgamosbo‘shqolganlokalo‘zgaruvchinitezqaytaradi.

**Konteynervauniqo‘llanilishi.**Xotiradako‘phollardamerosxo‘rob’e ktlarinisaqlovchisinflarvujudgakeladi.

Ularningxotiradano‘chirilishibazaviysinfbilanbog‘liqdir. Masalan, sinfbo‘lsinvatugunichidagimatndanolinganmo‘rosxo‘rlariNodevaAttribu tehamda(char\*)qatorlardanto‘ldirilgan.

Yokipapkaniqaytao‘qishvayanao‘zgartirishda, birmartayuklanadifaylmenejerifayllarvakataloglarro‘yxati.

Yuqoridako‘rsatilgandek, new operatoriganisbattan delete operatorifoydalanishqiyinroqvamuammoliroqbo‘ladi.

Bazaviysinfob’ektibilanbog‘liqkattablokdamerosxo‘rsinfob’ektlariuchun xotiraajratishkerak.

Bazaviysinfob’ektiniyo‘qqilishpaytida, destruktorodatdagidek, merosxo‘rsinfuchunchaqiriladi, lekinxotiraniqaytarishingizshartemas — bukattablokdao zodqilinadi.

PointerBumpAllocator sinfyaratamiz. Businfkattablokdanturlio‘lchamdagiqismlarnikesibtashlashvaeskisitugag ach, yangikattabloknitanlashgaimkonberadi.

PointerBumpAllocatorsinfiningdasturfragmenti.

```
template<size_t PageSize = 65536, size_t Alignment
= 8 /* sizeof(void*) */>
class PointerBumpAllocator
{
public:
    PointerBumpAllocator() : free(0) { }

    void* AllocBlock(size_t block) {
```

```

        // todo: lock(this)
        block = align(block, Alignment);
        if (block > free) {
            free = align(block,
PageSize);
            head = GetPage(free);
        }
        void* tmp = head;
        head = (char*)head + block;
        free -= block;
        return tmp;
    }

    ~PointerBumpAllocator() {
        for (vector<void*>::iterator i =
pages.begin(); i != pages.end(); ++i) {
            VirtualFree(*i, 0,
MEM_RELEASE);
        }
    }

private:
    void* GetPage(size_t size) {
        void* page = VirtualAlloc(NULL,
size, MEM_COMMIT | MEM_RESERVE, PAGE_READWRITE);
        pages.push_back(page);
        return page;
    }

    vector<void*> pages;
    void* head;
    size_t free;
};
typedef PointerBumpAllocator<> DefaultAllocator;

```

Nihoyat, new va delete operatorlaribilan childObject nituzamizvaberilgan allocator bilanmurojaatqilamiz:

```
template<class T, class A = DefaultAllocator>
```



```

struct ChildObject
{
    static void* operator new(size_t s, A& allocator) {
        return allocator.AllocBlock(s);
    }
    static void* operator new(size_t s, A* allocator) {
        return allocator->AllocBlock(s);
    }

    static void operator delete(void*, size_t) { }
    static void operator delete(void*, A*) { }
    static void operator delete(void*, A&) { }
private:
    static void* operator new(size_t s);
};

```

Child sinfiga o'zgaruvchilarni qo'shish uchun barcha o'lonlarni new operator yordamida amalga oshirish kerak bo'ladi. New operator quyidagicha bo'ladi.

```
new (... parametrlar... ) ChildObject (...konstruktor parametrlari... )
```

Qulaylik uchun A& va A\* uchun new operatorlarni qo'llaymiz.

```
node = new(allocator) XmlNode(nodename);
```

Agar alokator ajratuvchisi sifatida qo'shilsa, ikkinchidan foydalanish qulayroq:

```
node = new(this) XmlNode(nodename);
```

Bundantushinarlibo'ladi, ortiqcha belgilardan qochishda amallarni bo'lish uchun ko'rsatki va havolal arkonvertatsiya bo'ladi.

Ob'ektlar yaratishda qaysi New operatoridan foydalanilganligiga qaramasdan, delete operatoriyordamida o'chirish amalga oshirilmaydi, kompilyatorning o'z standart delete operatoridan foydalanadi. Quyidagicha sintaktikasosida:

```
delete node;
```

Agar ChildObject sinfi ob'ekti yoki uning meros xo'ri ob'ekti dan foydalanayotgan bo'lsa, new istisno vaqtida operatoriga mos delete operatoricha qiriladi. Shuning uchun bu ob'ekt dan foydalanishdabirinchi size\_t parametrni void\* ga o'zgartirish lozim.

New operatorini private bo‘limigajoylashtirish, uniallokatorsizishlashagaruxsatbermaydi.

### Standartbo‘yichataqsimlovchilar.

1. Tuzilmalio‘zgaruvchisiuchundinamikxotiraaajratish.

Tuzilmalio‘zgaruvchisiuchunxotiraaajratishvabo‘shatish.

Quyidagitavsifgaegabo‘lgan data tuzilmasiberilganbo‘lsin:

```
struct Date
{
int day;
int month;
int year;
};
```

struct

Datetipidagio‘zgaruvchiuchunxotiradanjoyajratishvafoydalanishuchunqu yidagichadasturfragmentiniyozishkerak:

5.10-dastur. xotiradanjoyajratishvafoydalanish.

```
#include "stdafx.h"
#include <iostream>
using namespace std;

// tuzilma, kun, oy, yil;
struct Date
{
    int day;
    int month;
    int year;
};

int main(){
    // new operatori bilan xotiradan joy ajratish
    struct Date * pd; // struct Date ga ko‘rsatkich

    try{
        pd = new struct Date; // xotiraga joy
    }
    catch (bad_alloc ba) {
        cout << "Xorita ajratilmadi" << endl;
    }
}
```

```

        return -1;
    }

    // Agar xotira ajratilgan bo'lsa, unga
16.14.2020 qimat qilib beramiz.
    pd->day = 16;
    pd->month = 04;
    pd->year = 2020;

    cout << pd->day << ":" << pd->month << ":" <<
pd->year << endl;

    // pd o'zgaruvchisini xotiradan bo'shatish
    delete pd;
    system("pause");
    return 0;
}

```

2. Sinfob'ekti uchun dinamik xotira ajratish.

CDayWeek

sinfob'ekti ning ko'rsatkichiga xotira ajratish va foydalanishni keltiramiz

5.11-dastur. xotira ajratish va foydalanish.

```

// ConsoleApplication1.cpp : main project file.

#include "stdafx.h"
#include <iostream>
using namespace std;

// sinf, hafta kunlari
class CDayWeek{
    int d;
public:
    // konstruktor
    CDayWeek() { d = 1; }

    // funksiyalari
    int Get(void) { return d; }
    void Set(int nd) {

```

```

        if ((1<=nd)&&(nd<=7))
            d = nd;
    }
};

int main(){
    // new bilan xotiradan joy ajratish
    CDayWeek *p;

    try {
        // p ko'rsatkich uchun xotiradan joy
        ajratish
        p = new CDayWeek(); // CDayWeek sinf
        konstruktorini chaqirish
    }
    catch (bad_alloc ba) {
        cout << "Xotira ajratilmadi" << endl;
        cout << ba.what() << endl;
        return -1;
    }

    // qimat berish
    p->Set(3);
    cout << p->Get() << endl;
    // p ko'rsatkichdan xotirani bo'shatish
    delete p;
    system("pause");
    return 0;
}

```

3.

new

operatoriyordamidamassivlargaxotiradanjoyajratishvafoydalanish.new operatorimassivuchunxotiraaajratishuchunfoydalanishmumkin.

Massivgaxotiraaajratishda new operatorningumumiyshakli:

```
ptrArray = newtype [size];
```

Bunda, *ptrArray* - xotiraaajratilishikerakbo'lganmassivnomi, *type* - elementlarningtiplari, *size* - ularbazaviyvaixtiyoriybo'lishimumkin, *massivo'lchami* (elementlarsoni).

Delete operatori yordamida massivob'ektidan xotira nibo'shatish uchun quyidagi fragment yoziladi:

```
delete[] ptrArray;
```

4. Bazaviy tipliko'rsatkich massivigadina mikxotira ajratish va foydalanish. Quyida float tipidagi ko'rsatkich massivi uchun xotira ajratadi. So'ngra qator elementlari ixtiyoriy qiymatlar bilan to'ldiriladi. Shundayso'ng ajratilgan xotira delete [] operatoritomonidan bo'shatiladi.

```
float * ptrArray;  
ptrArray = new float[10];  
int d;  
for (int i = 0; i < 10; i++)  
    ptrArray[i] = i * 2 + 1;  
  
d = ptrArray[3];  
delete[] ptrArray;
```

5. Tizimli o'zgaruvchilarmassivi uchun xotira ajratish va undan foydalanish. 3

tatalabatuzilmalarmassivi uchun xotira ajratish va bo'shatish ko'rsatilgan. shuningdek, tuzilmalarmassivi ning element maydonlariga murojaat qilish ham ko'rsatilgan.

5.12-dastur. Xotira ajratish va foydalanish.

```
#include "stdafx.h"  
#include <iostream>  
using namespace std;  
  
struct TStudent  
{  
    string name;  
    string numBook;  
    int course;  
    float rank;  
};  
  
int main(){  
    int n_students = 3;  
    TStudent * pS;
```

```

try{

    pS = new struct TStudent[n_students];
}
catch (bad_alloc ba){
    cout << "Xotira ajratilmadi " << endl;
    cout << ba.what() << endl;
    return -1;
}

pS->name = "Aliyev Akbar";
pS->numBook = "965874123";
pS->rank = 3.93;
pS->course = 1;

(pS + 1)->name = "Bahromov Batir";
(pS + 1)->numBook = "20930032";
(pS + 1)->rank = 4.98;
pS[1].course = 3;

pS[2].name = "Qobilov Usmon";
pS[2].numBook = "12128983";
pS[2].rank = 4.32;
pS[2].course = 2;

cout << pS->name.c_str() << endl;
cout << pS[1].rank << endl;
cout << (pS + 2)->numBook.c_str() << endl;

delete[] pS;
system("pause");
return 0;
}

```

6. Massivob'ektlariuchunxotiraaajratishvaundanfoydalanish. New operatoridanfoydalangandamassivob'ektlariuchunxotiraaajratishko'rsa tiladiva delete operatoribilanxotiradanmassivob'ektlarinibo'shatishko'rsatiladi.

## 5.12-dastur. Xotiraajratishvafoydalanish.

```
#include "stdafx.h"
#include <iostream>
using namespace std;

class CMonth{
private:
    int month;

public:
    CMonth() { month = 1; }

    CMonth(int nmonth){
        if ((1 <= nmonth) && (nmonth <= 12))
            month = nmonth;
        else
            month = 1;
    }

    int Get(void) { return month; }
    void Set(int nmonth){
        if ((1 <= nmonth) && (nmonth <= 12))
            month = nmonth;
    }
};

int main(){
    int nMonths = 12;
    CMonth * pM;

    try {
        pM = new CMonth[12];
    }
    catch (bad_alloc ba)
    {
        cout << "Xotira ajratimadi: " << endl;
        cout << ba.what() << endl;
        return -1;
    }
}
```

```

    }

    for (int i = 1; i <= 12; i++)
        pM[i - 1].Set(i);

    cout << pM[3].Get() << endl;
    cout << pM[5].Get() << endl;

    delete[] pM;
    system("pause");
    return 0;
}

```

Yuqoridagidasturfragmentidamassivobektlarichki o'zgaruvchi 1 qiymatigainitsializatsiyaqilinadi, chunkibuqiymat Cmonth() parametrlarisizkonstruktordao'rnatiladi.

Bukonstruktormassivuchuninitsializatorvazifasinibajaradi. Biroq, sinfboshqakonstruktorniamalgaoshiradi, ya'nikonstruktorbilan 1 qiymatliparmetryokiparmetrlikonstruktorni. C++ sintaktikigako'ra, massivobektlariniparmetrlikonstruktorbilanishgatushiribbo'lmaydi.

Shuninguchun, cmonth sinfparametrlarisizkonstruktoramalgaoshirishkerak. Cmonth() parametrsizkonstruktorsinf dasturfragmentidanolibtashlansa, massivobekt laruchunxotiraaajratishmumkinbo'lmaydi.

Bittaob'ektlaruchunxotiraaajratishmumkin, massivuchunmumkinemas.

Agarma'lumbirsinfmassivob'ektlariuchunxotiraaajratishkerakbo'lsa, businfparametrlarisizkonstruktorgaegabo'lishivabajarilishikerak.

7. Massivo'lchaminidinamik kattalashtirish (kichiklashtirish) uchunxotiraaajratishvafoydalanish. Tuzilmalaruchunxotiraniqaytaajratish, tuzilmalarniishgatushirish. masalan, Dayweek tuzilishitipuchunxotiraaajratishjarayoniko'rsatiladi.

Xotiranidinamikravishdaajratishvaqaytaajratishbuusulningxotiranistatika jratishtanisbatanasosiyafzalligihisoblanadi.

Kerakbo'lgandavakerakbo'lgandadasturdaxotiraaajratishingizmumkin.

Dayweek tuzilishiparmetrlarisizkonstruktoryordamidaamalgaoshiriladi (yuqoridagidek).

5.13-dastur. Xotiraaajratishvafoydalanish.

```

#include "stdafx.h"
#include <iostream>
using namespace std;

```



```

struct DayWeek{
    int day;
    DayWeek() { day = 1; }
};

int main(){
    DayWeek * p;

    try {
        p = new DayWeek[5];
    }
    catch (bad_alloc ba){
        cout << "Xotira ajratilmadi" << endl;
        cout << ba.what() << endl;
        return -1;
    }
    for (int i = 0; i < 5; i++)
        p[i].day = i + 1;
    int d;
    d = (p + 1)->day;

    DayWeek * p2;

    try {
        p2 = new DayWeek[7];
    }
    catch (bad_alloc ba){
        cout << "Xotira ajratilmadi" << endl;
        cout << ba.what() << endl;
        return -1;
    }
    for (int i = 0; i < 5; i++)
        p2[i] = p[i];
    delete[] p;

    p = p2;
}

```

```

for (int i = 5; i < 7; i++)
    p[i].day = i + 1;

d = p[5].day;
d = (p + 6)->day;

delete[] p;

return 0;
}

```

Main() funkssiyasidaxotiraavval 5  
tuzidmadaniboratmassivuchunajratiladi. Keyinbuxotira 7  
tuzilmalimassivuchunqaytaajratiladi (reallocated).  
Buninguchunqo‘shimcha p2 ko‘rsatikichishlatiladi.  
Qaytataqsimlashdaavval p2 (7 element) uchunxotiraajratiladi.  
So‘ngrama’lumotlar p dan p2 gako‘chiriladi. Bu  
ppko‘rsatikichuchunajratilganedi, xotirabo‘shatiladi (5 taelementdan).  
Keyingiqadamdapqiymatp2 gao‘rnatiladi. Shundayqilib,  
harikkiko‘rsatikichbirxilxotiramaydoniko‘rsatadi.  
Massivbilanishlagandanso‘ng, p massivuchunxotiraajratiladi.  
Buyuqoridakurrsatibo‘tilgandasturlashdinamikxotiraniajratishvafoy  
dalanishningstandartshakllaribo‘libhisoblanadi.

## NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1. Standartalgoritm larkutubxonasinimauchunyaratilgan.
2. Algoritm kutubxonasi tomonidan taqdim etiladigan funksiyalar odatdanechtatoifagabo‘linadivaqaysilar?
3. Mutatorlarnimavaqandayvazifani amalga oshiradi.
4. Berilgan qiymat bo‘yicha elementlarni qidirish algoritmi qaysivanechtaparametr qabul qiladi?
5. Konteynerdama vjud elementlar orasidama’lumbir qiymat saqlaydigan element qidirish uchun qaysi funktsiya ishlatiladi?

6. Konteynerdan berilgan elementlarini sanash vazifasini bajaruvchi algoritmlar nisanabbering?
7. Funksiya taqqoslash uchun parametrlar sifatida ikki ta parametroladi. Bu qaysi funksiya.
8. Ikki ob'ektning qiymatlarini almashtirish algoritmini ayting.
9. Tartiblangan to'plamlar boshqatartiblangan to'plamni o'z ichiga olish yoki o'lmasligini tekshirish algoritmini ayting.
10. Iteratorlar nima uchun foydalaniladi.
11. **begin()** va **end()** funksiyalaridan nima maqsadda foydalaniladi.
12. Agar **begin()** va **end()** funksiyalar o'zaro zaroteng bo'lmasa, ularning orasida kamidancha nechta element bor.
13. Keyingi elementga murojlat qilish uchun iteratorni harakatlantirish amaliyati nima.
14. Agar konteynerda o'zgarishsiz qiymatli elementlar bo'lmasa, bu holda konteyner elementlariga murojlat qilish uchun nima foydalanish shkerak.
15. **cbegin()** va **cend()** funksiyalaridan nima maqsadda foydalaniladi.
16. Agar konteynerda o'zgaruvchi qiymatli elementlar bo'lmasa, bu holda konteyner elementlariga murojlat qilish o'zgarishsiz iteratorlardan foydalanish sh mumkinmi.
17. Iteratorlari qo'shimcha amallar qaysi konteynerlar uchun ishlatilmaydi.
18. Qaysi amal `iter1` va `iter2` lar orasida gapozitsiyalar sonini qaytaradi.
19. `(move_it1 != move_it2)` va `!(move_it1 == move_it2)` munosabatlar bir xilmi, bir xil bo'lmasa nima uchun.
20. Xotira nima qisimlovchilar va ularga qo'yilgan talablarni nima uchun kerak.
21. Memoriya o'zgaruvchilari uchun `new/delete` operatorlari bilan ishlatiladi, shuningdek `BlockAlloc` nima uchun joylatish kerak.
22. Agar, `new` operatoridan so'ngi zudlik bilan `delete` operatori ishlatilsa, xotira kamroq ishlatiladi va qanday gajima'lumotlar ham kamayadi.
23. Kaysi operator `new` operatoriyoki `delete` operatoridan foydalanish qiyinroq va muammoliroq, sababini tushintiribbering.
24. `New` operator qaysi bo'limiga joylashtirilsa, uning alokatori sizni shahar xususat bermaydi.
25. Tuzilmali o'zgaruvchi uchun xotira ajratish va bo'shatish uchun qanday amallar bajarish kerak.
26. Sinf ob'ekti uchun dinamik xotira ajratish uchun qanday amallar bajarish kerak.

27. new operatoriyordamidamassivlargaxotiradanjoyajratishvafoydalanishuchunqandayamallar bajarishkerak.
28. Bazaviytipliko'rsatkichmassivigadinamikxotiraaajratishvafoydalanishuchunqandayamallar bajarishkerak.
29. Ma'lumbirsinfmassivob'ektlariuchunxotiraaajratishkerakbo'lsa, businfqandayparametrikonstruktorgaegabo'lishikerak.
30. Massivo'lchaminidynamikkattalashtirish (kichiklashtirish) uchunxotiraaajratishvafoydalanishuchunqandayamallar bajarishkerak.




## AMALIY KO'NIKMA VA MALAKALARNI ANIQLASH HAMDA RIVOJLANTIRISH UCHUN ASSISMENT TOPSHIRIQLARI.

<b>BIRINCHI ASSISMENT TOPSHIRIG'I</b>	
	Standartalgoritmlargaoidberilganquyidagidasturb o'yichaberilgantopshiriqlarkeraklifragmentlarias osidabajaring. 👉 Bundadasturdagiba'zio'zgartirishlarnitopishorq alitopshiriqlarbosqichma — bosqichamalganoshiriladi.
dastur	topshiriqlar
<pre>// Created by MBBahodir #include "stdafx.h"  #include &lt;algorithm&gt; #include &lt;array&gt; #include &lt;iostream&gt; #include &lt;utility&gt; #include &lt;vector&gt; #include &lt;string&gt;  using namespace std; bool myfunction (int i, int j) { return i&lt;j; }</pre>	<p>1. Dasturdanechtaglobalfunksiyayara tilgan.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <p>2. Massivdaberilganmassivelementla riborliginitekshiruvchifragmentnio'zga rtiring.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>

<pre> int main() {     array&lt;int,10&gt; arr = {1, 3, 5, 7, 9, 11, 13, 15, 17, 19 };     int container[] = {5,10,15,20,25,30,35, 40,45,50};     int continent[] = {40,30,20,10};     string myStr[] = {"air","water","fire" ,"earth"};     vector&lt;string&gt; foo (myStr, myStr + sizeof(myStr)/sizeof( myStr[0]));     vector&lt;string&gt; bar (4);     int myints[] = {10,20,30,30,20,10,10 ,20};     vector&lt;int&gt; v(myints,myints+8);      cout &lt;&lt; "Qidirish va o'zgartirish uchun son kirit: ";     int search;     int replace;     cin &gt;&gt; search &gt;&gt; replace;      auto found = find(arr.begin(), arr.end(), search); </pre>	<p>3. Dasturning qaysi fragmentini izohlab qo'ysa, massivdaberilgan massiv elementlaribor deb chiqadi.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <p>4. Massivdaberilgan massiv elementlaribor degan fragment qachon ishlamaydi.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <p>5. Move amalini nechamarta bajarilgan.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <p>6. Birinchi qaysi kutubxonaga 'luqli.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <p>7. Ikkinchi qaysi kutubxonaga 'luqli.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
---	---

<pre>         if (found == arr.end()) {             cout &lt;&lt; search &lt;&lt; " topilmadi. " &lt;&lt; endl;         } else {             *found = replace;         }          for (int i : arr){             cout &lt;&lt; i &lt;&lt; ' ';             container[i] = i;         }         cout &lt;&lt; endl;          sort (container,container+ 10);         sort (continent,continent+ 4);          if (includes(container,c ontainer+10,continent ,continent+4, myfunction) )             cout &lt;&lt; "ichma ich massiv !\n";             else                 cout &lt;&lt; "ichma ich massiv emas !\n";             cout &lt;&lt; endl; </pre>	<p>8. Foovektorningelementlari kranga chiqarish fragmentini kiriting</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
---	--


<pre>         move ( foo.begin(), foo.begin()+4, bar.begin() );          foo = move (bar);          cout &lt;&lt; endl;         system("pause");         return 0; } </pre>	
<p>9. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____</p> <p>10. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.</p>	

<b>IKKINCHI ASSISMENT TOPSHIRIG‘I</b>	
	<p>Iteratorlarvaularningqo‘llanilishigaoidberilganqu yidagidasturbo‘yichaberilgantopshiriqlarkeraklifr agmentlariasosidabajaring.</p> <p>☞ Bundadasturdagiba‘zio‘zgartirishlarnitopishorq alitopshiriqlarbosqichma — bosqichamalganoshriladi.</p>
dastur	topshiriqlar
<pre> #include "stdafx.h" #include &lt;iostream&gt; #include &lt;vector&gt; using namespace std; int main() {     vector&lt;int&gt; myvector;     for (int i = 0; i &lt; 15; i++)     { </pre>	<p>1. Dasturda auto iter = myvector.begin(); uchun iterator yozing.</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p>

<pre>         myvector.push_back( rand() % 100);     }     int myints[] = {16,2,77,29,28};     const vector&lt;int&gt; myvector_one (myints, myints + sizeof(myints) / sizeof(myints[0]) );      auto iter = myvector.begin();     while(iter!=myvector. end())     {         cout &lt;&lt; *iter &lt;&lt; "   ";         ++iter;     } /* vector&lt;int&gt; v (myints, myints+ sizeof(myints)/sizeof(m yints[0]));     for (auto i = v.begin(); i &lt; v.end(); i++)     {         cout &lt;&lt; *i &lt;&lt; "   ";     } */      cout &lt;&lt; endl;     system("pause"); return 0; } </pre>	<p>2.Dasturdavector&lt;int&gt; myvector;uchuno‘zgarmasiteratorya rating.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
	<p>3. Dasturdagiortiqchakutubxonal arnio‘chiribtashlangvasabablariniayt ing.</p> <hr/> <hr/>
	<p>4. Ikkitavektornio‘zarosolishtirin g.</p> <hr/> <hr/> <hr/>
	<p>5. Vektorlarniteskarichiqarishdast urfragmentlariniyozing.</p> <hr/> <hr/> <hr/>
	<p>6. Vector saralangvavektordagi 2,5,7 elementlarnipozitsiyasinianiqlang.</p> <hr/> <hr/> <hr/>




	<p>_____</p> <p>7. Dasturdagi izohga olingan dastur fragmenti olib tashlansa, dasturda qandayo 'zgarishlar bo'ladiv natijasichi.</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p>
<p>8. Dasturda jamibo'lib, nechao'zgartirish kiritildi. _____</p> <p>9. Shudasturning analogini yaratishsiz gamusta qilvazifadir.</p>	

<b>UCHINCHI ASSISMENT TOPSHIRIG'I</b>	
	<p>Xotirani ajratish va foydalanishga oid berilgan quyidagi dastur bo'yicha berilgan topshiriqlar kerakli fragmentlari asosida bajaring.</p> <p>☝ Bundagi dasturda giba'zio'zgartirishlarni topish orqali topshiriqlar bosqichma bosqich amalga oshiriladi.</p>
dastur	topshiriqlar
<pre>// Created by MBBahodir #include &lt;iostream&gt; using namespace std;  struct TStudent {</pre>	<p>1. Dasturda nechta yangi tipdagi ob'ektlari yaratilgan.</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p>

<pre> string name; string numBook; int course; float rank; };  int main(){     int n_students = 3;     TStudent * pS;      try{ </pre>	<p>2. Dasturda gibarchatalabalar haqida gima'lumotniekrangachiqaring.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
<pre>         pS = new struct TStudent[n_students];     }     catch (bad_alloc ba){         cout &lt;&lt; "Xotira ajratilmadi " &lt;&lt; endl;         cout &lt;&lt; ba.what() &lt;&lt; endl;         return -1;     }      pS-&gt;name = "Aliyev Akbar";     pS-&gt;numBook = "965874123";     pS-&gt;rank = 3.93;     pS-&gt;course = 1; </pre>	<p>3. Massivo'lchaminidynamikkattalashtirishuchun fragmentyozing, masalan 5 talaba uchun.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
<pre>         (pS + 1)-&gt;name = "Bahromov Batir";         (pS + 1)-&gt;numBook = "20930032";         (pS + 1)-&gt;rank = 4.98;         pS[1].course = 3; </pre>	<p>4. Yangitipuchun get() va set() funksiyalarini yarating.</p> <hr/> <hr/> <hr/> <hr/>
	<p>5. Massivdan berilgan qiymatni izlash dastur fragmentini yozing.</p> <hr/> <hr/> <hr/> <hr/> <hr/>

<pre>         pS[2].name = "Qobilov Usmon";         pS[2].numBook = "12128983";         pS[2].rank = 4.32;         pS[2].course = 2;          cout &lt;&lt; pS- &gt;name.c_str() &lt;&lt; endl;         cout &lt;&lt; pS[1].rank &lt;&lt; endl;         cout &lt;&lt; (pS + 2)- &gt;numBook.c_str() &lt;&lt; endl;          delete[] pS;         system("pause");         return 0; } </pre>	<hr/> <hr/> <hr/> <hr/>
<p>6. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____</p> <p>7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.</p>	

## 2.2. Sonlisinflarbilanishlash

 Sonli sinfvaularningfunksiyalariasosida complex, valarray, slice, gsilce, indirect\_array, mask\_array sinflarivafunksiyalari, intervalsonlisinfvafunksiyalari, sonlisinfelementlariniyaratish, foydalanishgaasosidataqqoslashfunksiyalari, dasturlashdao‘zlarinitutishlari, uslublari, amallari, talablari, vazifalarivausullarikeltirilganbo‘lib, nazariybilimlarniasoslashuchundasturlartuzibko‘rsatilgan.

Bilimlarnimustahkamlashuchun 35  
tanazariysavolvaamaliyko‘nikmavamalakalarnirivojlantrishuchun 6  
taassimenttopshirig‘ivaharassimentda 7(6) tatopshiriq, jami  
49tatopshiriqberilgan.  
Butopshiriqlarnibajarishmavzunimustahkamlashuchunxizmatqiladi.

*Kalitso 'zlar.* kutubxona, sinf, shablon, matematikamallari, kompleksson, to'plam, to'plamosti, to'lamustidaamalar, saralash, intervalson.

*Bilishshartbo 'lgantushunchalar.* Tiptushunchasi, sinfvasinfob'ekti, merosxo'rsinf, to'plam, massiv, elementgamurojaat, funksiyavako'rsatkich, ma'lumotlarnikirishvachiqishi, dasturlashgaoiddastlabkitushunchalarvaC++tiliniqo'llabquvvatlovchimu hitdaishlashnibilishlozim.

*Bilibolasiz.* Sonlisinfvaularningfunksiyalari, komplekssonlarbilanishlashusullarivafunksiyalari, valarray soniob'ektibilanishlash, muo'rosxo'rsinfasosidato'plamlarniustidaamallarnibajarish, surish, shartlarasosidayangito'plamnihosilqilish, qismto'plamlarniyaratish, intervalsonvauningfunksiyalarnio'rganishingiz mumkin.

## REJA

1. Complex sonlisinfvafunksiyalari
2. Vallaray sonlisinfvafunksiyalari
3. Slice va gslice sonlisinflarivafunksiyalari
4. Qo'shimchasonlisinflar

## KIRISH

Hayotturmushtarzidafoydalaniladigansonlarto'plamlardanboshqasonlarto'plamivaularustidabajarilajiganamallarmavjud. Masalan, maktabdanbilamizki, komplekssonlar, oraliqsonlar, qolaversa, zamonaviyilmiysohalardaturlifazolardaturlichasonlarto'plamiishlatiladi. Intervalsonlari, noravshanto'plamlarshularkabi. Busonlarto'plamihayotdagiqaysidirtizimnimodellashtirishvaunijarayonlarigaoidmasalalarniyechishgaxizmatqiladi. Bumasalarniayniqsakompyuterdayechishkerak. Shuninguchunba'zibirko'pfoydalaniladigansonlarto'plamiuchunsinflaryaratilgan. Busonlardaularningamallari, funksiyalarixususiyatlarikeltirilgan. Bundaysonlargacomplex, vallaray, slice, gslice kabisinflarvamualliftomonidan 2013 yildayaratilgan interval sinfkutubxonalarinimisolqilibolishmumkin.

**Complex (kompleks) sonlisinfvafunksiyalari.** C++ standartikutubxonasi dagikomplekssonlarsinfiob'ektmodelidanfoydalanishning yaxshinamunasidir.

Arifmetikamallarqaytaaniqlashtufaylibusinfob'ektlario'rnatilganma'lum ottoplariidanbirigategishlikabiishlatiladi. Bundantashqari,

an'anaviyarifmetiko'zgaruvchilarvakomplekssonlarbirvaqtningo'zidaixti yoriyamallardaishtiroketishimumkin. (Eslatibo'tamiz, buyerdakomplekssonlarmatematikasigaoidumumiysavollarbilanshug'ull anmayapmiz.

Matematikabo'yichakoplekssonlarnazariyasinio'qishkerak).

Businfniishlatishuchuncomplexsarlavhafaylqo'shilishikerak.

Kichikdasturfragmentigaqarang:

```
#include <complex>
// ...
complex <double> a;
complex <double> b;
// ...
complex< double > c = a * b + a / b;
```

Kompleksvaarifmatikamallaro'zarohamkorlikdabajarailadi.

```
complex< double > a;
complex< double > complex_obj = a + 3.14159;
```

Komplekstiylarniarifmetiktiplarbilanhamyaratishvaularuchun operator= niishlatishmumkin.

```
complex< double > complex_obj;
double dval = 3.14159;
complex_obj = dval;
// ....
int ival = 3;
complex_obj = ival;
```

Ammoteskarisi, ya'nikomplekstipniarifmetiktipgatenglashtirishmumkinemas.

```
complex< double > complex_obj;
double dval = complex_obj;
```

Bundaarifmetiktipniakslantirishxatodeganxabarberadi.

Kompleksning 2 qismibo'ladihaqiqiyvamavhum. Haqiqiy (real) yokimavhum – qismini oddiyarifmetiktipgaqiymatqilibberishmumkin. Komplekssonlarsinfio'znavbatidarealvamavhumqismlarniqaytaruvchiikk itafunksiyagaega.

Bizsinfa'zolaridanfoydalanishuchunsintaktikiyordamidaularniolishimiz mumkin:

```
double re = complex_obj.real();
double im = complex_obj.imag();
```

Yokibusintaktikgaekvivaletsintaktik:

```
double re = real(complex_obj);
```

```
double im = imag(complex_obj);
```

Komplekssonsinfito'rtmatematikamallar - operatorlariqo'llab-quvvatlaydi. Bularga[+=], [-=], [\*=]va[/=]operatorlarikiradi.Masalan,

```
complex< double > complex_obj;  
complex< double > second_complex_obj;  
complex_obj += second_complex_obj;
```

Komplekssonlarnikiritish/chiqarishhamqo'llab-quvvatlanadi. Chiqishoperatorivergullarbilanajratilganrealvamavhumqismlarniqavslarichidachopetadi. Masalan, chiqishoperatorlarinibajarishnatijasi:

```
complex< double > complex0( 3.14159, -2.171  
);  
complex< double > complex1( complex0.real() );  
cout << complex0 << " " << complex1 << endl;
```

output

```
(3.14159,-2.171) (3.14159,0)
```

Kirishtishoperatorlariquyidagiformatlarbilanishlaydi:

Komplekssoningrealqisminikiritishoddiy kiritiladi, masalan, 525.25; mavhumqismiqavslarichidakiritiladi, masalan, (123.5), to'liqkiritishuchunqavsichidarealvamavhumqismlarivergulbilanajratiladi, Masalan, (12.5, 2.5). 6.1-dasturgaqarang.

6.1-dastur. Koplekssonlarnikiritishvachiqarish.

```
#include "stdafx.h"  
#include <iostream>  
#include <complex>  
  
using namespace std;  
  
int main(){  
    complex< double > a, b, c;  
    cout << "Complex sinfi oid turli t=farmatli 3  
ta son kiritibg:" << endl;  
    cin >> a >> b >> c;  
    cout << "Complex sinfiga oid sonlar:" << endl;  
    cout << a << b << c << endl;  
    system("pause");  
}
```

```
Complex sinfiga oid turli formatli 3 ta son
kiritibg:
3.1415
(-0.25)
(2.71, -9.8)
Complex sinfiga oid sonlar:
(3.1415,0)(-0.25,0)(2.71,-9.8)
```

Ushbufunksiyalardantashqari,  
kopmlekssonlarsinfiqidagifunksiyalarigaega:

```
sqrt(), abs(), polar(), sin(), cos(), tan(), exp(),
log(), log10() va pow().
```

Yozishvaqtidamavjud C++ standartkutubxonasio‘ngoperandli  
(masalan, qo‘shishi +=)  
operatorlarkomplekssonibo‘lmasamurakkabtayinlashamallariniqo‘llab-  
quvvatlamaydi. Masalan, buyozuvgaruxsatberilmaydi:

```
complex_obj += 1;
```

C++ standartiko‘ra, buoperatorlarbo‘lishikerak,  
ishlabchiqaruvchilarko‘pinchastandartlarniyozibtugatishgaetibkelolmayd  
ilar.

Bundayamallarniamalgaoshirishuchuno‘zoperatorimiznianiqlashim  
iz mumkin.

Buyerdaqo‘shishoperatoriniamalgaoshiruvchifunksiyaningvariantikeltiril  
gan.

```
inline complex<double>& operator+=(
complex<double>&cval, double dval ){
    return cval += complex<double>( dval );
}
```

Bumisol yordamida,  
turiboshqamurakkabtayinlashuchunuchoperatorlarifunksiyalariniyozish  
muammoemas.

Quyidagidasturuchunmurakkabtayinlashuchunuchoperatorlarifunksiyala  
riniqo‘shingvatekshirishuchununiishgatushirish.

```
int main() {
    complex< double > cval ( 4.0, 1.0 );
    cout << cval << endl;
    cval += 1;
```

```

    cout << cval << endl;
    cval -= 1;
    cout << cval << endl;
    cval *= 2;
    cout << cval << endl;
    cout /= 2.;
    cout << cval << endl;
system("pause");
}

```

C++

standartidakomplekssonuchuninkrementvadekrementamallariningbajarili shiko‘rsatilmagan. Biroq, ularningsemantikasijudaaniq, agar `cval += 1` niyozaolsak, bu `cval`ninghaqiqiyqismini 1 bilanoshirishnianglatadi, undainkrementamalijudaosonko‘rinadi. Murakkab<double>tipiuchunbuamallarniamalgaoshirishvaquyidagidasturniishgatus hirish:

```

int main() {
    complex< double > cval( 4.0, 1.0 );
    cout << cval << endl;
    ++cval;
    cout << cval << endl;
}

```

Buyuqoridagilardantashqarikomplekssonlaribilashishlashsinfiningko‘plab, matematikfunksiyalarmavjud. AmmoC++standartigaqarabayrimamallardan, funksiyalardanfoydalanibbo‘lmaydi. Shuninguchunfoydalanuvchiningo‘zmustaqilkutubxonayaratibishlashim aqul.

### Valarray

**sonsinfivauningfunksiyalari.**YaqinoradaC++damassivelementlaribilani shlashvasamaralisaqlashuchun Valarray konteyneripaydobo‘ldi. BuhammaC++standartlaridabo‘lmasligihammumkin.

-Businfmassivelementindekslaribo‘yichabirlashtrishoperatorlarivaelementlaromatematikamallarnibajaradi.

-Vektorbilansolishtirganda Valarray sinfmatematikamallarnibajarishdasamaraliroqhisoblanadi.

Valarray sinfningochiqfunksiyalari:



## **apply()**

bu massiv elementlaribo'yichabirvaqtdao'zgarishlarnibajaradivayangimas sivrqaytaradi.

**sum()** – massivdagielementlaryig'indisiniqaytaradi.

6.2-dastur. apply () va sum () funksiyalaridanfoydalanish.

```
#include "stdafx.h"
#include<iostream>
#include<valarray>
using namespace std;
int main(){
    int myints[] = { 5, 25, 55, 85, 115 };
    valarray<int> varr (myints,
sizeof(myints)/sizeof(myints[0]));
    valarray<int> varr_one ;

    varr_one = varr.apply([](int x){return x=x+5;});

    cout << "Massivning yangi qiymatllar: ";
    for (int &x: varr_one) cout << '\t' << x;
    cout << endl;

    cout << "Oldingi massiv sum() => ";
    cout << varr.sum() << endl;
    cout << "Yangi massiv sum() => ";
    cout << varr_one.sum() << endl;
    system("pause");
    return 0;
}
```

### 6.2-dastur.output

Massivning yangi qiymatllar: 10 30 60 90 120

Oldingi massiv sum() => 285

Yangi massiv sum() => 310

## **Min()**

**va**

**max()**fnuksiyalarimosravishdamassivengkichikvaengkattaelementlariqi matlariniqaytaradi.

```
cout << "min() => ";
    cout << varr.min() << endl;
```

```
cout << "max() => ";  
cout << varr.max() << endl;
```

### **shift()**

berilgan son qiymat bo'yicha indeksasosidamassivni surib yangi massiv qaytaradi. Agar son musbat bo'lsa chapga suriladi, manfiy bo'lsa, o'ngga suriladi.

### **cshift()**

berilgan son qiymat bo'yicha indeksasosidamassivni aylanashaklida surib yangi massiv qaytaradi. Agar son musbat bo'lsa chapga suriladi, manfiy bo'lsa, o'ngga suriladi.

6.3-dastur. shift() va cshift() dan foydalanish.

```
#include "stdafx.h"  
#include<iostream>  
#include<valarray>  
using namespace std;  
int main(){  
    int myints[] = { 5, 25, 55, 85, 115 };  
    valarray<int> varr (myints,  
sizeof(myints)/sizeof(myints[0]));  
    valarray<int> varr_one, varr_two;  
    cout << "Massivning eski qiymatlar: ";  
    for (int &x: varr_one) cout << '\t' << x;  
    cout << endl;  
  
    varr_one = varr.shift(-2);  
    cout << "Massivning shift(-2) qiymatlar: ";  
    for (int &x: varr_one) cout << '\t' << x;  
    cout << endl;  
  
    varr_one = varr.shift(2);  
    cout << "Massivning shift(2) qiymatlar: ";  
    for (int &x: varr_one) cout << '\t' << x;  
    cout << endl;  
  
    varr_one = varr.cshift(2);  
    cout << "Massivning cshift(2) qiymatlar: ";  
    for (int &x: varr_one) cout << '\t' << x;  
    cout << endl;
```

```

varr_one = varr.cshift(-2);
cout << "Massivning cshift(-2) qiymatllar: ";
for (int &x: varr_one) cout << '\t' << x;
cout << endl;

system("pause");
return 0;
}

```

### 6.3-dastur.output

Massivning eski qiymatllar:

Massivning shift(-2) qiymatllar:	0	0	5	25	55
Massivning shift(2) qiymatllar:	55	85	115	0	0
Massivning cshift(2) qiymatllar:	55	85	115	5	25
Massivning cshift(-2) qiymatllar:	85	115	5	25	55

**swap ()** - birmassivni ikkinchi sibir massivga almashtiradi.

```
varr.swap(varr1);
```

Yuqoridagilardan tashqari quyidagi funksiyalar mavjud:

operator=	Massiv qiymatini o'zlashtirish
operator[]	Massiv indeks orqali elementiga murojaat
resize	Massiv elementlar soni (hajmi) niqayta aniqlash
size	Massiv elementlar soni (hajmi)

Valarrayning standart amallari

```

valarray operator+() const;
valarray operator-() const;
valarray operator~() const;
valarray<bool> operator!() const;

```

Valarray tiplari ustida amallar:

```

valarray& operator*=(const valarray& rhs);
valarray& operator/=(const valarray& rhs);
valarray& operator%=(const valarray& rhs);
valarray& operator+=(const valarray& rhs);
valarray& operator-=(const valarray& rhs);
valarray& operator^=(const valarray& rhs);
valarray& operator&=(const valarray& rhs);
valarray& operator|=(const valarray& rhs);
valarray& operator<<=(const valarray& rhs);
valarray& operator>>=(const valarray& rhs);

```

Valarraytipivaodatiyoddiytiplarustidaamallar:

```
valarray& operator*= (const T& val);  
valarray& operator/= (const T& val);  
valarray& operator%= (const T& val);  
valarray& operator+= (const T& val);  
valarray& operator-= (const T& val);  
valarray& operator^= (const T& val);  
valarray& operator&= (const T& val);  
valarray& operator|= (const T& val);  
valarray& operator<<= (const T& val);  
valarray& operator>>= (const T& val);
```

Yuqoridakeltirilganoperatorilar Valarray sinfiningfunksiyalarihisoblandi.

Valarray tiplaribilanbajariladiganamallarniqaytaaniqlashmumkin. Bundabirinchiamal Valarray tipi = Valarray tipi \* Valarray tipi, ikkinchiamal Valarray tipi = oddiytip \* Valarray tipi, uchinchitip Valarray tipi = Valarray tipi\* oddiytipketmaketligiasosidaaniqlangan.

[\*] - amaliniqaytaaniqlash:

```
template <class T> valarray<T> operator* (const  
valarray<T>& lhs, const valarray<T>& rhs);  
template <class T> valarray<T> operator* (const T&  
val, const valarray<T>& rhs);  
template <class T> valarray<T> operator* (const  
valarray<T>& lhs, const T& val);
```

[\*] - amaliniqaytaaniqlash:

```
template <class T> valarray<T> operator* (const  
valarray<T>& lhs, const valarray<T>& rhs);  
template <class T> valarray<T> operator* (const T&  
val, const valarray<T>& rhs);  
template <class T> valarray<T> operator* (const  
valarray<T>& lhs, const T& val);
```

[/] - amaliniqaytaaniqlash:

```
template <class T> valarray<T> operator/ (const  
valarray<T>& lhs, const valarray<T>& rhs);  
template <class T> valarray<T> operator/ (const T&  
val, const valarray<T>& rhs);  
template <class T> valarray<T> operator/ (const  
valarray<T>& lhs, const T& val);
```

[%] - amaliniqaytaaniqlash:

```
template <class T> valarray<T> operator% (const  
valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator% (const T&  
val, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator% (const  
valarray<T>& lhs, const T& val);
```

[+] - amaliniqaytaaniqlash:

```
template <class T> valarray<T> operator+ (const  
valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator+ (const T&  
val, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator+ (const  
valarray<T>& lhs, const T& val);
```

[-] - amaliniqaytaaniqlash:

```
template <class T> valarray<T> operator- (const  
valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator- (const T&  
val, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator- (const  
valarray<T>& lhs, const T& val);
```

[^] - amaliniqaytaaniqlash:

```
template <class T> valarray<T> operator^ (const  
valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator^ (const T&  
val, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator^ (const  
valarray<T>& lhs, const T& val);
```

[&] – mantiqiyamaliniqaytaaniqlash:

```
template <class T> valarray<T> operator& (const  
valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator& (const T&  
val, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator& (const  
valarray<T>& lhs, const T& val);
```

[|] – mantiqiyamaliniqaytaaniqlash:

```
template <class T> valarray<T> operator| (const  
valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator| (const T& val, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator| (const valarray<T>& lhs, const T& val);
```

[<<] – chapgasurishamaliniqaytaaniqlash:

```
template <class T> valarray<T> operator<< (const valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator<< (const T& val, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator<< (const valarray<T>& lhs, const T& val);
```

[>>] – o'nggasurishamaliniqaytaaniqlash:

```
template <class T> valarray<T> operator>> (const valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator>> (const T& val, const valarray<T>& rhs);
```

```
template <class T> valarray<T> operator>> (const valarray<T>& lhs, const T& val);
```

[&&] – tezkormantiqiyamaliniqaytaaniqlash:

```
template <class T> valarray<bool> operator&& (const valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<bool> operator&& (const T& val, const valarray<T>& rhs);
```

```
template <class T> valarray<bool> operator&& (const valarray<T>& lhs, const T& val);
```

[||] – tezkormantiqiyamaliniqaytaaniqlash:

```
template <class T> valarray<bool> operator|| (const valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<bool> operator|| (const T& val, const valarray<T>& rhs);
```

```
template <class T> valarray<bool> operator|| (const valarray<T>& lhs, const T& val);
```

[==] – mulohazaamaliniqaytaaniqlash:

```
template <class T> valarray<bool> operator== (const valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<bool> operator== (const T& val, const valarray<T>& rhs);
```

```
template <class T> valarray<bool> operator== (const
```

```
valarray<T>& lhs, const T& val);
```

[!]=] – mulohazaamaliniqaytaaniqlash:

```
template <class T> valarray<bool> operator!= (const  
valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<bool> operator!= (const  
T& val, const valarray<T>& rhs);
```

```
template <class T> valarray<bool> operator!= (const  
valarray<T>& lhs, const T& val);
```

[<] – mulohazaamaliniqaytaaniqlash:

```
template <class T> valarray<bool> operator< (const  
valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<bool> operator< (const  
T& val, const valarray<T>& rhs);
```

```
template <class T> valarray<bool> operator< (const  
valarray<T>& lhs, const T& val);
```

[>] – mulohazaamaliniqaytaaniqlash:

```
template <class T> valarray<bool> operator> (const  
valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<bool> operator> (const  
T& val, const valarray<T>& rhs);
```

```
template <class T> valarray<bool> operator> (const  
valarray<T>& lhs, const T& val);
```

[<=] – mulohazaamaliniqaytaaniqlash:

```
template <class T> valarray<bool> operator<= (const  
valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<bool> operator<= (const  
T& val, const valarray<T>& rhs);
```

```
template <class T> valarray<bool> operator<= (const  
valarray<T>& lhs, const T& val);
```

[>=] – mulohazaamaliniqaytaaniqlash:

```
template <class T> valarray<bool> operator>= (const  
valarray<T>& lhs, const valarray<T>& rhs);
```

```
template <class T> valarray<bool> operator>= (const  
T& val, const valarray<T>& rhs);
```

```
template <class T> valarray<bool> operator>= (const  
valarray<T>& lhs, const T& val);
```

**Slicesinfi.**Bu - xizmatchisinfbolib, valarray sinfningsmerosxo'ridirvabiro'lchovlito'plamostilarniyaratishdaishlatiladi. Agar valarray massivniikkio'lchovlimatritsasifatidaqarasak, undanbiro'lchovlivektornihosilqilishmumkin.

Businf slice\_array tipdagiob'ektnitavsiflovchiparametrlarnisaqlaydi. Agarto'plamostibilvositayaratilgan valarray sinftipidabo'lsa, sinfob'ekti valarray ob'ektiuchunargumentsifatidako'rsatiladi. To'plamostidasaqlanadiganqiymatlar valarray tipinioladivaquyidagixususiyatlarnio'zichigaoladi:

valarraydagiboshlang'ichindeks.  
massivningumumiyuzunligivaundagielementlarsoni;  
valarray elementlarningketma-  
ketindekslariorasidagiqadamyokimasofanianiqlash.

Agarqismto'plamyordamidaaniqlanganmassivdoimiy valarrayningkichikbo'lagibo'lsa, bumassivyangi valarraydir. Doimiy valarray tipidagito'plambirqismto'plamyordamidaaniqlanganbo'lsa, dastlabki valarray uchunmohavolalarsemantikasidir. O'zgarmasbo'lgan valarray laruchunbaholashmexanizmivaqtvaxotirasarflaydi.

Qismto'plamasosidaaniqlanganbo'lsavafarqlansa, barchaelementlarigamurojaatbo'lsa, valarray tipidagimassivlaruchunamallar bajarilishikafolatlanadi.

Slice sinfiuchun Slice konstrukturiishlatiladi. Bir-biridanbirxilmasofadabo'lganvabelgilanganelementbilanboshlanadigan, birnechaelementlardaniborat valarray birto'plamostisiniyaratadi.

Slice sinfiningfunksiyalariquyidagilar:  
**size()** – valarray to'plamostiningelementlarisoninianiqlaydi.  
**start()** – valarray to'plamostiningboshlang'ichindeksiniqaytaradi  
**stride()** - valarray to'plamostiningelementlariorasidagimasofanitopadi.

6.3-dastur. Slicesinfidanfoydalanish.

```
#include "stdafx.h"

#include <valarray>
#include <iostream>
using namespace std;
int main( ){
    int i;
    size_t sizeVA, sizeVAR;
```



```

size_t startVAR, strideVAR;

valarray<int> va(28), vaResult;

for ( i = 0 ; i < 28 ; i += 1 )
    va [ i ] = (rand()%100)+1;

cout << "valarray elementlari: \n ( ";
    for ( i = 0 ; i < 28 ; i++ )
        cout << va[i] << '\t';
cout << ")." << endl;

sizeVA = va.size();
cout << "valarray elementlar soni: "
<< sizeVA << "." << endl << endl;
// slice konstruktori
slice vaSlice( 0, 6, 4);
vaResult = va[vaSlice];

cout << "valarray to'plam ostisi vaResult = "
<< "va[slice( 0, 6, 4)] =\n ( ";
    for ( i = 0 ; i < vaSlice.size() ; i++ )
        cout << vaResult [ i ] << " ";
cout << ")." << endl;

sizeVAR = vaSlice.size( );
cout << "to'plam ostining elementlar soni: "
<< sizeVAR << "." << endl;

startVAR = vaSlice.start( );
cout << "to'plam ostining birinchi element
indeksi: "
<< startVAR << "." << endl;

strideVAR = vaSlice.stride( );
cout << "to'plam ostining elementlari orasidagi
masofa: "

```

```

<< strideVAR << "." << endl;
    system("pause");
}

```

### 6.3-dastur. output

```

valarray elementlari:
( 42  68   35    1   70   25   79
59   63   65    6   46   82   28
62
92   96   43   28   37   92    5
3   54   93   83   22   17   ).
valarray elementlar soni: 28.

valarray to'plam ostisi vaResult = va[slice( 0, 6,
4)] =
( 42 70 63 82 96 92 ).
to'plam ostining elementlar soni: 6.
to'plam ostining birinchi element indeksi: 0.
to'plam ostining elementlari orasidagi masofa: 4.

```

Dasturgasinf gaberilganta'rifasosidauningkonstruktori 3  
taparametrqabulqilarekan.

Birinchiparamertbelgilanganelementlarindeksi, ikkinchisielementlarsoni,  
elementlarindeksiorasidagimasofa.

**Gslice** **sinfi.** Xizmatchisinifbo'lib, valarray  
ningko'po'lchamlito'plamostilaribilanishlashgamo'ljallangan. Agar  
valarray massivdako'po'lchamlito'plamsifatidaqaralsa, Gslice  
sinfiundanko'po'lchamlivektornioliadi.

Sinf gslice\_arraytipbilantavsiflanuvchiparamertlarnisaqlaydi,agarv  
alarray\*\*<Type>tipidagiob'ektuchunargumentsifatida class gslice  
ob'ektiberilganbo'lsa, Valarray

to'plamostisibilvositaqurilganhisoblanadi. \*\*  
to'plamostielementlaribo'lib, Valarray bazaviysinfdanolinganva 3  
taparamertniqabulqiladi:

- Boshlang'ichindeks;
  - Vektoruzunligi valarray<size\_t>sinfi;
  - Vektorqadami (elementlariorasidagimosofa) valarray<size\_t>sinfi;
- Vektorlarninguzinligibirxilbo'lishikerak.

Agart o'plam gslice bilan aniqlangan va valarray ning to'plam ostida bo'lsa, gslice yangi valarray hisoblanadi. O'z garmas bo'lgan Valarray uchun baholash mexanizmi vaqt va xotira ni tejaydi.

boshlang'ich va oxirgi to'plamlar gslice bilan aniqlangan bo'lsa, farqli, barcha indekslariga murojaat bo'lsa, Valarray to'plam ustida barcha amallarka foydalaniladi.

gslice sinfi uchun gslice konstruktor i ishlatiladi. Belgilangan element bilan boshlanadigan, bir nechta elementlardan iborat valarray to'plam ostida yaratiladi.

gslice sinfining funksiyalari quyidagilar:

**size()** – valarray

to'plam ostidagi umumiy massiv elementlar sonini aniqlaydi.

**start()** – valarray

to'plam ostidagi umumiy boshlang'ich indeksini qaytaradi

**stride()** - valarray

to'plam ostidagi umumiy elementlar orasidagi masofani topadi.

Gslice konstruktori:

```
gslice(  
    size_t _StartIndex,  
    const valarray<size_t>& _LenArray,  
    const valarray<size_t>& _IncArray);
```

Parametrlari:

**\_StartIndex** – to'plam ostidagi birinchi elementi indeksini

**\_LenArray** – har bir to'plam ostidagi massiv elementlar sonini

**\_IncArray** -

har bir to'plam ostidagi massiv elementlar orasidagi masofa (qadam);

Standart konstruktor boshlang'ich indeksini uzunlikdagi vektorlar va vektorlari uchun uzunlik va qadam nol qiymatini saqlaydi.

Ikkinchi konstruktor boshlang'ich indeksini uchun **\_startindex** saqlaydi, qator uzunligi uchun **\_lenarray** va qadam uchun **\_incarray** qiymatlarini saqlaydi.

gslice – bir nechta qismdantash kiltopgan valarray uchun to'plam ostidagi valarrayni aniqlaydigan va hammasi belgilangan birelementin deksidan boshlanadi. gslice i slice ning farqlari birinchi bir nechta qism valarray lar bilan ishlatiladi.

6.4-dastur. gslicesinidan foydalanish.

```
#include "stdafx.h"  
#include <valarray>
```

```

#include <iostream>
using namespace std;
int main(){

    int i;

    valarray<int> va ( 20 ), vaResult;
    for ( i = 0 ; i < 20 ; i+=1 )
        va [ i ] = i+1;

    cout << "valarray massiv elementlari: " << endl
<< "(";
    for ( i = 0 ; i < 20 ; i++ )
        cout << '\t' << va [ i ];
    cout << " )" << endl;

    valarray<size_t> Len (2), Stride(2);
    Len [0] = 6;
    Len [1] = 4;
    Stride [0] = 2;
    Stride [1] = 4;

    gslice vaGSlice (0, Len, Stride );
    vaResult = va[vaGSlice];

    cout << " vaGSlice asosida vaResult
elementlari:" << endl
<< "va[vaGSlice] = (";

    for ( i = 0 ; i < 10 ; i++ )
        cout << '\t' << vaResult [ i ];
    cout << ")" << endl;

    const valarray <size_t> sizeGS = vaGSlice.size (
);

    cout << "vaResult elementlar soni:"
<< " vaGSlice.size ( ) = ( ";

```

```

        for ( i = 0 ; i < 2 ; i++ )
            cout << sizeGS[ i ] << '\t';
    cout << ")." << endl;

    size_t vaGSstart = vaGSlice.start ( );

    cout << "vaResult uchun birinchi element
indeksi: "
<< vaGSstart << "." << endl;

    const valarray <size_t> strideGS =
vaGSlice.stride ( );

    cout << "vaResult ychun qadamlar:"
<< " vaGSlice.stride ( ) = ( ";
        for ( i = 0 ; i < 2 ; i++ )
            cout << strideGS[ i ] << '\t';
    cout << ")." << endl;
    system("pause");
}

```

#### 6.4-dastur. Output

```

valarray massiv elementlari:
(      1      2      3      4      5      6
7      8      9     10     11     12     13
14
15     16     17     18     19     20 )
  vaGSlice asosida vaResult elementlari:
va[vaGSlice] = (      1      5      9     13
3      7     11     15     5     9)
vaResult elementlar soni: vaGSlice.size ( ) = ( 6
4      ).
vaResult uchun birinchi element indeksi: 0.
vaResult ychun qadamlar: vaGSlice.stride ( ) = ( 2
4      ).

```

Dasturtahlilibilanmutaqilshug‘ullaning.

**Slice\_array sinfi.** Ichkiyordamchishablonsinfi hisoblanadi. Valarray qismlar bilan aniqlangano 'plamostimassivlar uchun amallarni qo'llab quvvatlaydi. Sinf sintaktiki quyidagicha:

```
template <class Type>
class slice_array : public slice {
public:
    typedef Type value_type;
    void operator=(const valarray<Type>& x) const;
    void operator=(const Type& x) const;
    void operator*=(const valarray<Type>& x) const;
    void operator/=(const valarray<Type>& x) const;
    void operator%=(const valarray<Type>& x) const;
    void operator+=(const valarray<Type>& x) const;
    void operator-=(const valarray<Type>& x) const;
    void operator^=(const valarray<Type>& x) const;
    void operator&=(const valarray<Type>& x) const;
    void operator|=(const valarray<Type>& x) const;
    void operator<<=(const valarray<Type>& x)
const;
    void operator>>=(const valarray<Type>& x)
const;
}
```

Businfob'ekti

valarray<Type> ob'ektidagi elementlar ketma-ketligi bilan yoziladigan slicesinfob'ekt bilan birgalikda valarray <Type> sinfiga havolasaqlaydigan ob'ektlarni tasniflaydi.

Shablon sinfi muayyan valarray amallarini monidan bilvosita yaratilgan vada surdabe vosita foydalanish mumkin emas. Ichkiyordamchishablonsinfi quyidagi sintaktik bilan ishlatiladi:

```
slice_array< Type> valarray< Type:: operator[] ( slice);
```

Slice\_array <Type> ob'ektifaqat [SL] formatliva SL slice valarray ifodasi ni yozish orqali yaratiladi. slice\_array sinfining funksiyalari valarray< Type> uchun belgilangan mos funksiyalar bilan bir xil ishlatiladi, faqat tanlangan elementlar ketma-ketligiga 'sirko'rsatadi. Slice\_array sinftomonidan nazorat natijasida konstruktor gauch parametrlar belgilanadi: massivning birinchi element indeksini, elementlar soni va elementlar orasidagi masofa.

**gslice\_array sinfi.** Ichkiyordamchishablonsinfi hisoblanadi. Valarray qismlar bilan aniqlangano 'plamostimassivlar uchun amallarni qo'llab quvvatlaydi. Sinf sintaktiki quyidagicha:

```
template <class Type>
class gslice_array : public gslice {
public:
    typedef Type value_type;
    void operator=(const valarray<Type>& x) const;
    void operator=(const Type& x) const;
    void operator*=(const valarray<Type>& x) const;
    void operator/=(const valarray<Type>& x) const;
    void operator%=(const valarray<Type>& x) const;
    void operator+=(const valarray<Type>& x) const;
    void operator-=(const valarray<Type>& x) const;
    void operator^=(const valarray<Type>& x) const;
    void operator&=(const valarray<Type>& x) const;
    void operator|=(const valarray<Type>& x) const;
    void operator<<=(const valarray<Type>& x)
const;
    void operator>>=(const valarray<Type>& x)
const;
}
```

Businfob'ekti - valarray<Type> ob'ektidagi massiv elementlarni ketma-ketligi bilan yoziladigan gslice sinfi bilan birgalikda valarray <Type> sinfiga havolasaqlaydigan ob'ektlarni tasniflaydi.

Shablon sinfi muayyan valarray amallari tomonidan bilvosita yaratilgan va dasturda bevosita foydalanish mumkin emas. Ichkiyordamchishablonsinfi quyidagi sintaksis bilan ishlatiladi:

```
gslice_array<Type> valarray<Type>::operator[] ( const gslice& )
```

gslice\_array <Type> ob'ektifaqat [GL] formatliva GL gslice valarray ifodasi ni yozish orqali yaratiladi. slice\_array sinfining funksiyalari valarray<Type> uchun belgilangan mos funksiyalar bilan bir xil ishlatiladi, faqatlangan elementlarni ketma-ketligi ta'sir ko'rsatadi. Slice\_array sinfi tomonidan nazorat natijasida konstruktor ga uch parametrlar belgilanadi: massivning birinchi element indeksini, elementlar soni va elementlar orasidagi masofa. Dastur fragmentiga qarang.

```
const size_t lv[] = {2, 3};
const size_t dv[] = {7, 2};
const valarray<size_t> len(lv, 2), str(dv, 2);
```

**indirect\_array** **sinfi.** Ichki yordamchishablonsinfi hisoblanadi. Bazaviy `valarray` sinfi elementlari asosida `to'plam` o'lib, belgilangan `to'plam` asosida amallarni ta'minlash orqali `valarray` ob'ektlarining `to'plam` ob'ektlar bo'lgan ob'ektlarni o'qib o'lish quvvatlaydigan yordamchisi `inf`.

`Businf` ob'ekti - `valarray<Type>` ob'ekti dagi elementlarni ketma ketligi bilan yoziladigan `valarray<size_t>` sinfi ob'ekti `xa` bilan birgalikda `valarray <Type>` sinfi gahavola saqlaydigan ob'ektlarni tasniflaydi.

`Indirect_array<type>` ob'ekti faqat `va[xa]` shaklining ifodasi ni yozish orqali yaratiladi. Tanlangan elementlarni ketma ketligi asosida `Indirect_array` sinfi funksiyalari `valarray` uchun belgilangan tegishli funksiyalar bir xil ishlaydi. Ketma ketlik `XA` dan iborat bo'ladi. VA ichidagi `XA[I]` indeksiga akslatirilganda `I` elementlarni o'qib o'lish bo'ladi.

6.5-dastur. `indirect_array` sinfi dan foydalanish.

```
#include "stdafx.h"
#include <valarray>
#include <iostream>

int main( )
{
    using namespace std;
    int i;

    valarray<int> va ( 10 );
    for ( i = 0 ; i < 10 ; i += 2 )
        va [ i ] = i;
    for ( i = 1 ; i < 10 ; i += 2 )
        va [ i ] = -1;

    cout << "Valarray maasiv elementlari: ( ";
        for ( i = 0 ; i < 10 ; i++ )
            cout << va [ i ] << '\t';
    cout << ")." << endl;

    valarray<size_t> indx(5);
    int j = 0;
    for (int i = 1; i < 10; i+=2)
```



```

    {
        indx [j] = i;
        //cout << indx[j] << '\t';
        j++;
    }
    cout << endl;
    va[indx] = 10;

    cout << "O'zgartirilgan valarray elementlari: (
";
        for (i = 0 ; i < 10 ; i++ )
            cout << va [ i ] << '\t';
    cout << ")." << endl;
    system("pause");
}

```

6.5-dastur.output

```

Valarray maasiv elementlari: ( 0      -1      2
-1      4      -1      6      -1      8      -1
).

O'zgartirilgan valarray elementlari: ( 0      10
2      10      4      10      6      10      8
10).

```

**mask\_array sinfi.** Ichki yordamchi shablonsinfi hisoblanadi. Bazaviy valarray sinfi elementlarini o'sha elementlarni o'z ichiga olib, belgilangan o'plam osti uchun mantiqiy amallarni ta'minlash orqali valarray ob'ektlarining o'plam osti ob'ektlar bo'lgan ob'ektlarni o'qib olinadigan quvvatlaydigan yordamchi sinfi.

Businfob'ekti - valarray<Type> ob'ekti dagi elementlarni ketma ketligi bilan yoziladigan valarray <bool> sinfi ob'ekti bilan birgalikda valarray <Type> sinfi gah avolasaqlaydigan ob'ektlarni ta'sniflaydi.

Ketma-ketlik ba.size elementlardan ko'pmas. J element faqat ba[J] gachin (true) o'rnatilgan da gina faol bo'ladi. Ketma-ketlik da element nisoni ba elementlarga teng bo'ladi. Agar i - ba da eng kichik haqiqiy chin (true) elementning indeks bo'lsa, u holda ta'nlangan ketma-ketlik da a[i] ning qiymati nolgacha teng.

6.6-dastur. mask\_array sinfi sinfidan foydalanish.

```

#include "stdafx.h"
#include <valarray>
#include <iostream>
using namespace std;
int main( ){
    int i;
    valarray<int> va ( 10 );
    for ( i = 0 ; i < 10 ; i++ )
        va [ i ] = i*3;

    cout << "valarray elementlari: ( ";
        for ( i = 0 ; i < 10 ; i++ )
            cout << va [ i ] << '\t';
    cout << ")." << endl;

    va [(va > 20 || va < 10) ] = 10;
    cout << "o'zgartirikgan valarray elementlari: (
";
        for ( i = 0 ; i < 10 ; i++ )
            cout << va [ i ] << '\t';
    cout << ")." << endl;
    system("pause");
}

```

#### 6.6-dastur.Output

```

valarray elementlari:
( 0    3    6    9    12    15    18
21    24    27    ).
o'zgartirikgan valarray elementlari:
( 10   10   10   10   12   15   18
10   10   10   ).

```

In sinfi<sup>2</sup>.  
 Businfintervalmatematikaningamallarini bajarishgayo'naltilganbo'lib,  
 C++ningeskistandartiasosidakeltiriladivataqribiyhisoblashlaruchun yaratilgan.  
 Sinfquyidagichaaniqlangan:

<sup>2</sup>Muallif MO'MINOV Bahodir Boltaevich tomonidan 2013 yillarda tadqiqot ishlari uchun yaratilgan. Barcha xoxlovchilar uchun foydalanish mumkin.

```

class In {
    private:
    public:
        double _a, a_;
        double m,n;
    In() {_a=a_=0;}
    In(double a, double b) {
        if (a>b)
        { a=a+b;
          b=a-b;
          a=a-b;
        }
        _a=a, a_=b;
    }

    //kiritish va chqarish...
    friend istream& operator>>(istream & S, In &
d) throw()
    {
        S>>d._a>>d.a_;
        return S;
    }
    friend ostream& operator << (ostream & S, In
&d) throw()
    { S<<"["<<d._a<<" "<<d.a_<<"]\t";
      return S;}

    // In a = 9;
    inline In & In::operator= (const double & a)
    {
        _a = a_ =a;
        return *this;
    }
    // - ishorasi
    friend In operator -(In &a) throw()
    {
        return In(-a.a_,-a._a);
    }
}

```

Sinfda 2 takonstruktorbo‘lib, birinchisiintervalson 0 gatenglashtiradi,

ikkinchisimosparametrlargaasoslanibintervalsonniyaratidi.

In kutubxonasi funksiyalari:

class In – intervalsinf.

intervalning \_a, - quyichegarasi a\_ - yuqorichegarasi;

In() {\_a=a\_=0;} Intervalningqiymatini 0 gatenglashtirish;

In(double a, double b) Intervalgaqiymatberish. Agar \_a> a\_ bo‘lsa, qiymatlaro‘rni almashadi.

- 1) [ >> ] - intervalsonni o‘qish :: 1 2
- 2) [ << ] - intervalsonni chiqarish, masalan: [a,b]
- 3) [=] - intervalsonga o‘ddiysonni tenglashtirish, masalan: In a; a=1;
- 4) [-] - intervalsonni ishorasini almashtirish, masalan: In a; b=-a;
- 5) [+] - intervalqo‘shish, masalan: [a]+[b]; [a]+b; a+[b];
- 6) [-] - intervalayirish, masalan: [a]-[b]; [a]-b; a-[b];
- 7) [\*] - intervalko‘paytirish, masalan: [a]\*[b]; [a]\*b; a\*[b];
- 8) [/] - intervalbo‘lish, amallari [a]/[b]; [a]/b; a/[b];
- 9) [ += ] - operator intervalqo‘shish [a] += [b] [a] += b
- 10) [ -= ] - operator intervalayirish [a] -= [b] [a] -= b
- 11) [ \*= ] - operator intervalko‘paytirish [a] \*= [b] [a] \*= b
- 12) [ /= ] - operator intervalbo‘lish [a] /= [b] [a] /= b
- 13) [ [] ] - operator intervalyoki
- 14) [ & ] - operator intervalva
- 15) [ |= ] - operator intervalyoki
- 16) [ &= ] - operator intervalva
- 17) [ == ] - operator interval tenglik
- 18) [ != ] - operator interval tengmas
- 19) [ <= ] - operator interval kichik yoki teng
- 20) [ >= ] - operator interval kattayoki teng
- 21) [ < ] - operator interval kichik
- 22) [ > ] - operator interval katta
- 23) [ ! ] - operator interval 0 intervalga tegishligini aniqlaydi.
- 24) GetInf funksiyasi - intervalning kichik qiymati olish
- 25) GetSup funksiyasi - intervalning katta qiymati olish
- 26) SetInf funksiyasi - intervalning kichik qiymatini o‘zgartirish
- 27) SetSup funksiyasi - intervalning katta qiymatini o‘zgartirish
- 28) IsEmpty funksiyasi - intervalning xatoligi, masalan: [2,1] – xato;
- 29) abs funksiyasi - intervalmoduli

- 30) Midfunksiyasi -intervalningmarkazi
- 31) Radfunksiyasi -intervalningradusi
- 32) Widfunksiyasi -intervalninguzunligi  $2 * \text{Rad}$
- 33) MaxInfunksiyasi -intervalningkattasimodulda
- 34) MinInfunksiyasi -intervalningkichigimodulda
- 35) Singfunksiyasi -intervalningishorasi -1,0,1 ko‘rinishda

#### Indefinekutubxonasi

- 1) tab - "\t"dir.
- 2) NewLine - "\n"dir.
- 3) ZERO - 0.0000000qiymat
- 4) real - doubletipidir

#### ERRORkutubxonasi

- 1) DIV\_BY\_ZERO\_ERROR
- 2) ERROR\_INTERVAL\_EMPTY\_INTERVAL
- 3) ERROR\_INTERVAL\_SING\_INTERVAL
- 4) ERROR\_INTERVAL

#### INTMATH kutubxonasi

- 1) min funksiyasi – ikkihaqiqiysonningkichigi
- 2) max funksiyasi – ikkihaqiqiysonningkattasi
- 3) abs funksiyasi – haqiqiysonningmoduli

#### INMATH intervalmatematikakutubxonasidastandartqiymatlar.

- 1) #define E 2.7182818284590452354
- 2) #define LOG2E 1.4426950408889634074
- 3) #define LOG10E 0.43429448190325182765
- 4) #define LN2 0.69314718055994530942
- 5) #define LN10 2.30258509299404568402
- 6) #define PI 3.14159265358979323846
- 7) #define PI\_2 1.57079632679489661923
- 8) #define PI\_4 0.78539816339744830962
- 9) #define PI1\_ 0.31830988618379067154
- 10) #define PI2\_ 0.63661977236758134308
- 11) #define SQRTPI2 1.12837916709551257390
- 12) #define SQRT2 1.41421356237309504880
- 13) #define SQRT1\_2 0.70710678118654752440
- 14) sin – sin(x);
- 15) cos – cos(x);
- 16) tan – tan(x);
- 17) sinh – sinh(x);

- 18) cosh – cos(x);
- 19) tanh – tanh(x);
- 20) asin – asin(x);
- 21) acos – acos(x);
- 22) atan – atan(x);
- 23) exp – exp(x);
- 24) log – ln(a);
- 25) log10 – log(x);
- 26) pow(In &a, int n) funksiyasi – a intervalsonning n darajasi
- 27) pow(In &a, double n)
- 28) pow(const In &b, In &a)
- 29) pow(const double b, In &a)
- 30) pow(In &b, In &a)
- 31) sqrt (In &a)funksiyasiintervalsondankvadratildizolish
- 32) sqrt(In &a, double n)funksiyasiintervalsondan n-chidarajaliildizolish
- 33) ceil funksiyasiintervalsonniyaxlitlash
- 34) floor funksiyasi interval sonningkattayaxliylash
- 35) round funksiyasi interval sonnikichikkayaxlitlash

INDEFMURAKKAB hosilaolishhuchunkutubxona

class def\_murakkab murakkabhosilalarsinfi...

- 1) div\_Def\_uvfunksiyasi -  $u/v$  nihisoblash;
- 2) mult\_Def\_uvfunksiyasi -  $u*v$  nihisoblash;
- 3) add\_Def\_uv funksiyasi -  $u+v$  nihisoblash;
- 4) sub\_Def\_uv funksiyasi -  $u-v$  nihisoblash;
- 5) pow\_Def\_ufunksiyasi -  $f(x)$ ning d darajasining hosilasi;
- 6) f\_sqrtfunksiyasi - ildiz osti  $f(x)$ ning hosilasi

class def\_fx x ning n darajasinig hosilasi sinfi...

- 1) fx(In x) funksiyasi - x ning 1 darajasini qaytaradi;
- 2) fx(In x, int n)funksiyasi - x ning n darajasini qaytaradi;
- 3) dfx(In x, int n) funksiyasi - x ning n darajasini hosilasi;
- 4) dfx(In x, int n, int m) funksiyasi- x ning n darajasini m tartibli hosilasi;

class def\_func – matematik funksiyalardan hosila;

- 1) d\_exp(In x,int m)funksiyasi – exp(x) ning m tartibli hosilasi;
- 2) d\_exp(In x, In k, int m)funksiyasi – exp(kx) ning m tartibli hosilasi;
- 3) const\_a(In b)funksiyasi – o‘zgarmas interval sonning hosilasi;

- 4) `f_ax(In a, In x,int m)`funksiyasi-  $a^x$  ning  $m$  tartibli hosilasi;
- 5) `f_ax(In a, In x,int m, int k)` funksiyasi- $a^{kx}$  ning  $m$  tartibli hosilasi;
- 6) `f_sin(In x,int m)` funksiyasi- $\sin$  ning  $m$  tartibli hosilasi;
- 7) `f_cos(In x,int m)`funksiyasi-  $\cos$  ning  $m$  tartibli hosilasi;
- 8) `f_sqrt(float x, int m)`funksiyasi- $\sqrt{\phantom{x}}$  ning  $m$  tartibli hosilasi;
- 9) `f_ln(In x, int m)` funksiyasi-  $\ln(x)$  ning  $m$  tartibli hosilasi;
- 10) `f_log(float a, In x, int m)` funksiyasi- $\log(a,x)$  ning  $m$  tartibli hosilasi;
- 11) `f_tan(In x)` funksiyasi- $\tan(x)$  ning 1 tartibli hosilasi;
- 12) `f_ctan(In x)` funksiyasi- $\cot(x)$  ning 1 tartibli hosilasi;
- 13) `f_acos(In x)` funksiyasi- $\arccos(x)$  ning 1 tartibli hosilasi;
- 14) `f_asin(In x)` funksiyasi- $\arcsin(x)$  ning 1 tartibli hosilasi;
- 15) `f_atan(In x)` funksiyasi- $\arctan(x)$  ning 1 tartibli hosilasi;
- 16) `f_actan(In x)` funksiyasi- $\operatorname{arccot}(x)$  ning 1 tartibli hosilasi;
- 17) `f_cosh(In x)` funksiyasi-giperbolik  $\cos(x)$  ning 1 tartibli hosilasi;
- 18) `f_sinh(In x)` funksiyasi-giperbolik  $\sin(x)$  ning 1 tartibli hosilasi;
- 19) `f_tanh(In x)` funksiyasi-giperbolik  $\tan(x)$  ning 1 tartibli hosilasi;
- 20) `f_ctanh(In x)` funksiyasi-giperbolik  $\cot(x)$  ning 1 tartibli hosilasi;

`class def_arg_func` argumentida  $f(x)$  bo‘lgan funksiyalar hosilasi sinf;

- 1) `d_exp(func f, func df, In x)` funksiyasi- $\exp(f(x))$  ning hosilasi;
- 2) `f_ax(In x, double a, func f, func df)` funksiyasi- $a^{f(x)}$  ning hosilasi;
- 3) `f_ax(In x, In a, func f, func df)` funksiyasi- $a^{f(x)}$  ning hosilasi;
- 4) `f_sin(In x, func f, func df)` funksiyasi- $\sin(f(x))$  ning hosilasi;
- 5) `f_cos(In x,func f, func df)` funksiyasi- $\cos(f(x))$  ning hosilasi;
- 6) `f_sqrt(In x, func f, func df)` funksiyasi- $\sqrt{f(x)}$  ning hosilasi;
- 7) `f_ln(In x, func f, func df)` funksiyasi- $\ln(f(x))$  ning hosilasi;
- 8) `f_log(float a, In x, func f, func df)` funksiyasi- $\log(a,(f(x)))$  ning hosilasi;
- 9) `f_log(In a, In x, func f, func df)` funksiyasi- $\log(a,(f(x)))$  ning hosilasi;
- 10) `f_tan(In x, func f, func df)` funksiyasi- $\tan(f(x))$  ning hosilasi;
- 11) `f_ctan(In x, func f, func df)` funksiyasi- $\cot(f(x))$  ning hosilasi;

- 12) `f_acos`(In x,func f, func df) funksiyasi-`acos(f(x))` ning hosilasi;
- 13) `f_asin`(In x,func f, func df) funksiyasi-`asin(f(x))` ning hosilasi;
- 14) `f_atan`(In x, func f, func df) funksiyasi-`atan(f(x))` ning hosilasi;
- 15) `f_actan`(In x,func f, func df)funksiyasi- `actan(f(x))` ning hosilasi;
- 16) `f_cosh`(In x,func f, func df) funksiyasi-`cosh(f(x))` ning hosilasi;
- 17) `f_sinh`(In x, func f, func df) funksiyasi-`sinh(f(x))` ning hosilasi;
- 18) `f_tanh`(In x, func f, func df) funksiyasi-`tanh(f(x))` ning hosilasi;
- 19) `f_ctanh`(In x, func f, func df) funksiyasi-`ctanh(f(x))` ning hosilasi;
- 20) `pow_arg_func`(func f, func df, func dx, In x, int d) funksiyasi-`(f(x)^d)` ning hosilasi;
- 21) `f_msqrt`(func f, func df, In x, float m) funksiyasi-`m_sqrt(f(x))` ( $1/m$ ) ning hosilasi;

**INGAUSSK\_H** – Integralni Gauss usulida hisoblash uchun koyfisentlar uchun kutubxona

- 1) `P` – ko‘p hadni hisoblash;
- 2) `dP` – ko‘p hadni hosilasini hisoblash
- 3) `A` – Koyfisentlarni hisoblash;
- 4) `x0` – `x` uchun boshlang‘ish yaqinlashish;
- 5) `xeps` – `x` ni eps aniqlikda olish uchun;
- 6) `convert_x` – olingan `x` larni `a,b` oraliqqa convertlash;
- 7) `convert_A` - olingan `A` larni `a,b` oraliqqa convertlash;

**INTEGRALIN\_H** interval integrallash uchun kutubxonaintegral\_MTT1 – markaziy to‘g‘ri to‘rtburchak usuli;

- 1) `integral_Trap`funksiyasi– trapetsiya usuli;
- 2) `integral_Simson`funksiyasi – Simson usuli;
- 3) `integral_gauss1`funksiyasi – Gauss usuli;
- 4) `integral_gauss2`funksiyasi - Gauss usuli ikki karrali

**INGAUSS\_H CHATS**ni gauss usuli bilan yechish uchun kutubxona `N = 3` uchun

- 1) `PrintArray` – massivni chiqarish;
- 2) `PrintVector` – vektorni chiqarish;
- 3) `ScanArray` – massivni kiritish;



- 4) ScanVectorB – vektorni kiritish;
- 5) SwapColumns – ustunlarni almashtirish;
- 6) SwapLines – qatorlarni almashtirish;
- 7) Destroy – quyi uchburchakni )ga aylantirish;
- 8) Solution – yechimni topish
- 9) F – yechimning aniqligini aniqlash;
- 10) GAUSS – birlashgan Gauss usulini qo‘llash;

6.7-dastur. Chiziqialgebraiktenglamalarsistemasini yechish dasturi.

```
#include <iostream.h>
#include <conio.h>
#include <e:\mylibs\interval.h>

int main(){
int n;
cout<<"n ni kirit:>>";
cin>>n;
In a[n][n],b[n];
//_____A ni kirit:_____
for(int i=0; i<n; i++)
for(int j=0; j<n; j++)
a[i][j]=KiritIn();
//_____b ni kirit:_____
for(int j=0; j<n; j++)
b[j]=KiritIn();
//_____to'g'ri yo'l_____
In r[n][n];
for(int j=0; j<n-1; j++)
for(int i=j+1; i<n; i++){
r[i][j] =DivIn(a[i][j],a[j][j]);
for(int k=j; k<n; k++)
a[i][k]=SubIn(a[i][k],MultIn(r[i][j],a[j][k]));
b[i]=SubIn(b[i],MultIn(r[i][j],b[j])); }
//_____a[i][i]:_____
for(int j=0; j<n; j++){
b[j] =DivIn(b[j],a[j][j]);
for(int i=0; i<n; i++)
r[i][j] =DivIn(a[i][j],a[i][i]); }
```

```

//_____a[][]_____convert_
for(int j=0; j<n; j++)
    for(int i=0; i<n; i++)
        a[i][j] = r[i][j];
//_____ out _____
for(int i=0; i<n; i++) {
    for(int j=0; j<n; j++)
        { cout<<"["; chopIn(a[i][j]); cout<<"]\t"; }
        cout<<"= ["; chopIn(b[i]); cout<<"]\t";
        cout<<endl;}

In x[n],s;
for(int i=n-1; i>-1; i-- )
    { s. _a=0; s. a_=0;
      for( int j=n-1; j>i; j-- )
          s=AddIn(s, MultIn(a[i][j],x[j]));
          x[i]=DivIn(SubIn(b[i],s),a[i][i]);
    }

for(int i=0; i<n; i++){
    cout<<"["; chopIn(x[i]); cout<<"]" <<endl;}

getch();
return 0;
}

```

6.7-dastur.

Ikkikarraliintegralnigaussusulibilantaqribiyhisoblashdasturi.

```

#include <iostream.h>
#include <conio.h>
#include <math.h>
double P(double x, float n){
    if (n==0) return 1;
    if (n==1) return x;
    return (((2*n-1)/n)*x*P(x,n-1))-(((n-1)/n)*P(x,n-2));
}

```

```

double dP(double x, float n){
    return ((n/(1-x*x))*P(x,n-1))-(x*P(x,n));
}

double A(double x, float n){
    return 2./((dP(x,n)*dP(x,n))*(1-x*x));
}

double x0(int i, double n){
    return -cos(M_PI*((4*(i+1)-1)/(4*n+2))); }

double xeps(double x, double n, double eps){
    double temp=x-P(x,n)/dP(x,n);
    while (fabs(temp-x)>eps) {
        x=temp;
        temp=x-P(x,n)/dP(x,n);
    }
    return x;
}

double convert_x(double xx, float a, float b){
    return (0.5*(b+a))+(0.5*(b-a)*xx);
}

double convert_A(double AA, float a, float b){
    return ((b-a)/2)*AA;
}

int main(){
    float n,eps;
    freopen("InputData.in","r",stdin);
    cin>>n; cin>>eps;
    float a,b; cin>>a>>b;
    float x[(int)n];
    freopen("OutputData[a,b].out","w",stdout);
    cout<<"*** --- ---- n="<<n<<" ---- --- ***"<<endl;
    cout<<"*** -- a="<<a<<" b="<<b<<" -- ***"<<endl;
    cout<<"*** -- -- --- --- --- --- --- --- ***"<<endl;
}

```

```

for(int i=0; i<n; i++) {
    x[i]=x0(i,n); cout<<i+1;
    x[i]=xeps(x[i],n,eps);
    printf(" :%.15f",convert_x(x[i],a,b));;
    printf(" :%.15f\n",convert_A(A(x[i],n),a,b));
}
//getch();
return 0;
}

```

### 6.7-dastur. Output

```

----- 6 -----
eps=5
1.40656 3.17244
2.2895:2.2895
2.22385
6
0.882941
***  _____ HISOBLANDI _____ ***
*** -- n=5 uchun -- ***
1 :-0.90963196754455566000 :0.19411758457043865000
2 :-0.54064083099365234000 :0.47043172906368108000
3 :-0.000000000000000006123 :0.56888888888888889000
4 :0.54064083099365234000 :0.47043172906368108000
5 :0.90963196754455566000 :0.19411758457043865000

-----

I=2.21087712506044950000

```

Ushbu insoniyatdailmiy matematik va matematika-fizika tenglamalarini taqribiy kompyuterda hisoblashlarini amalga oshirish uchun ishlatiladi. Bukutubxona aynan A254-

Sloyiha sirda ishlab chiqilgan.

Loyiha sirda vazifa sifatida fizika jarayonlari matematik modellashtirish vaularni sonli echimlarni olish uchun dasturiy ta'minotlarni ishlab chiqishga bag'ishlangan. Kutubxonada 10 dan ortiq mo'rosxo'rsinf, 200 dan ortiq funksiyalar, matematik funksiyalarning modifikatsiyalangan funksiyalaribor.

## NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1. Nima uchun sonlisonlarni yaratiladi?
2. Qanday sonlisonlarni bilasiz?

3. Sonlisonlarda amallarni bajarish uchun funksiyalarni qaysi usul yordami da aniqlanadi?

4. Kompleks va arifmetik amalar o'zaro hamkorlikda ishlaydimi?

5. Kompleksni arifmetik tip katenglashtirish mumkinmi?

6. Haqiqiy (real)

yoki mavhum qismini oddiy arifmetik tipga qiymat qilib berish mumkinmi?

7. Kompleks sonning real qismi qaysi funksiyabilanolinadi?

8. Kompleks sonning mavhum qismi qaysi funksiyabilanolinadi?

9. `double re = complex_obj.real();` va `double re = real(complex_obj);` sintaktiklarining farqibormi?

10. Kompleks sonning mavhum qismini qanday kiritiladi?

11. Kompleks sonni o'lik kiritish uchun qavs ichida real va mavhum qismlar qanday belgibilan ajratiladi?

12. Kompleks sonni qanday operandlarni o'lib quruv tlamaydi?

13. Valarray konteynerini nima uchun kerak?

14. Massiv element indekslarini o'yichabirlashtirish operatorlarini va elementlar aritmetik amallarni hisoblash sinfi ayting?

15. **apply()**

bu massiv elementlarini o'yichabir vaqt davomida o'zgarishlarni bajaradigan imoniqa ytaradi?

16. **cshift()**

berilgan son qiymatini o'yichaindeks asosida massivni qanday shaklida surib y angimassiv qaytaradi?

17. Valarray

tiplari bilan bajariladigan amallarni qayta aniqlash mumkinmi?

18. Kaysi inf


sinfining meros xo'ridir va biro' ichovlito'plamostilarni yaratishda ishlatiladi ?

19. slice sinfi qanday tipdagi ob'ektni avsiflovchi parametrlarni saqlaydi.

20. slice sinfiningkonstruktoridanechtaparamertborvaqaysilar, sanabbering?
21. Agar qism to'plam yordamida aniqlangan massiv doimiy valarrayning kichik bo'laga ibo'lsa, bu massiv yanginima?
22. **stride()** - valarray  
to'plam ostining elementlari orasidagi imani topadi.
23. Qaysi sinf valarray  
ning ko'pchilik to'plam osti bilan ishlashga mo'ljallangan.
24. Sinf gslice - Valarray  
bazaviy sinfdan olingan va nechtaparamertni qabul qiladi?
25. Agar to'plam gslice bilan aniqlangan va valarray  
ning to'plam osti bo'lsa, gslice yanginim hisoblanadi.
26. gslice – bir nechta qism dani tashkil topgan valarray uchun to'plam osti  
valarrayni aniqlaydigan hammasi belgilangan bir element indeksidan boshlanadi  
dimiyoki xar xil element indeksidan boshlanadimi?
27. gslice i slice ning farqlarini madani borat.
28. Qaysi sinf Valarray  
qismlari bilan aniqlangan to'plam osti massivlar uchun amallarni qo'llab-quvvatlaydi.
29. Shablon sinf muayyan valarray  
amallar tomonidan bilvosita yaratilgan va dastur dabevosita foydalanishni  
mumkinmi yoki mumkin emasmi?
30. Bazaviy valarray sinfelementlari orasidagi to'plam osti bo'lib,  
belgilangan to'plam osti o'rtasida amallarni ta'minlash orqali valarray  
objektlarining to'plam osti bo'lgan objektlarni qo'llab-  
quvvatlaydigan yordamchisi ni ayting?
31. Tanlangan elementlar ketma ketligi orasida Indirect\_array  
sinf funksiyalari valarray  
uchun belgilangan tegishli funksiyalar qanday ishlaydi.
32. Bazaviy valarray sinfelementlari orasidagi to'plam osti bo'lib,  
belgilangan to'plam osti uchun mantiqiy amallarni ta'minlash orqali valarray  
objektlarining to'plam osti bo'lgan objektlarni qo'llab-  
quvvatlaydigan yordamchisi ni ayting?
33. Interval matematikaning amallari ni bajarishga yo'naltirilgan sinfnomi  
ni ayting?
34. In sinfdanechtakonstruktor borvaular qanday ishlaydi?
35. In(double a, double b) intervalga qiymat berish funksiyasida agar \_a>  
a\_ bo'lsa, nima vazifa bajariladi.



## AMALIY KO'NIKMA VA MALAKALARNI ANIQLASH HAMDA RIVOJLANTIRISH UCHUN ASSISMENT TOPSHIRIQLARI.

BIRINCHI ASSISMENT TOPSHIRIG'I	
	<p>Komplekssonlarsinfiga oid berilgan quyidagi dastur bo'yicha berilgan topshiriqlar kerakli fragmentlari soida bajarang.</p> <p>👉 Bundagi dasturda giba'zio'zgartirishlarni topish orqali topshiriqlar bosqichma bosqich amalga oshiriladi.</p>
dastur	topshiriqlar
<pre>// Created by MBBahodir #include "stdafx.h" #include &lt;complex&gt;  using namespace std;  int main(){     complex&lt; double &gt; a, b, c;     cout &lt;&lt; "Complex sinfi oid turli t=farmatli 3 ta son kiritibg:" &lt;&lt; endl;     cin &gt;&gt; a &gt;&gt; b &gt;&gt; c;     cout &lt;&lt; "Complex sinfiga oid sonlar:" &lt;&lt; endl;     cout &lt;&lt; a &lt;&lt; b &lt;&lt; c &lt;&lt; endl;     system("pause"); }</pre>	<p>1. Dasturda xatosini toping.</p> <hr/> <hr/> <hr/>
	<p>2. Dasturda komplekssonlardan iborat massivni yarating.</p> <hr/> <hr/> <hr/> <hr/>
	<p>3. Komplekssonlarni yig'indisini hisoblovchi funksiyani yarating.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>

4. Komplekssonlar ustida matematika  
mallarni bajaruvchi funksiyalarni yarat

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

5. Dasturda jamibo'lib, nechao'zgartirish kiritildi. \_\_\_\_\_
6. Shudasturning analogini yaratish sizga mustaqil vazifa dir.



<b>IKKINCHI ASSIMENT TOPSHIRIG'I</b>	
🚩	<p>Valarray sonsinfi va uning funksiyalariga oid berilgan quyidagi dastur bo'yicha berilgan topshiriqlar kerakli fragmentlari asosida bajarilgani.</p> <p>👉 Bunday dasturda giba 'zio' zgartirishlarni topish orqali topshiriqlar bosqichma bosqich amalga oshiriladi.</p>
dastur	topshiriqlar
<pre>#include "stdafx.h" #include&lt;iostream&gt; #include&lt;valarray&gt; using namespace std; int main(){     int myints[] = { 5, 25, 55, 85, 115 };     valarray&lt;int&gt; varr (myints, sizeof(myints)/sizeof(m yints[0]));     valarray&lt;int&gt; varr_one ;      varr_one = varr.apply([](int x){return x=x+5;});      cout &lt;&lt; "Massivning yangi qiymatlar: ";     for (int &amp;x: varr_one) cout &lt;&lt; '\t' &lt;&lt; x;     cout &lt;&lt; endl;</pre>	<p>1. Dastur xatolarini toping.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
	<p>2. Dasturda <code>varr_one = varr.apply([](int x){return x=x+5;});</code> nimavazifani amalga oshiradi.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>

```

    cout << "Oldingi
massiv sum() => ";
    cout << varr.sum()
<< endl;
    cout << "Yangi
massiv sum() => ";
    cout <<
varr_one.sum() << endl;

cout << "min() => ";
    cout << varr.min()
<< endl;
    cout << "max() =>
";
    cout << varr.max()
<< endl;

varr_one =
varr.shift(2);
    cout << "Massivning
shift(2) qiymatllar: ";
    for (int &x:
varr_one) cout << '\t'
<< x;
    cout << endl;

    varr_one =
varr.cshift(-2);
    cout << "Massivning
cshift(-2) qiymatllar:
";
    for (int &x:
varr_one) cout << '\t'
<< x;
    cout << endl;

    system("pause");
    return 0;

```

3. Dastur daberilgan valarrayning kichik va eng katta elementlarifarqini topuvchi dastur fragmentini yozing.

---



---



---



---



---



---

4. Ikki ta massivni o'zaro solishtirib, teng bo'lsa 0, katta bo'lsa 1, kichik bo'lsa -1 qilib valarray massivi yarating.

---



---



---



---

5. Massivlarni teskaricha qarishdastur fragmentlarini yozing.

---



---



---



---



---



---



---




---

}	
<p>6. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____</p> <p>7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.</p>	

<b>UChINChIASSIMENTTOPShIRIG‘I</b>	
🎓	<p><b>Slice</b>  sinfigaoidberilganquyidagidasturbo‘yichaberil  gantopshiriqlarkeraklifragmentlariasosidabajar  ing.  👉 Bundaydasturdagiba‘zio‘zgartirishlarnitopish  orqalitopshiriqlarbosqichma _____  bosqichamalanganoshiriladi.</p>
dastur	topshiriqlar
<pre>// Created by MBBahodir #include "stdafx.h"  #include &lt;valarray&gt; #include &lt;iostream&gt; using namespace std; int main( ){     int i;     size_t sizeVA, sizeVAR;     size_t startVAR, strideVAR;      valarray&lt;int&gt; va(28), vaResult;      for ( i = 0 ; i &lt; 28 ; i += 1 )         va [ i ] =</pre>	<p>1. Dasturdanechtayangitipdagiob‘  ektlariyaratilgan.</p> <hr/> <hr/> <hr/> <hr/>
	<p>2. Dasturdagibarchavaelementlari  niekrangachiqaring.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>


<pre> (rand()%100)+1;      sizeVA = va.size();     cout &lt;&lt; "valarray elementlar soni: " &lt;&lt; sizeVA &lt;&lt; "." &lt;&lt; endl &lt;&lt; endl;     // slice konstruktori     slice vaSlice( 0, 6, 4);     vaResult = va[vaSlice];      cout &lt;&lt; "valarray to'plam ostisi vaResult = " &lt;&lt; "va[slice( 0, 6, 4)] =\n ( ";     for ( i = 0 ; i &lt; vaSlice.size() ; i++ )         cout &lt;&lt; vaResult [ i ] &lt;&lt; " ";     cout &lt;&lt; ")." &lt;&lt; endl;      sizeVAR = vaSlice.size( );     cout &lt;&lt; "to'plam ostining elementlar soni: " &lt;&lt; sizeVAR &lt;&lt; "." &lt;&lt; endl;      startVAR = vaSlice.start( );     cout &lt;&lt; "to'plam ostining birinchi </pre>	<p>3. Massivdan 2 indeksdan 4 taelementNqadambilanyangimassive hiqarishmumkin. N ningqiymatlarinianiqlang.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <p>4. vaSlice.size( );ningtipiqanday?</p> <hr/> <hr/> <hr/> <hr/> <p>5. Massivdanberilganqiymatniizl ashdasturfragmentiniyozing.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
---	---

<pre> element indeksi: " &lt;&lt; startVAR &lt;&lt; "." &lt;&lt; endl;      strideVAR = vaSlice.stride( );     cout &lt;&lt; "to'plam ostining elementlari orasidagi masofa: " &lt;&lt; strideVAR &lt;&lt; "." &lt;&lt; endl;      system("pause"); } </pre>	
<p>6. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____</p> <p>7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.</p>	

<b>TO‘RTINChIASSISMENTTOPShIRIG‘I</b>	
	<p>gslice sinfigaoidberilganquyidagidasturbo‘yichaberil gantopshiriqlarkeraklifragmentlariasosidabajar ing. ☞ Bundasturdagiba‘zio‘zgartirishlarnitopish orqalitopshiriqlarbosqichma bosqichamalganoshiriladi.</p>
dastur	topshiriqlar
<pre> // Created by MBBahodir #include "stdafx.h" #include &lt;valarray&gt; #include &lt;iostream&gt; using namespace std; int main(){     int i; </pre>	<p>1. Dasturdanechtayangitipdagiob’ ektlariyaratilgan.</p> <hr/> <hr/> <hr/> <hr/> <hr/>

<pre> valarray&lt;int&gt; va ( 20 ), vaResult; for ( i = 0 ; i &lt; 20 ; i+=1 )     va [ i ] = i+1;  valarray&lt;size_t&gt; Len (2), Stride(2); Len [0] = 6; Len [1] = 4; Stride [0] = 2; Stride [1] = 4; </pre>	<p>2. Dasturdagi barcha va elementlari niekrangachiqaring.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
<pre> gslice vaGSlice (0, Len, Stride ); vaResult = va[vaGSlice];  cout &lt;&lt; " vaGSlice asosida vaResult elementlari:" &lt;&lt; endl &lt;&lt; "va[vaGSlice] = (";  for ( i = 0 ; i &lt; 10 ; i++ )     cout &lt;&lt; '\t' &lt;&lt; vaResult [ i ];     cout &lt;&lt; ")" &lt;&lt; endl;  const valarray &lt;size_t&gt; sizeGS = vaGSlice.size ( );  cout &lt;&lt; "vaResult elementlar soni:" &lt;&lt; " vaGSlice.size ( ) = ( "; for ( i = 0 ; i &lt; </pre>	<p>3. Massivdan <span style="float: right;">2</span>  indeksdan (4,6) ta element (N,M) qada mbilanyangimassivchiqarishmumkin . (N,M) ning qiymatlarini aniqlang.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
<pre> const valarray &lt;size_t&gt; sizeGS = vaGSlice.size ( );  cout &lt;&lt; "vaResult elementlar soni:" &lt;&lt; " vaGSlice.size ( ) = ( "; for ( i = 0 ; i &lt; </pre>	<p>4. vaSlice.size( ); ning tipi qanday va natijasichi ?</p> <hr/> <hr/> <hr/>
<pre> cout &lt;&lt; "vaResult elementlar soni:" &lt;&lt; " vaGSlice.size ( ) = ( "; for ( i = 0 ; i &lt; </pre>	<p>5. Massivdan berilgan qiymatni izlash dastur fragmentini yozing.</p> <hr/> <hr/> <hr/> <hr/>


<pre> 2 ; i++ )         cout &lt;&lt; sizeGS[ i ] &lt;&lt; '\t';         cout &lt;&lt; ")." &lt;&lt; endl;          size_t vaGSstart = vaGSlice.start ( );          cout &lt;&lt; "vaResult uchun birinchi element indeksi: " &lt;&lt; vaGSstart &lt;&lt; "." &lt;&lt; endl;          const valarray &lt;size_t&gt; strideGS = vaGSlice.stride ( );          cout &lt;&lt; "vaResult ychun qadamlar:" &lt;&lt; " vaGSlice.stride ( ) = ( ";         for ( i = 0 ; i &lt; 2 ; i++ )                 cout &lt;&lt; strideGS[ i ] &lt;&lt; '\t'; cout &lt;&lt; ")." &lt;&lt; endl;         system("pause"); } </pre>	<hr/> <hr/> <hr/> <hr/> <hr/>
<p>6. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____</p> <p>7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.</p>	

<b>BESHINCHI ASSIMENT TOPSHIRIG‘I</b>	
	<p>indirect_arraysinfigaoidberilganquyidagidastu rbo‘yichaberilgantopshiriqlarkeraklifragmentl ariasosidabajaring.</p> <p>👉 Bundaydasturdagiba‘zio‘zgartirishlarnitopish</p>

orqalitopshiriqlar bosqichma bosqichamalganoshiriladi.	
dastur	topshiriqlar
<pre>// Created by MBBahodir #include "stdafx.h" #include &lt;valarray&gt; #include &lt;iostream&gt;  int main( ) {     using namespace std;     int i;      valarray&lt;int&gt; va ( 10 );     for ( i = 0 ; i &lt; 10 ; i += 2 )         va [ i ] = i;     for ( i = 1 ; i &lt; 10 ; i += 2 )         va [ i ] = -1;      valarray&lt;size_t&gt; indx(5);     int j = 0;     for (int i = 1; i &lt; 10; i+=2)     {         indx [j] = i;         //cout &lt;&lt; indx[j] &lt;&lt; '\t';         j++;     }     cout &lt;&lt; endl;     va[indx] = 10;</pre>	<p>1. Dasturdanechtayangitipdagi ob'ektlari yaratilgan.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <p>2. Dasturdagi barcha va elementlari niekrangachi qaring.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <p>3. Massivdan 2, 4 indeksli elementlarni qiymatini 100 o'zgartirish dasturini tuzing.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>



<pre> cout &lt;&lt; "0'zgartirilgan valarray elementlari: ( ";     for (i = 0 ; i &lt; 10 ; i++ )         cout &lt;&lt; va [ i ] &lt;&lt; '\t'; cout &lt;&lt; ")." &lt;&lt; endl;     system("pause"); } </pre>	<p>4. va[indx] = 10;ningtipiqanday?</p> <hr/> <hr/> <hr/>
<p>5. Dasturdajamibo'lib, nechao'zgartirishkiritildi. _____</p> <p>6. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.</p>	

<b>OLTINCHASSISMENTTOPSHIRIG'I</b>	
	<p><b>mask_array</b>  sinfigaoidberilganquyidagidasturbo'yichaberilgantopshiriqlarkeraklifragmentlariasosidabajar ing.  👉 Bundaydasturdagiba'zio'zgartirishlarnitopish orqalitopshiriqlarbosqichma bosqichamalganoshiriladi.</p>
dastur	topshiriqlar
<pre> // Created by MBBahodir #include "stdafx.h" #include &lt;valarray&gt; #include &lt;iostream&gt; using namespace std; int main( ){     int i;     valarray&lt;int&gt; va ( 10 );     for ( i = 0 ; i &lt; 10 ; i++ )         va [ i ] = i*3;      va [(va &gt; 20    va &lt; </pre>	<p>1. Dasturdanechtayangitipdagiob'ektlariyaratilgan.</p> <hr/> <hr/> <hr/> <hr/> <p>2. Dasturdagibarchavaelementlari niekrangachiqaring.</p> <hr/> <hr/> <hr/>



qiymatqaytarishasosidagirekursivkamayishusuli,  
parserqurishvazifalarivausullarikeltirilganbo‘lib,  
nazariybilimlarniasoslashuchundasturlartuzibko‘rsatilgan.

bilimlarnimustahkamlashuchun 35  
tanazariysavolvaamaliyko‘nikmavamalakalarnirivojlantrishuchun 1  
taassimenttopshirig‘ivaharassimentda 6 tatopshiriqberilgan.  
Butopshiriqlarnibajarishmavzunimustahkamlashuchunxizmatqiladi.

*✍Kalitso‘zlar.* Oqim, satr, iterator, sinf, shablon,  
matematikamallar, fayl, EOF, rekursiya, iofda, qoida, grammatika.

*☑Bilishshartbo‘lgantushunchalar.* Tiptushunchasi,  
sinfvasinfob‘ekti, oqim, kirishelementgamurojaat,  
funksiyavako‘rsatkich, ma‘lumotlarnikirishvachiqishi,  
dasturlashgaoiddastlabkitushunchalarvaC++tiliniqo‘llabquvvatlovchimu  
hitdaishlashnibilishlozim.

*👉Bilibolasiz.* Sintaktiktahlillovchivazifasi, turli grammatikalar,  
dasturlashtilinianalizatoriningishlashtamoyili,  
pastvayuqorisathlargayo‘naltirilganalgoritmalar, rekursivkamayishusuli,  
analizatorlarniqurishdata‘rifvashartlarniqurishni, parsersinfi, FIRST  
to‘planningqurishalgoritmi, chaprekursivgrammatikalar,  
chaprekursivliknibartarafetishalgoritmi,  
qiymatqaytarishasosidagirekursivkamayishusuli,  
sintaktikanalizatorniyaratish, Forward va Input  
iteratorlardanfoydalanish, Strin – Stdin oqimlardanfoydalanish,  
sintaktiktahlillovchilarningbazaviyfunksiyalarinio‘rganishingiz mumkin.

## REJA

1. Ifodalarning sintaktik tahlili.
2. Parsersinfi.
3. Sintaktikanalizatorni yaratish.

## KIRISH

Bugungikungakelibdasturlashtillariningrivojlanishigaularnisintakti  
ktahlilqiluvchikonstruktorlarningko‘payib, ommalashganidir.  
Harbirdasturlashtilito‘g‘ridasturtuzilishinibelgilovchiqoidalarmajmuasiy  
ordamidatasvirlanadi.

Dasturlashtilining sintaktik konstruksiyalarinitasvirlashuchun engqulayfor  
malusulmatnli-erkingrammatikadir (masalan, Backus-Naur  
ningnormalshaklikengqo‘llaniladi).

Grammatikalartildanfoydalanadigandasturchilarsavutiluchunkom  
pilyatorlaryaratuvchilarningmuammolarinibirxildahalqilishgayordamber  
adi:

-  
Grammatikadasturlashtilininganiqvaosontushunadigansintaktikxususiyatlarinitalaydi.

- Ba'zigrammatikasinflaruchun, avtomatikravishdamanbadasturisintaktikto'g'riyokiyo'qliginianiqlaydigan samaraliparser (analizator, tahlillovchi) qurishmumkin.

- Aniqqurilgangrammatikadasturlashtiliganmanbadasturinito'g'riob'ektkodigatarjimaqilishvaxatolarnianiqlashuchunfoydalibo'lgantuzilmaniberishimumkin.

- Grammatikaasosidaishlabchiqilgankompilyatorlarjudaosonkengaytirilishimumkin (butilriyojlanishinatijasidapaydobo'lganyangikonstruksiyalarniqo'shishuchunayniqsafoydalidir).

Yanabirborta'kidlaymizki, matnli-erkingrammatikalardasturlashtiliningfaqatmatnli tarkibiyqismini, ya'nimuayyanilqurilishi qanday yozilganligini aniqlaydi. Dasturningsintaktikto'g'riligini aniqlashning yanabir muhim qismi dasturdatiplardan foydalanishning to'g'riligini matnli-erkingrammatikayordamida aniqlab bo'lmaydi. Shuning uchun, agar dastur grammatika sito'liq deb ishlansa, ubutunlaysintaktik jihatlardan to'g'rida qanday bo'lmaydi.

**Ifodalarning sintaktik tahlili.** Sintaktik tahlil degan dasturiy qilingan grammatika galeks emalarning ma'lum ketma-ketligi hosil qilgan tilga tegishli ekanligini belgilovchi jarayondir. Amalda, har qanday grammatika dan parser qurish giz mumkin, ammo amalda ishlatiladigan grammatika lar maxsus shaklga ega. Masalan, nuzunlikdagikiruvchisatru chun murakkabligi  $O(n^3)$  dan oshmasligi kerak bo'lgan har qanday kontekst-erkingrammatika uchun analizator (parser) qurish mumkin.

Lekin ko'phollarda tez ishlaydigan analizator qurish imkonini beradigan berilgan dasturlashtiliasosida grammatika qurish mumkin.

Amaliyotda til analizatorlari odatdacha ziq limurakkablikka ega. Masalan, dastur dachapdano'ngga qarab bittat erminal belgiga (leksik sinf) oldinga qarab ko'rish orqali erishiladi.

Sintaktik tahlillovchi kirish parametrlari - bu lekse mavajad vllarketma-ketligi, masalan, ichki tasvirlangan jadval, bu jadval sintaktik tahlillovchi uchun chiqish parametrlaridir.

Sintaktik tahlillovchi chiqish parametrlari —  
 jadvallar vatahlildaraxti hisoblanadi, masalan,  
 kompilyatorning keyingiko‘rishi uchun chiquvchi bo‘lib hisoblangan identifikatorlar jadvali va tiplar jadvali (masalan, tiplinazoratni amalga oshiruvchi ko‘rinish bo‘lishi mumkin).

Leksik va sintaktik tahlil bosqichlarining alohida qarashlarga ajratilishi shart emas. Odatdabufazalar bir xil ko‘rinishdabir-biribilano‘zarota‘sirlashadi.

Bunday ko‘rinishning asosiy fazasi ajralish fazasi bo‘lib hisoblanadi va sintaktik analizator leksika analizator gaharsafar boshqat terminal belgik erak bo‘lganda murojaat qiladi.

Sintaktik tahlillovchi sinflari.

Judako‘ptahlillovchi algoritmlarni quyidagi ikki sinflarning birigategishlidir .

1. top-down algoritmlari (past sathlargayo‘naltirilgan);
2. bottom-up algoritmlari (yuqori sathlargayo‘naltirilgan);

Buterminlarning kelib chiqishi sintaktik daraxt tugunlarining yaratilishu sulibilan bog‘liq: ildizdan (grammatika aksiomalari) yuqori tugunlargacha (terminal simvollar), yoki yuqori tugunlardan ildizgacha.

Past sathlargayo‘naltirilgan analizatorlar chiqishni qurish uchun grammatika aksioma dan boshlab vaterminal simvollar zanjiribilantugaydi.

Analizatorlar quyidagi xususiyatlarga ega bo‘lgan LL grammatika bilan bog‘liq:

- unatijaber maydigana analizatorlarga ega bo‘lishi mumkin;
- birinchi L harfi kiruvchi zanjirning chapdano‘nga qarab o‘qilishini bildiradi (left-to-right scan);
- ikkinchi L harfi - zanjirning chap chiqishini qurilayotganini bildiradi (*leftmost* derivation).

Past sathlargayo‘naltirilgan analizatorlarni yaratish juda qulay bo‘lib hisoblanadi, ularni qo‘ldah yaratish mumkin, masalan, rekursiv kamayishu sulibilan. LL

grammatika bilan ishlash va o‘zgartirish murakkab emas. LL grammatika bo‘lmagan grammatikalarda analizatorlar uchun rekursiv kamayishu sularidan foydalanish mumkin.

Boshqat omondan past sathlargayo‘naltirilgan analizatorlarga qaragand ayuqori sathlargayo‘naltirilgan analizatorlarko‘proqishlatiladi.

Chunki ularko‘proq grammatikalarni tahlil qilaoladi.

Shuning uchun bu analizatorlar uchun analizatorlarni yaratuvchi maxsus dastu

rlarmavjud. yuqorisathlargayo‘naltirilgananalizatorlarbilan LR – grammatikabog‘liq. Bundagi L harfioldingidek, kiruvchizanjirnichapdano‘nga qarabo‘qilishinibildiradi (left-to-right scan), R harfiesa – zanjirningo‘ngchiqishiqurilayotganinibildiradi (*rightmost derivation*). Bugungikunda LR – grammatikagaasoslanganalizatorlarbilanjudako‘pdasturlashtillarifoydalanmoqda.

**Rekursivkamayishusuli.** Engoddiyvajudako‘pfoydalanilganpastsathlargayo‘naltirilgananalizatorlarniqurishusulirekursivkamayishusulidir (recursive descent method).

Rekursivkamayishusuliasosiytamoyillarinio‘rganishuchunarifmetiki fodalarnibajarilishmasalasiniquaraymiz. Ulargabinaramallariqo‘shish (+), ayrish (-), ko‘pytirish (\*), butunbo‘lish (/) vaqavslaramallarikirsin. Odatdagidek, ko‘paytirishvabo‘lishamallariningustivorligitengvaularningustivorligiqo‘shishvaayirishamallariningustivorligidankattabo‘lib, buamallarningustivorligihamtengdir. Qo‘shishturidagiamallarga (+) va (-) - amallarini, ko‘paytirishturidagiamallargaesa (\*) va (/) - amallariqilibbelgilaymiz.

Qavslaramalistan darta rtibinio‘zgartirishuchunishlatiladi.

Vazifamizifodaqiymatinihisoblovchidasturyozishdaniborat.

Qaralayotganifodani quyidagiko‘rinishdabo‘lsin

$$T_1 + T_2 + \dots + T_n$$

Bunda  $T_i = F_{1i} * F_{2i} * \dots * F_{mi}$  ko‘rinishidagiifoda. Shuningdek,  $F_{ji}$  – busonyoqavsi chidagiifoda.

Berilganifodanihisoblashjarayoninio‘ylabko‘ramiz, bundabirinchi galda  $F_{11}$  hisoblanadiva  $F_{11}$  dankeyinqaysiamalturganinianiqlanadi. Agar bu ko‘paytirish amalibo‘lsa, chap operandinibilib, o‘ng operandinianiqlaymizvaamal nibajaramiz. Shundayqilib, ko‘paytirishamali uchunchap operandinianiqlaymiz.  $F_1 * F_2 * \dots * F_n$  amallarketmaketliginihisoblabbo‘lgandankeyin, keyingiamalsifatidabo‘lishamal inikelsa, bunihamyuqoridagijarayonbo‘yichahisoblaymiz.

Ifodalarni hisoblashdabarchahisoblashlarni quyidagisinflargaajratish mumkin:

1. Oddiyifoda, sonlarvaqavslarbilanberilganchiziqliifodalar, masalan, 28, (128-25+16);

2. Ifoda, ko'paytirishtipidagiamallaribo'lganifoda, ko'paytirishvabo'lishamallarikiradi, maslan,  $2*15, 28*(15/3)+3$ ;
3. Ifoda, qo'shishtipidagiamallaribo'lganifoda, qo'shishvaayirishamallarikiradi, masalan,  $5-3, (5*3)/10-2$ ;

Bularasodahisoblashjarayoninitasavurqilishimizmumkinvaquyidag iformulabilanifodalaymiz.

Expression (Term (Factor ()));

BundaFactor – oddiyifodanihisoblashprotsedurasibo'lib, son, qavslargaegachiziqliifoda, Term - ko'paytirishamalitipidagiamallarnisaqlovchiiifodaqiymatinihisoblashprot sedurasi, Expression - qo'shishamalitipidagiamallarnisaqlovchiiifodaqiymatinihisoblashprotsed urasi.

Oddiyifodanihisoblashprotsedurasi:

```
int Factor ()
{
    char ch = getChar();
    if (isDigit (ch)) return getValue(ch);
    if (ch == '(')
    {
        int result = Formula();
        if (getChar() == ')') return result;
        error ("kutilmagan belgi yoki amal");
        return 0;
    }
    return error ("kutilmagan belgi yoki amal");
}
```

Dasturko'paytirishtipigaoidamallarnihisoblashuchunifodanitahlilqila di.

Ko'paytirishtipigaoidamallarnihisoblashuchunifodanihisoblashningumu miyformulasiquyidagicha:

$$F_1 * F_2 * \dots * F_n$$

Aniqtushunishimizkeraki, hisoblashbajarilayotganamallarfaqatko'paytirishyokibo'lishiamalibo'ladi . BundayamallarnihisoblashuchunTerm protsedurasiniyaratishmumkin. Buprotseduraningparametrlaributunsoniliqiymatlarbo'lishikerak. Bundachapoperandanitanlashimizvakerakliamalnibajarishuchunsonyoki oddiyifodadaniborato'ngoperandanianiqlashimizlozim,

soʻngesaamal nibajarish mumkin.

Buni quyidagi dastur fragmentiasosida amalga oshirish mumkin.

```
int Term (int left)
{
    char ch = getChar(); int right;
    if (ch != '*' && ch != '/')
    {
        /* navbatdagi amal kutilmagan boʻlsa, uni
qaytarishimiz kerak*/
        returnChar(); /*kutilmagan belgini qaytarish */
        return left; /*kutilmagan qimatni qaytarish */
    }
    /* Hammasi yaxshi boʻlsa, oʻng operandni aniqlah
kerak*/
    right = Factor();
    if (ch == '*')
    {
        return Term(left * right);
        /* bu hisoblashni bajarishni aniqlaydi */
    }
    if (right == 0) return error ("Nolga boʻlinish");
    return Term(left / right);
}
```

Dastur qoʻshishti tipiga oid amallarni hisoblash uchun ifoda ni tahlil qiladi. Qoʻshishti tipiga oid amallarni hisoblash uchun ifoda ni hisoblashning umumiy formulasi quyidagicha:

$$T_1 + T_2 + \dots + T_n$$

Aniqtushunimiz keraki,

hisoblash bajarilayotgan amallar faqat qoʻshish yoki ayirish amal boʻladi.

Bunday amallarni hisoblash uchun Expression

protsedurasini yaratish mumkin.

Bu protseduraning parametrlari butun sonli qiymatlar boʻlishi kerak.

Bunda chap operandni tanlashimiz va kerakli amal nibajarish uchun son yoki oddiy ifoda dan iborat oʻng operandni aniqlashimiz lozim,

soʻngesaamal nibajarish mumkin.

Buni quyidagi dastur fragmentiasosida amalga oshirish mumkin.

```
int Expression (int left)
{
```



```

char ch = getChar (); int right;
if (ch != '+' && ch != '-')
{
    /* navbatdagi amal kutilmagan bo'lsa, uni
qaytarishmiz kerak*/
    returnChar(); /*kutilmagan belgini qaytarish */
    return left; /*kutilmagan qimatni qaytarish */
}
/* Hammasi yaxshi bo'lsa, o'ng operandni aniqlah
kerak*/
right = Term (Factor());
if (ch == '+')
{
    return Expression(left + right);
/* bu hisoblashni bajarishni aniqlaydi */
}
return Expression(left - right);
}

```

Yuqoridagi protseduralar ishlatilishi uchun quyidagi usullarni mustaqil ishlatib chiqish kerak. Umuman olganda rekursiv kamayish usuli qiziq tirgan edi, o'ylaymizki, quyidagi fragmentlarni yaratib buning uchun hosil qilassiz.

1. Kirishoqimidan navbatdagi belgini kutuvchi, `getChar` parametrsiz usuli.

2. Kutilmagan belgini kirishoqimigacha qarib berish `returnChar` usuli.

3. `isDigit` va `getValue` usullarini ham yozish shart emas. Agar parametrison bo'lsa, `isDigit` birinchi usul `true` qaytaradi. Ikkinchi usul esa parametr sifatida kirishoqimidan berilgan satrdan sonlarni ajratib olib qaytaradi.

4. `Error` usuli xatolar haqidagi xabarlarni chiqarish uchun ishlatiladi.

Bu usullarni mustaqil hal qilish mumkin, ammo `.NET` standart sinflar usullari asosida bazalarini modifikatsiya lab yaratish mumkin.

Bu dasturlar fragmentini shakllantirishda qiymat qaytarish asosidagi rekursiv kamayish usulidan foydalanildi (*recursive descent with backtracking*). Qiymat qaytarmaslik asosidagi rekursiv kamayish usulidan foydalanib qolmaganimizni ko'rib chiqamiz.

**Rekursiv kamayish usulidan foydalanish sharti.** Qiymat qaytarmaslik asosidagi rekursiv kamayish usulidan faqat quyidagi shart bajarilganda foydalanish mumkin.

**Shart:** har bir ifodaning birinchi belgisibuholatda qaysi ifodaga amal qilishini aniqlash uchun yetarlib o'lishi kerak.

Bu shartni aniqlash uchun **FIRST**

to'plamini aniqlash orqali amalga oshirish mumkin.

Ta'rif:

KS

grammatika uchun terminal va terminal bo'lmagan belgilardan iborat

G

grammatika va

w

zanjir asosida

$FIRST_k(w)$

to'plamni quyidagicha aniqlaymiz:

$$FIRST_k(w) = \{x \mid w \Rightarrow^* xv, |x| = k \text{ yoki } w \Rightarrow^* x, |x| < k\},$$

k – natural son.

Boshqacha qilib aytganda,

$FIRST_k$

(w)

to'plam

w

dan iborat bo'lgan terminal zanjirlarining uzunligi

k

bo'lgan barcha terminal prefikslardan iborat.

Universal algoritmi tildatiplarining kichik qismini yaratadigan grammatika uchun ko'rib chiqaylik:

type  $\rightarrow$  imple

type  $\rightarrow$  ^id

type  $\rightarrow$  array[simple] of type

simple  $\rightarrow$  integer

simple  $\rightarrow$  char

simple  $\rightarrow$  num ... num

Bu grammatika uchun quyidagilarni aniqlash mumkin:

$$FIRST_1(\text{simple}) = \{\text{integer}, \text{char}, \text{num}\}$$

$$FIRST_1(\text{^id}) = \{\text{^}\}$$

$$FIRST_1(\text{array [simple] of type}) = \{\text{array}\}$$

Agar w zanjir faqat terminallardan iborat bo'lsa,  $FIRST_k(w)$  - w zanjirda birinchi k belgilardir, aks holda  $|w| \geq k$ , yoki agar  $|w| < k$  bo'lsa, w zanjirning o'zi.

**Parsers infi.**

**FIRST**

**to'plamning qurish algoritmi.** Birinchi navbatda belgilar grammatikasi uchun **FIRST** to'plamini aniqlaymiz.

1-qadam. Agar x terminal bo'lsa,  $FIRST(x) = x$ ;

2-qadam.  $x \rightarrow \epsilon$  qoida uchun  $FIRST(x)$  to'plamga  $\epsilon$  ni qo'shamiz;

3-qadam. Agar x terminal emas va grammatika qoidasi  $x \rightarrow y_1 y_2 \dots y_k$  bo'lsa,  $FIRST(x)$  ga terminalni qo'shamiz, agar qandaydir i uchun terminal  $FIRST(y_i)$  ga tegishli bo'lsa va  $\epsilon - FIRST(y_1), \dots, FIRST(y_{i-1})$  to'plamlarga tegishli bo'lsa,  $y_1 y_2 \dots y_{i-1} \Rightarrow^* \epsilon$ , agar barcha  $j=1, 2, \dots, k$  uchun  $\epsilon - FIRST(y_j)$  to'plamga tegishli bo'lsa,  $\epsilon$  ni  $FIRST(y)$  to'plamga qo'shamiz.

FIRST(w) to'plamni qurish algoritminishakllantiramiz.

Kirish: KS grammatika  $G=(N, T, P, S)$  va  $w$  zanjirning terminal va terminal bo'lmagan belgilari.

Chiqish: FIRST(w).

Usul: FIRST (X<sub>1</sub>) dan barcha bo'sh bo'lmagan belgilarni FIRST (X<sub>1</sub> X<sub>2</sub> ... X<sub>k</sub>) ga qo'shamiz. Agar  $\epsilon$  - FIRST (X<sub>1</sub>) ga tegishli bo'lsa, FIRST (X<sub>2</sub>) dan barcha bo'sh bo'lmagan belgilarni, shu bilan iteratsiya ni davom ettiramiz. Oxirida, Agar barcha  $j$  uchun FIRST (X<sub>j</sub>) bo'sh belgiga ega bo'lsa, FIRST (X<sub>1</sub> X<sub>2</sub> ... X<sub>k</sub>) to'plamga  $\epsilon$  ni qo'shamiz.

**Misol:** Quyidagi qoidalar bilan berilgan grammatika qaraymiz:

$S \rightarrow BA$

$S \rightarrow +BA$

$A \rightarrow \epsilon$

$B \rightarrow DC$

$C \rightarrow *DC$

$C \rightarrow \epsilon$

$D \rightarrow (S)$

$D \rightarrow a$

Ushbu grammatika uchun FIRST to'plami quyidagicha aniqlanadi.

FIRST(D) = { (, a }, FIRST(C) = { \*,  $\epsilon$  }, FIRST(B) = FIRST(D), FIRST(A) = { +,  $\epsilon$  }, FIRST(S) = { (, a }.

LL(k)-grammatika ko'rib chiqamiz.

Ta'rif:  $G = (VT, VN, P, S)$  grammatika LL(k)-grammatika deb atiladi, agar  $FIRST^k(x) = FIRST^k(y)$  teng,  $u = u_1$  bo'lsa va ikkita ixtiyoriy chap chiqishlar uchun ifoda o'rinli bo'lsa,

$S \Rightarrow^* wAv \Rightarrow wuv \Rightarrow^* wx$

$S \Rightarrow^* wAv \Rightarrow wu_1v \Rightarrow^* wy$

$Av$  dan chiquvchi terminal va terminal bo'lmagan belgilar va  $k$  birinchi belgilardan (agar mavjud bo'lsa) iborat  $wAv$  joriy zanjir uchun mavjud bo'lsa,  $w$  bilan boshlanadigan va aytilgan  $k$  terminal bilan davom etadigan chiqishdagi qoida terminal zanjirini olish uchun aqo'llash mumkin bo'lgan kamidabitta qoida mavjud.

Qoidaga asoslangan grammatika ni qarymiz:

$S \rightarrow aAS$

$S \rightarrow b$

$A \rightarrow a$

$A \rightarrow bSA$

vaikkitachiqish

$$(1) \quad S \Rightarrow^* wSv \Rightarrow wuv \Rightarrow^* wx$$

$$(2) \quad S \Rightarrow^* wSv \Rightarrow wu_1v \Rightarrow^* wy.$$

Agarxvayzanjirlarabilanboshlansin. Buesachiqishda  $S \rightarrow aAS$  qoidaishtiroketganinibildiradi. Haqiqattanhamu  $= u1 = aAS$ . Agarx va y zanjirlar  $b$  bilanboshlansin. Buesachiqishda  $S \rightarrow b$  qoidaishtiroketganinibildiradi. Haqiqattanhamu  $= u1 = b$ . Bundayko‘rinadiki, ikkitachiqishamallario‘xshashvatengkuchlidir. Bundayesayuqoridagigrammatika LL(1) grammatikaxususiyatlariniqo‘llabquvvatlaydi.

LL(1)

grammatikagaasoslanganQiymatqaytarmaslikasosidagirekursivkamayish usuliuchuntahlilovchiquyidagigrammatika, Masalan,

$$(1) S \rightarrow \text{if } S \text{ then } S \text{ else } S$$

$$(2) S \rightarrow \text{begin } S L$$

$$(3) S \rightarrow \text{print } E$$

$$(4) L \rightarrow \text{end}$$

$$(5) L \rightarrow ; SL$$

$$(6) E \rightarrow \text{num} = \text{num}$$

LL(1)

grammatikagaasoslanganrekursivkamayishusuliuchunquyidagigrammatika kaniqaraymiz:

$$(1) S \rightarrow \text{if } E \text{ then } S \text{ else } S$$

$$(2) S \rightarrow \text{begin } S L$$

$$(3) S \rightarrow \text{print } E$$

$$(4) L \rightarrow \text{end}$$

$$(5) L \rightarrow ; S L$$

$$(6) E \rightarrow \text{num} = \text{num}$$

Bugrammatikaniqo‘llabquvvatlvchitahlilovchinirekursivkamayishu suliasoslanibyaratishmumkin.

Buninguchunterminalbo‘limgangrammatikauchunharbiriuchunbittadanp rotsedurayozishkerakbo‘ladi.

```
class SimpleParser {
    const int IF = 1;
    const int THEN = 2;
    const int ELSE = 3;
    const int BEGIN = 4;
    const int END = 5;
    const int PRINT = 6;
```

```

const int SEMICOLON = 7;
const int NUM = 8;
const int EQ = 9;

public static void nextStep(int lc){
    if (lexical_class == lc)
        lexical_class = getLC();
    else
        error();
}

public static void S(void){
    switch(getLC())
    {
    case IF:
        E(); nextStep(THEN); S();
nextStep(ELSE); S(); break;
    case BEGIN:
        S(); L(); break;
    case PRINT:
        E(); break;
    default:
        error(); break;
    }
}

public static void L(void){
    switch (lexical_class)
    {
    case END:
        getLC(); break;
    case SEMICOLON:
        getLC(); S(); L(); break;
    default:
        error(); break;
    }
}

```

```

public static void E(void){
    nextStep(NUM); nextStep(EQ); nextStep(NUM);
}

public static void main(void){
    lexical_class = getLC();
    S();
}
}

```

**Chaprekursivgrammatikalar.** LL (k) – xususiyatlarigrammatikauchunkuchlicheklovlar yuklaydi.

Natijadagrammatika LL(1) xususiyatigaegabo‘lishiuchunba‘zangrammatikanio‘zgartirishmumkin. Buniamalgaoshirishuchunhardoimhammuvaaffaqiyatlibo‘lmaydi, lekinbir LL(1) grammatikaolishimkoniyatigaegabo‘ldi, keyinrekursivkamayishusuliasosidaanalizatorqurishdafoydalanishingizm umkin. Farazqilaylik, quyidagigrammatikaasosidatahlillovchiqurishkerakbo‘lsin.

$$\begin{aligned}
E &\rightarrow E + T | E - T | T \\
T &\rightarrow T * F | T / F | F \\
F &\rightarrow \text{num} | (E)
\end{aligned}$$

FIRST(T) to‘plamningterminallari FIRST(E+T) to‘plamgategishlibo‘lsin. Buholdaprotseduralarningketma – ketchaqirishbirqiyimatlianiqlabbilmaymiz, chunkikirishzanjiridagilarnitahlilqilishmizkerak.

Muammoshundaki, terminalbo‘lmagan E ningo‘ngpozitsiyasidaqoidaningo‘ngtomoni, chapqismiham Edabo‘ladi. Bundayholatlardaterminalbo‘lmagan E – chaprekursivdebaytiladi.

Ta‘rif. Agargrammatikada  $A \Rightarrow^* Aw$  chiqishmavjudbo‘lsa, Aterminalbo‘lmagan KS grammatika G chaprekursivdebaytiladi.

Bittabo‘lsahamchaprekursivqoidagaegabo‘lgangrammatikaLL(1)-grammatikabo‘laolmaydi. Ikkinchitomondanbilamizki, KS tilbittabo‘lsahamchaprekursivbo‘lmangrammatikabilananiqlanadi.

**Chaprekursivliknibartarafetishalgoritmi.** To‘g‘ridan-to‘g‘richaprekursivliknibartarafetishalgoritminikel tiramiz.

$G = (N, T, P, S)$  – KS grammatikava  $A \rightarrow A\omega_1/A\omega_2|\dots|A\omega_n|v_1|v_2|\dots|v_m$  qoidalar P dagibarchaqoidalarniifodalasin, Aningchapqisminisaqlamaydi, chunkibirorbirzanjirv<sub>i</sub> –

Aterminalbo‘lmaganbilanboshlanmaydi.

N

to‘plamgayanabirtaerminalbo‘lmagan

A qo‘shamizvaAdachaptomondagibo‘lganqoidalarnio‘zgartiramiz.

Quyidagiqoidalarniolamiz.

$$A \rightarrow v_1 | v_2 | \dots | v_m / v_1 A' | v_2 A' | \dots | v_m A' /$$

$$A' \rightarrow \omega_1 | \omega_2 | \dots | \omega_m / \omega_1 A' | \omega_2 A' | \dots | \omega_m A' /$$

Olingangrammatikaoldingisigaekvivaletkanliginiko‘rsatishmumkin

. Yuqoridakeltirilgangrammatikagabuakslantirishniqo‘yib, arifmetikamallarniyozishuchunquyidagichagrammatikaniolamiz:

$$E \rightarrow T | TE'$$

$$E' \rightarrow +T | +TE'$$

$$T \rightarrow F | FT'$$

$$T' \rightarrow *F | *FT'$$

$$F \rightarrow (E) | \text{num}$$

TuzilgangrammatikaLL(1)

xususiyatlarigaegaekanliginiko‘rsatishmumkin.

Agarikkitaqoidahamterminalbo‘lmaganbirxilbelgibilanbo‘lsa, aniqmuammonivaziyatgakeltiradi, masalan,

$$S \rightarrow \text{if } E \text{ then } S \text{ else } S$$

$$S \rightarrow \text{if } E \text{ then } S$$

Bundayhollardayanabirterminalbo‘lmaganqoida qo‘shamiz, qaysikiyuqoridagivaziyatgaoydinlikkiritib,

farqliqoidalarnikeltiribchiqaradivaquyidagiqoidaniolishimizmumkin:

$$S \rightarrow \text{if } E \text{ then } S S'$$

$$S' \rightarrow$$

$$S' \rightarrow \text{else } S$$

Olingangrammatikarekursivkamayishusulibilanamalgaoshirilishimumkin.

**Qiymatqaytarishasosidagirekursivkamayishusuli.** Shundayqilib, qiymatqaytarishasosidagirekursivkamayishusuliniqo‘llashuchungrammatikani FIRST to‘plamlarkesishmaydiganshaklgaaylantirishkerak. Bumurakkabvako‘pvaqttalabqiladiganjarayondir. Shuninguchun, amaliyotda, tez-tezishlatiladiganiborabor, qiymatqaytarishasosidagirekursivkamayishhodisadebataladi.

Buninguchunleksikanalizatoran‘naviyusullarnito‘laligicha, scan va next

ana‘naviyusullarvanusxalashkonstruktorigaegabo‘lganob’ektsifatidanam oyonbo‘ladi.

Keyin,

noaniqlikyuzagakelishimumkinbo‘lganbarchaholatlarda,

tahlil qilishni boshlashdan oldin leksikanalizatorning hozirgi holatini aniqlaymiz (ya'ni, leksikanalizatorning nusxasini boshlaymiz) va matnning tahlil qilishni davom ettirishga harakat qilamiz.

Butahlil muvaffaqiyatsiz bo'lsa, leksikanalizator holatini tiklaymiz va shunday qilib, keyingi grammatika variantini yordamida yanashuzanjirni tahlil qilishga harakat qilina divatugaguncha davom ettiriladi. Barcha tahlilimkoniyatlarimuvaffaqiyatsiz bo'lsa, xatolar haqida xabar beramiz.

Butahlil lovluchini bu usul qiyamat qaytarmaydigan rekursiv kamayish usuli ko'ra potentsial sekinlikni o'choq-oydinko'ri mumkin, lekin vaziga uning asl shakli da grammatika da qolishi va dasturchi uchun harakatsaq lab qolish uchun bosh qarishosondir.

Xuddi shu sintaktik tahlil

bemoll kompilyatorida sintaktik tahlil uchun joriy qilingan. Masalan, quyidagi fragment datahlil paytida konstruktsiya ifoda operatoriyoki izoh kanligini aniqlovchi usul amalga oshiradi.

Ikki konstruktsiya nuqtabilan ajratilgan va ehtimol kvadrat qavslarni o'z ichiga olgan identifikatorlar ketma-ketligi bilan boshlanishi mumkin.

```
AST.Stmt stmt_or_decl ()
{
    Lexer saved = new Lexer (lexer);
    try {
        AST.Type t = type_opt ();
        if (lexer.Is (Token.Tag.Ident))
            return decl_tail (saved.Curr.coor, t);
    }
    catch (ParseFailed) {
        lexer = saved;
        AST.Expr expr = this.expr ();
        return new AST.Stmt.Expr (compiler,
            saved.Curr.coor | lexer.req
            (Token.Tag.Semicolon).coor, expr);
    }
}
```

try blokida unitav sifatde bhisoblab, konstruktsiya ajratishga harakat qilamiz. Bu muvaffaqiyatsiz bo'lsa, tahlil lovluchi istisnocha qiradi va keyin catch blok datahlil boshlanadi va shu konstruktsiya ifoda sifatida tahlil qilinadi.



**Sintaktikanalizatorni yaratish.** Odatda, mashinatilidagi matn gaplar, gap-gaplar ostidan, gap ostida, o'z navbatida, gaplar ostidan va shukabib belgilardan hosil bo'ladi. Masalan, XML hujjatda boshlanish tegi, qiymat, yopish teglar mavjud. Boshlanish tegi [<] belgisi, tegnomi va qanday atribut va qiymatlari, [>] belgisidan iborat. Yopish tegi esa, [</] belgisi, tegnomi, [>] belgisibilan tugaydi. Atribut sanomi, [=], [“], belgilar to'plamivayana [“] belgisidan iborat vatakrorlanishim mumkin.

Qiymat esa qandaydir belgilar to'plami yoki qandaydir elementlardan (gap, gap osti, ifoda) iborat. Tahlil natijasida tahlildaraxtini hosil qilish mumkin.

Bunday tillarni har bir terminal bo'lmagan tilning ma'lum bir jumlasiga mos keladigan Backus-Naur shakli (BNSh) yordamida tasvirlash qulay.

Dasturlarni yozganda, odatda ularni funksiyalar va funksiya ostilarga ajratamiz va sifatik tahlillovchi yozmoqchibo'lganimiz uchun,

har bir BNSh terminal bo'lmagan sintaktik tahlillovchimizning bitta funksiyasi gamoskelsin va har bir bunday funksiya:

- ushbu jumlaniberilgan pozitsiyadan ajratish gaharakat qilish;
- bu ishni bajar dimiyo'qminatijasini qaytarish;
- tahlil tugagan iyokixatosodir bo'lgan pozitsiyani qaytarish;
- tahlil natijasida olish kerak bo'lgan ba'zi qo'shimchama'lumotlarni qaytarish;

tarish;

Masalan, BNSh ko'rinishida `expr ::= expr1 expr2 expr3` funksiyani yozamiz:

```
bool read_expr(char *& p, ....){
    if(!read_expr1(p, ....))
        return false;
    // read_expr1() funksiyasi expr1 tahlil tugaga
    pozitsiyada p qo'yadi
    // bu pozitsiyada p bilan hech qanday amal
    bajarilmaydi.
    if(!read_expr2(p, ....))
        return false;
    if(!read_expr3(p, ....))
        return false;
    return true;
}
```

Davom sifatida, BNSh ko'rinishida `expr ::= expr1|expr2|expr3` funksiyani yozamiz:

```

bool read_expr(const char *& p, ....){
    const char * l = p;
    if(read_expr1(p, ....))
        return true;
    // expr1 tahlil jarayonida p xato bo'lgan joyni
    ko'rsatadi
    p = l;
    if(read_expr2(p, ....))
        return true;
    p = l;
    if(read_expr3(p, ....))
        return true;
    return false;
}

```

Busintaktahlilashqiymatqaytarishasosidagirekursivkamayishusuli gaasoslangan.

IxtiyoriyatniBNShko'rinishidakeltirishmumkin. Masalan, XML tilidagiquyidagichayozishmumkin:

```

element ::= '<identifier>'some_data'</identifier>'

```

Agaridentifikatorlarimoskelsabuhaqiqattanhamto'g'rikeladi.

Bundayamallarnisintaktiktahlillovchigaqo'shishmuammoemas. Masalan, terminalfunksiyalarquyidagichabo'lishimumkin:

```

bool read_fix_char(const char *& it, char c){
    if(!*it)    return false; // satr oxiri
    if(*it!=c)  return false;// boshqa simvol
                // xato bo'lganda joyiga qolish
    bajariladi
    it++;      // keyingisiga o'tish
    return true;
}
bool read_fix_str(const char * & it, const ch_t *
s){
    while(*it)// satr tugamagancha
        if(!*s)
            return true;
        else if(*it!=*s)
            return false;
        // xato bo'lganda joyiga qolish
}

```

```

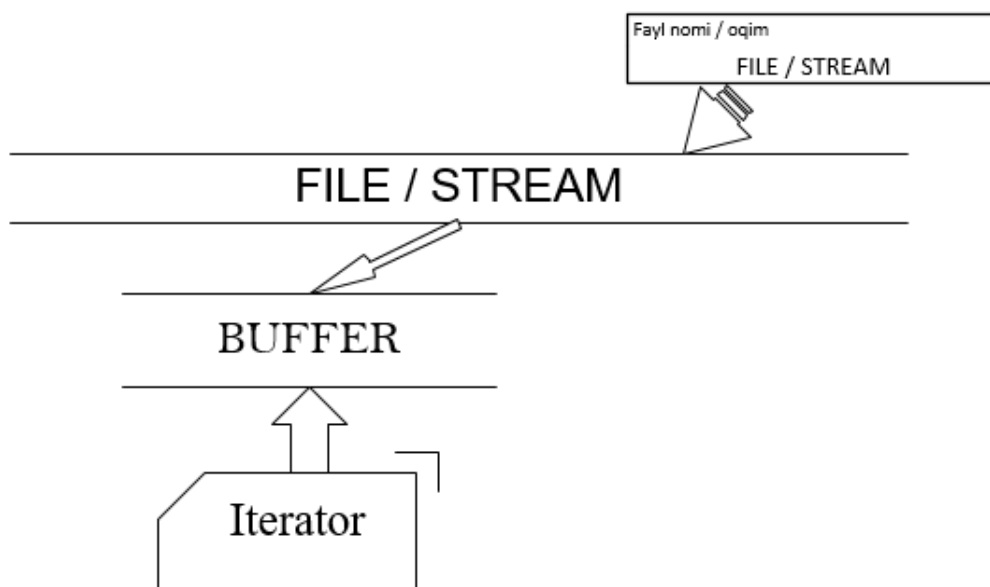
bajariladi
    else
        it++, C++;
    if(!*s) return true;
    return false;
}

```

Satrnisintaktiktahlilqilganimizdayuqoridagifunksiyalarningbarchatiplari uchun forward iteratorlaryetarliekanliginibiling.

forwarditeratorlarnitaqdimetadiganvafaylnibutunlayemas, fayloʻrnigakichikqismlargaxotiradasaqlaydigansinfoqimniyaratishniqaraymiz. Agarshablonlarasosidatahlillovchifunksiyaniyaratishnixoxlasak, ularnisatrlarva/yokioqimlardanfoydalanishmumkin (base\_parse.h kutubxonasiterminalfunksiyalariuchun, businfnidiskdagimavzugaoidpapkagaqarang). Birinchinavbatda unix «barchafayllarbor»: ideyalogiyasigabaʻzianiqliklarnikiritibolamiz: Shundayfayllarboʻladiki, ulardiskboʻyichajoylashganvaularniixtiyoriypozitsiyasidanbirnechamarta oʻqishmumkin(ularni random-access fayllardebaytiladi), shundayoqimlarboʻladiki, tarmoqdan, klaviaturadan, boshqailovalardanyoʻnaltirilgan (bularniforward oqimlardebaytiladi). Bundayoqimlarnibirmartagaoʻqishkerak. Shuninguchun, unisivabunisibilanishlashuchunbiryondashuvdafaylnioʻqiydigan, forwarditeratorlarniqoʻllabquvvatlaydigan, xotiradafaylnitoʻliqemas, balkimaʻlumbirboʻlaginisaqlaydigansinf-oqimniyaratishmaqsadgamuvofiq.

Bundayoqimlar Input iteratorlaruchunanchaoldinyaratilgan, maqsadifaqatularda Input iteratorlar, buiteratorharakatlanishiuchunbittabufferboʻladi. Qachonki, buferoxirigakelgandabufergafaylningkeyingiboʻlagiyuklanadihamdaitera torboʻshatilganbuferboshidanharakatlanishniboshlaydi (7.1-rasmgaqarang).



7.1-rasm. Input iteratorlardan foydalanish.

Forward iteratorlarning input iteratrlardan farqiularni nusxalash mumkinligidir.

Bunday masalada oqimlarda forward iteratorlardan foydalanish uchun buferlar ro'yxatini hosil qilish orqali echiladi.

Qandaydir iterator birinchi buferga murojaat bilan ro'yxatga yangi bufer qo'shiladi va fayldagi blok ma'lumotlar bilan to'ldiriladi.

Ammo bu usulda fayl to'liq xotiraga yuklanadi, bu holat kerakmas.

Shunday qilamizki,

buferning chap tomonidagi joylashgan oxirgi chap iterator joylangan barcha buferlar o'chirilsin. Lekin,

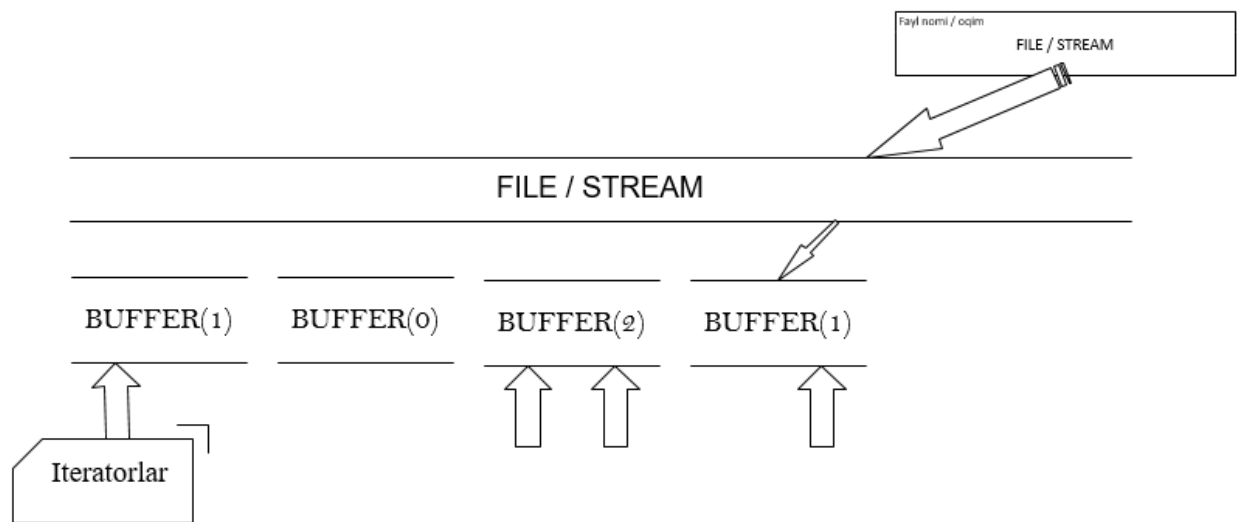
unga joylashgan bufer ro'yxat bilan har bir iterator mos bo'lmaydi,

lekin faqat ularning sonini mos bo'ladi (hisoblovchi iteratorlari).

Biror bir bufer 0 iterator qolganini va u engchaplikni bilsa,

u barcha unda o'ng buferdagi barcha qo'shni lari o'chiriladi. Chunki ular ham

0 iteratorga ega bo'ladi (7.2-rasmga qarang).



7.2-rasm. Forward iteratorlardan foydalanish.

Bunday kelib chiqadiki,

har bir iterator quyidagilarni o'z ichiga olish kerak:

- belgi ga ko'rsatkich (qaytanomlanishda qaytariladigan qiymat);
  - buferga dagi belgilarning massivning oxiriga ko'rsatkich iterator harakatlanganda taqqoslash amalga oshiriladi);
  - bufer o'xyatibo'yicha iteratorlar
- (bufer ma'lumotlariga murojjat uchun vaular orqali butun oqim ma'lumotlariga murojjat).

Iterator bufer chegarasiga yetganda, ukeyingibuferlarro'yxatdaborliginitekshiradi, bo'lmasa - uniyuklaydi, oldingibufer iteratorlar sonini kamaytiradi va keyingisi sonini oshiradi. Agar oldingibuferda 0 ta iterator qolgan bo'lsa, u o'chiriladi. Agar iterator faylning oxiriga chakelgan bo'lsa, u faqat oldingibufer dancha qiriladi va "faylning oxiri" holatiga o'tadi. Bir iteratorning nusxasini olganda joriy buferda iteratorlarning soni bittaga ko'payadi, iteratorning destruktoriga ishlaganda bittaga kammayadi, va, yana 0 bufer bo'lsa qolgan iteratorlar, u va o'ngda unkeyingibuferlar, shuningdek 0 iterator bo'lar o'chiriladi. Amalga oshirish uchun diskda mavjud papkadagi forward\_stream.h ga qarang.

Bunday iteratorlar anaviy iteratorlardan ularni ajratib turadigan ba'zi xususiyatlarga ega. Masalan, biroqimni (buferlar va ba'zi qo'shimcha ma'lumotlarro'yxatini saqlaydigan) o'chirish holdin (destruktorga tomonidan), barcha iteratorlar o'chirilgan bo'lishi kerak, ularni o'chirish vaqtida o'zlariga bog'liq bo'lmagan holda buferlarga o'z navb

atidao‘chirilganliginianiqlashmurojaatqiladi. Agar begin()  
 usulinichaqirish (birinchibuferniyaratish)  
 natijasidabirinchiiteratornibirmartaolsakvaushupaytgachabirinchibufferall  
 aqachono‘chiribtashlanganbo‘lsa, yana begin()  
 usuliyordamidaiteratorniololmaymiz. Bundantashqarioqimning end()  
 usuliyo‘q. Natijada, oqimdaichkiiteratoryaratishkerakbo‘ladi,  
 ya‘nibarchaoqimlaryaratilayotgandayaratiladiganvamosyozuvlar iter()  
 usuliyordamidaolinishimumkin. Bundantashqari,  
 algoritmlarniishlatganda, iteratorlarantez-teznusxalashkerakemas,  
 chunkixotirachapdano‘nggachabo‘lganiterator dabufferlarnisaqlaydi.  
 Kattafayllaruchunbukattaxotiratalabqilishigaolibkeladi.

Yordamsifatidaharxilturdagibufferlarmavjud: oddiy  
 (basic\_simple\_buffer), qator-ustunliiteratorni (basic\_adressed\_buffer).  
 Bularyordamidahisoblashuchunturlikodlashlarorasidaqaytakodlashnibaja  
 ruvchibufferqilishimumkin.

Shumunosabatbilanoqimbufferningtipibilanparametrlashtiriladi.

Buyruqlarsatriningoxirinullbelgibilanbelgilanadi.

Bushunianglatadiki, uningoxirinitopishuchunsatorqaliharakatqilsak,  
 ko‘rsatgichniboshqako‘rsatgichbilansolishtirishimizshartemas  
 (STLdaan‘anaviytarzdaamalgaoshiriladi), aksincha,  
 ko‘rsatgichimiztomonidanko‘rsatilganqiymatnitekshirishkerak.

Fayllarbilanbog‘liqvaziyathamshunday.

Belgikiritishdabelgilarasosidasonlarniolamizvaularkattaqiymatlarsatrig  
 aegabo‘lganiuchunyanaharbirbelgini EOF() funksiyabilantaqqoslanadi.  
 Realvaqtdakeladiganforwardoqimlaruchunfaylningoxiriholatioldindanno  
 ma’lum.

Bumuanmonisatrliko‘rsatgichlaruchun atend(it)  
 funksiyasinituzishorqaliuumlashtirishimumkin:

```
bool atend(const char * p)
{ return !*p; }
```

Agariteratorfaylningoxiriniko‘rsatsa, oqimbo‘yichaiteratorlarga true  
 qiymatqaytaradi.

Foydalanuvchibilaninteraktivhamkorlikqilishuchun (stdin orqali)  
 bloklibufferlashamalgaoshirilmaydi,  
 chunkiblokdaodatdabirnechtasatrjoylashtiriladivabittasatr kiritilgandanso  
 ‘ng, dasturfoydalanuvchidankirishnikutishnidavometadi,  
 chunkiblokhalito‘lmagan.

Satrningoxirigachabo‘lganbelgilarbuffernito‘ldiradivabuninguchunsatrlib  
 uferlashkerak.

Ushbukutubxonadaoqimishgatushirilgandafayltpinitanlabbuferlashturini tanlashingiz mumkin (basic\_block\_file\_on\_FILE yoki string\_file\_on\_FILE).

strin – stdin orqalisatrlibuferlashuchunbuferningichkiiteratoridir. Interaktivbo‘lmaganfayllaruchunoqimiyaratishdabirinchi belgigaishoraqiluvchiiteratoriyaratiladi, ya‘nibirinchibufer yuklanadi. Bungastrin uchunruxsatberilmaydi, chunkidasturchisatrtrejimidankiritishnikutishdanoldinbirorbirnarsaamalga oshirishyoki ekrangachiqarigim mumkin.

Shusababibirinchibufernito‘ldirishdasatrlifayllari ishlatilgandabelgil ibuyruq ‘\n‘danfoydalaniladi. Unio‘qishuchun, start\_read\_line (it) funksiyasimavjud, shundanso‘ngsatrni tahli qilishuchundastursatrnikiritishnikutishrejimigak eladivakeyingibelgi ‘\n‘tashqaridachiqishda iteratorbusatrniajratiboladi.

Agarbundankeyinyanafoydalanuvchidanma‘lumotlarkerakbo‘lsa, dasturchiyanaekrandabirornarsanichiqarishnixohlashim mumkinvauniolish danoldinyana start\_read\_line(strin) dnfoydalanishmumkin. Bundaquyidagichatakrorlanishhosilqilinadi:

```
while(true){
    cout << "Satr kirit:" <<endl;
    start_read_line(strin); //satr kiritish
    analys(strin);
}
```

Albatta, bufragmentfaqatqaytanomlashvaqtidabuferninyuklashda iteratorortalabqilish mumkin, lekinbuqaytanomlashqachonqo‘shimchatekshirishlarolibkelishivabutuntiziminimurakkablashtirishim mumkin.

7.1-dastur. STRIN Kutubxonasi.

```
#define ONE_SOURCE //strin.cpp
#include "../strin.h"

using namespace str;
using std::cout;
using std::endl;

#define ifnot(expr) if(!(expr))
#define r_if(expr) if((expr)==0)
```

```

#define r_while(expr)      while((expr)==0)
#define r_ifnot(expr)     if(expr)
#define r_whilenot(expr)  while(expr)

template <class it_t, class mes_t>
void dump(it_t tmp, mes_t mes){
    string s;
    read_fix_length(tmp,50,&s);
    std::cerr <<mes <<dump(s.c_str()) <<endl;
}

template<class it_t>
const char * read_sum(it_t & it, double * prez);

template<class it_t>
const char * read_expr(it_t & it, double * prez){
    const char * err;
    if(read_s_fix_char(it,'(')){
        r_ifnot(err = read_sum(it,prez))
            return err;
        ifnot(read_s_fix_char(it,')'))
            return "qavsni yoping";
        return 0;
    }
    int x;
    ifnot(read_dec(it,&x)){
        return "son";
    }
    *prez = x;
    return 0;
}

DEF_STRING(md,"*/")
template<class it_t>
const char * read_mul(it_t & it, double * prez){
    typedef char_type<it_t> ch_t;
    const char * err;

```



```

    r_ifnot(err = read_expr(it, prez))
        return err;
    while(true){
        ch_t zn;

        ifnot(read_s_charclass_c(it, make_span(md<ch_t>().
s), &zn))
            return 0;
        double x;
        r_ifnot(err=read_expr(it, &x))
            return err;
        if(zn==(ch_t) '*')    *prez *= x;
        else                  *prez /= x;
    }
    return 0;
}

DEF_STRING(pm, "+-")
template<class it_t>
const char * read_sum(it_t & it, double * prez){
    typedef char_type<it_t> ch_t;
    const char * err;
    r_ifnot(err = read_mul(it, prez))
        return err;
    while(true){
        ch_t zn;

        ifnot(read_s_charclass_c(it, make_span(pm<ch_t>().
s), &zn))
            return 0;
        double x;
        r_ifnot(err = read_mul(it, &x))
            return err;
        if(zn==(ch_t) '+')    *prez += x;
        else                  *prez -= x;
    }
    return 0;
}

```

```

int main(){
    using namespace std;
    if(0)    {
        string str_expression_1 = "5+84/(51)";
        string str_expression_2 = "15 + (16 /6 *4)";
        string str_expression_3 = "7 + ( 4* 4)";
        string str_expression = str_expression_3;
        cout <<"hisoblash ifodasi:" <<endl
<<str_expression <<endl;
        const char * p= str_expression.c_str();
        const char * err;
        double rez;
        r_ifnot(err=read_sum(p,&rez)){
            int pos=p-str_expression.c_str();
            for(int i=1; i<pos; i++)
                cout <<' ';
            cout <<'^' <<endl;
            cout <<"Xatolik: " <<err <<endl;
            return -1;
        }
        cout << "natija: " << rez << endl;
        cout <<"======" << endl;
    }

    if(1)    {
        try{
            while(!atend(strin)){
                cout <<"arifmetik ifoda kiriting va
'end' bilan tugating " <<endl;
                read_start_line(strin);
                const char * err;
                double rez=0;
                err=read_sum(strin,&rez);
                linecol lc=get_linecol(strin);
                read_until_str(strin,"end");
                r_if(err)
                    cout << "natija: " << rez <<

```

```

endl;
        else{
            cout <<" zanjir " <<lc <<"
xatolik: " <<err <<endl;
        }
    }
}
catch(const char * mes){
    cerr << "xatolik: " << mes << endl;
    return -1;
}
catch(string & mes){
    cerr << "xatolik: " << mes << endl;
    return -1;
}
catch(stream_exception & mes){
    cerr << "xatolik: " << mes.what() <<
endl;
    return -1;
}
catch(...){
    cerr << "noaniq xatolik" << endl;
}
}
return 0;
}

#ifndef STRIN_H
#define STRIN_H

#include <exception>
#include "base_parse.h"
#include "forward_stream.h"

#ifdef ONE_SOURCE
#   define STRIN_EXTERN
#else
#   define STRIN_EXTERN extern

```

```

#endif

namespace str{

    STRIN_EXTERN forward_adressed_stream STREAMin
#ifdef ONE_SOURCE
    (true, new file_on_FILE_stringbuf(stdin))
#endif
    ;
    STRIN_EXTERN forward_adressed_stream::iterator &
strin = STREAMin.iter();
    inline int __set01(typename
forward_adressed_stream::iterator & it){
        set_linecol(it,linecol(0,1));
        return 0;
    }
    STRIN_EXTERN int __unused_int
#ifdef ONE_SOURCE
    = __set01(strin)
#endif
    ;

class str_error{
#define constructor str_error
typedef str_error my_t;
protected:
    const char * message;
public:
    constructor():message(0){}
    constructor(const char * mes):message(mes){}
    constructor(int x):message(0){ if(x)throw "???"
int"; }
    constructor(bool x) :message(x?0:""){}
    operator bool()const{ return message==0; }
    const char * what()const { return message; }
#undef constructor
};
DECLARE_AND_OR_default_error_operators(str_error)

```

```

DECLARE_AND_OR_error_operators_for_short_circuiting(s
tr_error)
str_error operator>>(base_parse_error from, str_error
to){
    if(from) return str_error(true);
    elsereturn to;
}

class strpos_error : public str_error{
#define constructor strpos_error
typedef strpos_error my_t;
typedef str_error base_t;
    linecol lc;
    friend strpos_error operator&&(strpos_error l,
strpos_error r);
    friend strpos_error operator||(strpos_error l,
strpos_error r);
public:
    linecol ignored_lc=linecol(0,0);
    bool ignored()const{return lc==ignored_lc; }
    constructor(){};
    constructor(const char * mes):base_t(mes){}
    constructor(int x) :base_t(x){}
    constructor(bool x) :base_t(x){}

    constructor(const linecol & l)
        :lc(l){};
    constructor(const char * mes, const linecol & l)
        :base_t(mes),lc(l){};
    constructor(bool mes, const linecol & l)
        :base_t(mes),lc(l){};

    linecol where()const
    { return lc; }
    my_t & set_what(const char * mes)
    { base_t::message=mes;return *this; }
    template<class it_t>
    my_t & set_ignored(const it_t & it)

```

```

    {   ignored_lc =linecol(it);return *this;   }
#undef constructor
};

template <class it_t>
strpos_error operator>>(str_error from, const it_t *
to){
    return strpos_error(from.what(),linecol(*to));
}
std::ostream & operator<<(std::ostream & str, const
strpos_error & e){
    return str<<"("<<"ignored:"<<e.ignored()<<" at
"<<e.ignored_lc<<)"<<(bool)e<<";"<<(e?"OK":e.what())
<<" at "<<e.where();
}
strpos_error operator&&(strpos_error l, strpos_error
r){
    r.set_ignored(l.ignored_lc);
    if(l)
        if(r)
            if(r.ignored())
                return l;
            else
                return r;   /*TT*/
        else
            return r;   /*TF*/
    else
        if(r)
            return l;   /*FT*/
        else
            if(r.ignored())
                return l;
            else
                return r;   /*FF*/
}
strpos_error operator||(strpos_error l, strpos_error
r){
    r.set_ignored(l.ignored_lc);

```

```

    if(l)
        if(r)
            if(r.ignored())
                return l;
            else
                return r;    /*TT*/
        else
            return l;    /*TF*/
    else
        if(r)
            return r;    /*FT*/
        else
            if(r.ignored())
                return l;
            else
                return r;    /*FF*/
}
DECLARE_AND_OR_error_operators_for_short_circuiting(s
trpos_error)

} // namespace str
#endif // STRIN_H

```

Sintaktik tahlil o'vchilarning bazaviy funksiyalari.

Harsafar terminal funksiyalarni yozish dasturchilarga noqulaylik keltirmasligi uchun «base\_parse.h» sinfidatayyorlab qo'yilgan. Hozirda umumiy ko'rinishda (xaqiqiy sonlardan tashqari) tayyorlangan vakelajak dasatrb o'yichako'rsatkichlar va oqimbo'yicha iterat orlar uchun satrlifunksiyalar (strcmp, strstr, strchr, strspn, strcspn kabi) ishlaydigan qismining tuzish rejalashtirilgan. Shuningdek bu faylning oxiri haqida o'ylash kerak emas, faqat fragmentni to'g'ri yozsabo'ladi.

Quyidagi parsingning bazaviy funksiyalari qisqachakeltirilgan va ikki test parserni amalga oshirish paytida ularning foydalanish statistikasini aniq javob beradigan qiymatlarini faylga dastur keltirilgan.

```

size_t      n
ch_t        c
ch_t *      s
func_obj    bool is(c) // bool isspace(c)

```

span	spn
bispan	bspn
func_obj	err pf(it*) // int read_spc(it*)
func_obj	err pf(it*, rez*)

Len – belgilarsoni, \*pstr gaqo‘shilgan, satroxiriyokioxiriemasligianiqlovchistatistikavaqiymatlariqaytaradi.

int read_until_eof	(it&)	.*\$	0	0	1	OK
int read_until_eof	(it&, pstr*)	.*\$	len	len		OK
int read_fix_length	(it&, n)	.{n}	-1	0		OK
int read_fix_length	(it&, n, pstr*)	.{n}	-(1+len)	0		2 OK
int read_fix_str	(it&, s)	str	-(1+len)	0 ili (1+len)	9	OK
int read_fix_char	(it&, c)	c	-1	0 yoki 1	11	OK
int read_charclass	(it&, is)	[ ]	-1	0 yoki 1		OK
int read_charclass	(it&, spn)	[ ]	-1	0 yoki 1		OK
int read_charclass	(it&, bspn)	[ ]	-1	0 yoki 1		OK
int read_charclass_s	(it&, is, pstr*)	[ ]	-1	0 yoki 1		OK
int read_charclass_s	(it&, spn, pstr*)	[ ]	-1	0 yoki 1		1 OK
int read_charclass_s	(it&, bspn, pstr*)	[ ]	-1	0 yoki 1		5 OK
int read_charclass_c	(it&, is, ch*)	[ ]	-1	0 yoki 1		OK
int read_charclass_c	(it&, spn, ch*)	[ ]	-1	0 yoki 1		1 OK
int read_charclass_c	(it&, bspn, ch*)	[ ]	-1	0 yoki 1		OK
int read_c	(it&, ch*)	.	-1	0	5	OK
int read_while_charclass	(it&, is)	[ ]*	-(1+len)	len		



OK				
int read_while_charclass	(it&, spn)	[ ]*	-(1+len)	len
2 OK				
int read_while_charclass	(it&, bspn)	[ ]*	-(1+len)	len
OK				
int read_while_charclass	(it&, is, pstr*)	[ ]*	-(1+len)	len
OK				
int read_while_charclass	(it&, spn, pstr*)	[ ]*	-(1+len)	len
OK				
int read_while_charclass	(it&, bspn, pstr*)	[ ]*	-(1+len)	len
1 OK				
int read_until_charclass	(it&, is)	.*[ ]<-	-(1+len)	len
OK				
int read_until_charclass	(it&, spn)	.*[ ]<-	-(1+len)	len
1 OK				
int read_until_charclass	(it&, bspn)	.*[ ]<-	-(1+len)	len
OK				
int read_until_charclass	(it&, is, pstr*)	.*[ ]<-	-(1+len)	len
OK				
int read_until_charclass	(it&, spn, pstr*)	.*[ ]<-	-(1+len)	len
2 OK				
int read_until_charclass	(it&, bspn, pstr*)	.*[ ]<-	-(1+len)	len
OK				
int read_until_char	(it&, c)	.*c	-(1+len)	len
OK				
int read_until_char	(it&, c, pstr*)	.*c	-(1+len)	len
OK				

Oxirgibelginio‘qigandanso‘ng, iteratorundankeyinemas, balkiunga qaratilgan.

Funksiyanomidan foydalanib, xatokodini yokixat oxabarini qaytarish qulay, shuning uchun parserlarni matnidan aniqlik qo‘lga kiritishni, maxsus makroslarni keltiramiz:

```
#define r_if(expr)          if((expr)==0)
#define r_while(expr)      while((expr)==0)
#define r_ifnot(expr)      if(expr)
#define r_whilenot(expr)   while(expr)
```

```

#define rm_if(expr)          if((expr)>=0)
#define rm_while(expr)     while((expr)>=0)
#define rm_ifnot(expr)     if((expr)<0)
#define rm_whilenot(expr)  while((expr)<0)
#define rp_if(expr)        if((expr)>0)
#define rp_while(expr)     while((expr)>0)
#define rp_ifnot(expr)     if((expr)<=0)
#define rp_whilenot(expr)  while((expr)<=0)

```

Buma'lumotlarnisaqlashva bool tipigamosakslantirishbo'lsadaba'zisinfiqilishuchunjudamuhim.

Umuman, o'zgarmasbo'lganhavolalargaqarshiman, funksiyasio'zgartirishkerakargumentlarko'rsatgichlaribilanemas, balkibirko'rsatkichorqaliungao'tganbo'lishikerak

Ushbufunksiyalardanfoydalanishningikkitamisolkeltiramiz:

7.2-dastur. Arifmetikifodalarnihisoblash.

```

#define ONE_SOURCE //strin.cpp
#include "../strin.h"

using namespace str;
using std::cout;
using std::endl;

#define ifnot(expr)          if(!(expr))
#define r_if(expr)          if((expr)==0)
#define r_while(expr)       while((expr)==0)
#define r_ifnot(expr)       if(expr)
#define r_whilenot(expr)    while(expr)

template <class it_t, class mes_t>
void dump(it_t tmp, mes_t mes){
    string s;
    read_fix_length(tmp,50,&s);
    std::cerr <<mes <<dump(s.c_str()) <<endl;
}

template<class it_t>
const char * read_sum(it_t & it, double * prez);

```

```

template<class it_t>
const char * read_expr(it_t & it, double * prez){
    const char * err;
    if(read_s_fix_char(it, '(')){
        r_ifnot(err = read_sum(it,prez))
            return err;
        ifnot(read_s_fix_char(it, ')'))
            return "qavsni yoping";
        return 0;
    }
    int x;
    ifnot(read_dec(it,&x)){
        return "son";
    }
    *prez = x;
    return 0;
}

```

```

DEF_STRING(md,"*/")
template<class it_t>
const char * read_mul(it_t & it, double * prez){
    typedef char_type<it_t> ch_t;
    const char * err;
    r_ifnot(err = read_expr(it,prez))
        return err;
    while(true){
        ch_t zn;

        ifnot(read_s_charclass_c(it,make_span(md<ch_t>().
s),&zn))
            return 0;
        double x;
        r_ifnot(err=read_expr(it,&x))
            return err;
        if(zn==(ch_t) '*')    *prez *= x;
        else                  *prez /= x;
    }
}

```

```

    }
    return 0;
}

DEF_STRING(pm,"+-")
template<class it_t>
const char * read_sum(it_t & it, double * prez){
    typedef char_type<it_t> ch_t;
    const char * err;
    r_ifnot(err = read_mul(it,prez))
        return err;
    while(true){
        ch_t zn;

        ifnot(read_s_charclass_c(it,make_span(pm<ch_t>().
s),&zn))
            return 0;
        double x;
        r_ifnot(err = read_mul(it,&x))
            return err;
        if(zn==(ch_t)'+')    *prez += x;
        else                *prez -= x;
    }
    return 0;
}

int main()
{
    using namespace std;
    if(0)
    {
        string str_expression_1 = "5+84/(51)";
        string str_expression_2 = "15 + (16 /6 *4)";
        string str_expression_3 = "7 + ( 4* 4)";
        string str_expression = str_expression_3;
        cout <<"hisoblash ifodasi:" <<endl
<<str_expression <<endl;
        const char * p= str_expression.c_str();

```

```

const char * err;
double rez;
r_ifnot(err=read_sum(p,&rez)){
    int pos=p-str_expression.c_str();
    for(int i=1; i<pos; i++)
        cout <<' ';
    cout <<'^' <<endl;
    cout <<"Xatolik: " <<err <<endl;
    return -1;
}
cout << "natija: " << rez << endl;
cout <<"=====" << endl;
}

if(1)
{
    try{
        while(!atend(strin)){
            cout <<"arifmetik ifoda kiriting va
'end' bilan tugating " <<endl;
            read_start_line(strin);
            const char * err;
            double rez=0;
            err=read_sum(strin,&rez);
            linecol lc=get_linecol(strin);
            read_until_str(strin,"end");
            r_if(err)
                cout << "natija: " << rez <<
endl;
            else{
                cout <<" zanjir " <<lc <<"
xatolik: " <<err <<endl;
            }
        }
    }
    catch(const char * mes){
        cerr << "xatolik: " << mes << endl;
        return -1;
    }
}

```

```

    }
    catch(string & mes){
        cerr << "xatolik: " << mes << endl;
        return -1;
    }
    catch(stream_exception & mes){
        cerr << "xatolik: " << mes.what() <<
endl;
        return -1;
    }
    catch(...){
        cerr << "noaniq [atolik" << endl;
    }
}

return 0;
}

```

## 7.2-dastur. Output

```

--> runcpp -f calc.cpp
option -f
runcpp: g++ calc.cpp (options = -std=gnu++11 -Wall
-Wno-parentheses)
runcpp: compile messages lines 0
arifmetik ifoda kiriting va 'end' bilan tugating
1
end
natija: 1
arifmetik ifoda kiriting va 'end' bilan tugating
1+1
end
natija: 2
arifmetik ifoda kiriting va 'end' bilan tugating
1+1+1+1+7*7*(3)
end
zanjir 5:1 xatolik: son
arifmetik ifoda kiriting va 'end' bilan tugating

```

```

1end
natija: 1
arifmetik ifoda kiriting va 'end' bilan tugating
7*(3)
end
zanjir 8:1 xatolik: son

arifmetik ifoda kiriting va 'end' bilan tugating
1+(2)
end
natija: 3
arifmetik ifoda kiriting va 'end' bilan tugating
1+e end
zanjir 12:3 xatolik: son
arifmetik ifoda kiriting va 'end' bilan tugating
(7)*3end
natija: 7
arifmetik ifoda kiriting va 'end' bilan tugating
7*(3)end
zanjir 2:1 xatolik: son

arifmetik ifoda kiriting va 'end' bilan tugating
(5)+(5)end
natija: 10

```

Bugungi kundasturchilar uchun maxsus sintaktik tahlillovchilari yaratilishi ko'phollardashartemas. Chunki bumashaqqatli vako'pvaqttalabqiladi. Umuman olganda qanday dirdasturiy ta'minotlarni, axborot tizimlarining lingvistik ta'minoti yaratilsa, shu ta'minot uchun analizatori yaratish kerak bo'lishi mumkin. Dunyodasturlashtillarini ishlab chiqaruvchilar busohadajuda oldilab ketgan. Shuning uchun sintaktik tahlillovchilar bo'yicha nazariy tushunchaga egabo'lishi yetarlidi. Ammo ayrim matematik yangi sonli inflar uchun ishlab chiqish mumkin.

## NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1. Dasturlashtilining sintaktik konstruksiyalarini tasvirlash uchun eng qulay formal usul qaysi usul?
2. Dasturlarning sintaktik to'g'riligini aniqlashning yanabir muhim qismi - dastur datiplardan foydalanishning to'g'riligini qaysi grammatika yordamida aniqlab bo'lmaydi?
3. Sintaktik tahlil degandanimatishuniladi?
4. n uzunlikdagikiruvchisatruchunmurakkabligiqanchadanoshmasligikerak bo'lgan har qanday kontekst-erkin grammatika uchun analizator (parser) qurish mumkin?
5. Sintaktik tahlillovchikirish parametrlarini madani borat?
6. Sintaktik tahlillovchichiqish parametrlarini maldan borat?
7. Leksik va sintaktik tahlil bosqichlarining alohida qarashlarga ajratilishi shartmi?
8. Sintaktik tahlillovchisi nflarini ehtavo qaysi nfi algoritmlariga taluqli?
9. Pastsathlarga yo'naltirilgan analizatorlar chiqishni qurish uchun grammatika nima dan boshlab vanimabilantugaydi.
10. LL grammatika bo'lmagan grammatikalarda analizatorlar uchun qaysi usullarda nfoydalanish mumkin.
11. Pastsathlarga yo'naltirilgan analizatorlarga qaraganda yuqorisathlarga yo'naltirilgan analizatorlar qanchalik ishlatiladi.
12. Yuqorisathlarga yo'naltirilgan analizatorlar bilan qaysi grammatika bilan bog'liq.
13. Bugungi kunda qaysi grammatikaga asoslangan analizatorlar bilan juda ko'p dasturlashtillar foydalanmoqda.
14. Eng oddiy vujudako'p foydalanilgan pastsathlarga yo'naltirilgan analizatorlarni qurish usulini ayting?
15. Ifodalarni hisoblashdabarcha hisoblashlarni qandaysinflarga ajratish mumkin.
16. Chap operandni tanlashimiz vakerakli amalni bajarish uchun soniyoki oddiy ifoda dan borat'ng operandni aniqlashimiz lozim, so'ngesani manibajarish mumkin.
17. Qo'shishtipiga oid amallarni hisoblash uchun ifoda ni hisoblashning umumiy formulasi qanday?
18. Qiymat qaytarma slika sosidagi rekursiv kamayish usulidan faqat qaysi hart bajarilganda foydalanish mumkin.
19. KS grammatika uchun terminal va terminal bo'lmagan belgilardan borat G



- grammatikava  $w$  zanjirasosida  $FIRST_k(w)$  to‘plamniqandayaniqlaymiz?
20.  $FIRST_k(w)$  to‘plam  $w$  danolinganterminalzanjirlarninguzunligi  $k$  bo‘lganbarchaterminalnimalaridaniborat.
  21. Agar  $w$  zanjirfaqatterminallardaniboratbo‘lsa,  $FIRST_k(w)$  -  $w$  zanjirdabirinchi  $k$  belgilardir, aksholda  $|w| \geq k$ , yokiagar  $|w| < k$  bo‘lsa, nimabo‘ladi.
  22. Agarxteminalemasvagrammatikaqoidasix  $\rightarrow y_1 y_2 \dots y_k$ bo‘lsa,  $FIRST(X)$ gaqandayterminalniqo‘shamiz.
  23. Agar  $FIRST_k(x) = FIRST_k(y)$  teng, .... o‘rinli bo‘lsa  $G = (V_T, V_N, P, S)$  grammatikaLL(k)-grammatikadebaytiladi.
  24. LL(k) – xususiyalarigrammatikauchunqandaycheklovlaryuklaydi.
  25. Agargrammatikada  $A \Rightarrow^* Aw$  chiqishmavjudbo‘lsa, Aterminalbo‘lmagan KS grammatikanimadebaytiladi.
  26. Qiymatqaytarishasosidagirekursivkamayishusuliniqo‘llashuchungrammatikani FIRST to‘plamlarqandayshaklgaaylantirishkerak.
  27. Leksikanalizatoran‘naviyusullarnito‘laligicha, scan va next ana‘naviyusullarvanusxalashkonstruktorigaegabo‘lgannimasifatidanamoyonbo‘ladi?
  28. Mashinatilidagimatnnimalardaniborat?
  29. Harbirterminalbo‘lmaganilningma’lumbirjumlasigamoskeladiganqaysishaklyordamidatasvirlashqulay?
  30. Qaysiko‘rinishida $expr ::= expr1|expr2|expr3$  funksiyaniyozamiz?
  31. Forward iteratorlarnimauunkerak?
  32. Input iteratorlarnimauchunkerak?
  33. Input iteratorlardanfoydalanishnitushuntiribbering?
  34. Forward iteratorlardanfoydalanishnitushuntiribbering?
  35. Forward iteratorlarning input iteratrlardanfarqinimada?



**AMALIY KO‘NIKMA VA MALAKALARNI ANIQLASH  
HAMDA RIVOJLANTIRISH UCHUN ASSISMENT  
TOPSHIRIQLARI.**

<b>BIRINChIASSISMENTTOPShIRIG‘I</b>	
🚩	<p>Sintaktikanalizatorniyaratishgaoidberilganquyida gidasturbo‘yichaberilgantopshiriqlarkeraklifragmentlariasosidabajaring.</p> <p>👉 Bundadasturdagiba‘zio‘zgartirishlarnitopishorqalitopshiriqlarbosqichma — bosqichamalganoshiriladi.</p>

dastur	topshiriqlar
<pre> // Created by MBBahodir class SimpleParser {     const int IF = 1;     const int THEN = 2;     const int ELSE = 3;     const int BEGIN = 4;     const int END =;     const int PRINT = 6;     const int SEMICOLON = 7;     const int NUM = 8;     const int EQ = 9;      public static void nextStep(int lc){         if (lexical_class == lc)          lexical_class = getLC();         else             error();     }      public static void S(void){          switch(getLC())         {             case IF: </pre>	<p>1. Sinfxatosinitoping.</p> <hr/> <hr/>
	<p>2. Sinfdanob'ektolibdasturdafoydalaning.</p> <hr/> <hr/> <hr/> <hr/>
	<p>3. Faqatqo'shishamalinitahlilqiluvchivaifodanihisoblovchidasturtuzing.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
	<p>4. Faqatayrishamalinitahlilqiluvchivaifodanihisoblovchidasturtuzing.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>

```
        E());
nextStep(THEN); S();
nextStep(ELSE); S();
break;
        case BEGIN:
            S(); L();
break;
        case PRINT:
            E();
break;
        default:
            error();
break;
    }
}

    public static
void L(void){
        switch
        (lexical_class)
        {
            case END:
                getLC();
break;
            case
SEMICOLON:
                getLC();
S(); L(); break;
            default:
                error();
break;
        }
}

    public static
void E(void){
        nextStep(NUM);
```

5. Qo‘shishvaayrishamallarinitahlilqiluvchivaifodanihisoblovchidasturtuzing.

---

---

---

---

---

---

6. Ko‘paytirishamalnitahlilqiluvchivaifodanihisoblovchidasturtuzing.

---

---

---

---

---

---

<pre> nextStep(EQ); nextStep(NUM);     }      public static void main(void){         lexical_class = getLC();         S();     } } </pre>	
<p>7. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____</p> <p>8. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.</p>	

## 2.4. Murakkabsaralashalgoritmlari. Amaliy dasturlash

📖 Murakkab saralashalgoritmlarini, ularningmurakkabligi, rekursivfoydalanishvatakomillashtirish, ishlatishg‘oyalarivasaralashalgoritmlarining 16 tavariantiningturlidasturfragmentlari, saralashalgoritmlarnitaqqoslash, turlisondagimassivlar, turliqiymatlarda o‘zlarinitutishinisolishtirishnatijalari, kattasonlarbilashishlash, mashinaxatoliginianiqlash, foydalanuvchiningarifmetikamallarnibajaruvchifunksiyalarniyaratishbo‘yichag‘oyalar, kattasonlisinflarniishlatish, mavjudkattasonlarbilanishlovchilar, kattasonlaribilashishlovchikutubxonalariniyaratishvanamunalarikeltirilganbo‘lib, nazariybilimlarniasoslashuchundasturlartuzibko‘rsatilgan. bilimlarnimustahkamlashuchun 35 tanazariysavolvaamaliyko‘nikmavamalakalarnirivojlantrishuchun 4 taassimenttopshirig‘ivaharassimentda 29 tatopshiriqberilgan. Butopshiriqlarnibajarishmavzunimustahkamlashuchunxizmatqiladi.

✍️ **Kalitso‘zlar.** Saralash, massiv, tip, xotira, sinfkutubxona, massiv, satrlimassiv.

📌 **Bilishshartbo‘lgantushunchalar.** Tiptushunchasi, sinfvasinfob‘ekti, oqim, kirishelementgamurojaat, funksiyavako‘rsatkich, ma‘lumotlarnikirishvachiqishi, arifmetikamallar, kutubxonayaratishvadasturlashgaoiddastlabkitushunchalarhamdaC++tiliniqo‘llabquvvatlovchimuhitdaishlashnibilishlozim.

**Bilibolasiz.** Murakkabsaralashalgoritmlardan: Havosharchakabisaralash, Shakersaralash, Taroqsimon caralash, Qo'shishorqalisaralash, Shell caralash, Iearxiksaralash, Navsaralash, Tanlashorqalisaralash, Piramidakabisaralash, Tezkorsaralash, Birlashtirishorqalisaralash, Blokorqalisaralash, Bitlisaralash (LSD), Bitlisaralash (MSD), Bitoniksaralash, Timsortsaralash, bualgoritmlarnitaqqoslashvamurakkabligini, Tiplarvaularningxususiyatlari, Kattasonlarchegarasinitekshirish, Tiplargabog'liqkattasonlardanfoydalanishvaxatoligi, Kattasonlarningqo'shish, ayirish, ko'pytirishg'oyalarivaalgoritmlarinituzishni, Bignumber kutubxonasi, sinfivafunksiyalari, xususiyatlariyaratish, OpenSSL, Libgcrypt, GMP, Boost.Multiprecision, wide integer tiplari, BigIntkutubxonasiniyaratishuslubivabazaviyfunksiyalarinio'rganishingiz mumkin.

### **REJA**

1. Murakkabsaralashalgoritmlari.
2. Saralashalgoritmlaritaqqoslash.
3. Judahamkattasonlarbilanishlash.
4. Large sinfiningqo'llanilishi.
5. Integer sinfining qo'llanilishi

### **KIRISH**

Turlima'lumotlartuzilmalaridaniboratto'plamlarnisaralashalgoritmlarinibilasiz.

Ammoularniko'psonliqiyamatlarinio'zinitutishiqandaydegansavolqiziqtiradimi?

Nafaqatoddiysaralashbalkishundaymurakkabsaralashalgoritmlarihammavjud.

Ko'pelementlimassivlarkattasonlarbilashishlashgahamfoydalanishmumkin.

Shuningdektiylarningo'zaromodifikatsiyasibilanhamuzunsonlarbilanishlanadi.

**Saralashalgoritmlarivamurakkabligi.**Avvalo, algoritmlarnitadqiqqilishga, ularniimkonqadartezishlashiuchunoptimallashtirishmuhimdir. Buustidaishlayotganda, turlisaralashuchunsamaraliusullarnio'ylabtopishimkoniyatigaegabo'lishingizmumkin.

Ko'pjihatdan,  
 barchasaralashalgoritmlarnio'rganishvaularnisinashkerak.  
 Agardasturlashningo'zihaqidagapiradiganbo'lsak,  
 ba'zankutilmaganqiyinchiliklarpaydobo'lishimumkin (C++  
 optimizatorijudayaxshi). Biroq,  
 qaysitestlarnivaqandaymiqdordaamalgaoshirilishikerakliginihalqilishqiyi  
 nemas. Menko'rsataolmaydiganyagonanarsa-budeyarli 150  
 GBvazngaegaqiyamatlarnisaralashbo'ldi.

Asosiysaralashalgoritmlaritavsifivaularniamalgaoshirishusullari.  
 Saralashniqisqachavaaniqta'riflashgavamurakkabliginibelgilashgaharaka  
 tqilaman.

Murakkabma'lumotlartuzilmalarnifoydalanishda(daraxtsaralashkabi)  
 odatdaxotirakattamiqdordasarflanadivaengyomonholatdaboshqaxilfaqaty  
 ordamchiqatoryaratishkerakbo'ladi. Barqarorlik (stabillik)  
 saralashhunchasihammavjud. Demak,  
 elementlarningnisbiyartibitengbo'lgandao'zgarmaydi.

**Havosharchakabisaralash (Bubble sort).**Massivdachapdano'nggaqarabamalbajariladi.

Agarjoriyelementkeyingisidankattabo'lsa, ularnialmashtiramiz.  
 Bunimassivtartiblanmagunchabajaramiz. E'tiborbering,  
 birinchiiterasiyadankeyinengkattaelementmassivningoxirida,  
 to'g'rijoylashaganbo'ladi.

Ikkitaiterasiyadankeyinikkiinchiengekattaelementto'g'rijoylashaganbo'ladi  
 vahokazo. Ravshanki, n

taiterasiyadanko'pbo'lmagandankeyinmassivtartiblanganbo'ladi.  
 Shundayqilib, engyomonvao'rtachaholatdamurakkabligi $O(n^2)$ ,  
 engyaxshiholatdaesa $O(n)$  bo'ladi.

8.1-dastur. Saralashniamalgaoshirish.

```
void bubblesort(int* l, int* r) {
    int sz = r - l;
    if (sz <= 1) return;
    bool b = true;
    while (b) {
        b = false;
        for (int* i = l; i + 1 < r; i++) {
            if (*i > *(i + 1)) {
                swap(*i, *(i + 1));
                b = true;
            }
        }
    }
}
```

```

    }
    r--;
}
}

```

**Shakersaralash (Shaker sort, bundantashqari,**  
**aralashgansaralashvakoktelsaralashsifatidanilgan).**

Havosharchasaralashoxiridakichikelementlarjoylashtirilgantestlardaseki  
 nishlaydi (ba‘zan "tashboqa" debataladi).

Buelementalgoritmniningharbirqadamidachaptomongafaqatbittapozitsiyag  
 aharakatqiladi. Shuninguchunnafaqatchapdano‘ngga,

balkio‘ngdanchapghamo‘tadi.

Massivningtartiblanmaganqisminianiqlashuchunikkita begin va end  
 ko‘rsatikichniqo‘llabquvvatlaydi. Keyingiiterasiyada

endgaetganimizdaundanbiriniayiramizvao‘ngdanchapgako‘chamiz.

Xuddishunday,

beginngaetganimizdaungabiriniqo‘shamizvachapdano‘nggako‘chamiz .

Algoritmhavosharchasaralashbilanbirxilmurakkabligibor,

lekinrealvaqtdayaxshiishlaydi.

8.2-dastur. Saralashniamalgaoshirish.

```

void shakersort(int* l, int* r) {
    int sz = r - l;
    if (sz <= 1) return;
    bool b = true;
    int* beg = l - 1;
    int* end = r - 1;
    while (b) {
        b = false;
        beg++;
        for (int* i = beg; i < end; i++) {
            if (*i > *(i + 1)) {
                swap(*i, *(i + 1));
                b = true;
            }
        }
        if (!b) break;
        end--;
        for (int* i = end; i > beg; i--) {
            if (*i < *(i - 1)) {
                swap(*i, *(i - 1));
            }
        }
    }
}

```

```

        b = true;
    }
}
}
}

```

**Taroqsimon** **caralash(Comb sort)**. Shakersaralashning yanabi ro'zgartirilgani. "Toshbaqalar" dan qutulish uchun masofadaturganelementlarni qaytatash kilqilamiz. Keling, unituzatamiz vachapdano'nggaboramiz, bum asofadaturganelementlarni taqqoslaymiz, agarkera kbo'lsa, ularni qaytatartib gasolaylik. Shubhasiz, bu "toshbaqalar" nitezqator boshiga olishim koniniberadi. Dastlabqatoruzunligiga teng masofani olib, so'ngra unitaxminan 1.247 ga teng bo'lgan biror koefitsient ga bo'lish maqbuldir. Masofa bir biriga teng bo'lgan dasthakersaralash bajariladi. Eng yaxshisi, murakkabligi  $O(n \log n)$ , eng yomon,  $uO(n^2)$ . O'rtacha murakkabliginima uchun juda aniqlamas, amalda  $O(n \log n)$  kabiko'rinadi.

### 8.3-dastur. Saralashni amalga oshirish.

```

void combsort(int* l, int* r) {
    int sz = r - l;
    if (sz <= 1) return;
    double k = 1.2473309;
    int step = sz - 1;
    while (step > 1) {
        for (int* i = l; i + step < r; i++) {
            if (*i > *(i + step))
                swap(*i, *(i + step));
        }
        step /= k;
    }
    bool b = true;
    while (b) {
        b = false;
        for (int* i = l; i + 1 < r; i++) {
            if (*i > *(i + 1)) {
                swap(*i, *(i + 1));
                b = true;
            }
        }
    }
}

```



```

    }
}
}

```

**Qo‘shishorqalisaralash (Insertion sort).** Algoritmyakunlangandankeyinjavobnio‘zichigaolganbiryangimassiv yaratiladi.

Javobmassividageelementlarhardoim tartiblanib turishi uchun manba massiv idanelementlarni birma-bir kiritamiz.

Murakkabligi o‘rtacha va eng yomon holat  $O(n^2)$  va eng yaxshi  $O(n)$  bo‘ladi.

Boshqa algoritmlar nisbatan joriy qilish yaxshiroq (yangi massiv yaratish va albatta unimanidir qo‘yish qiyin):

joriy massivning ba‘zi bosh prefikslarini tartiblashtirilganiga ishonch hosil qilish va modomiki ular noto‘g‘ri tartibda ekan uni kiritish o‘rniga joriy elementni avvalgisibilano‘zgartiramiz.

8.4-dastur. Saralashni amalga oshirish.

```

void insertionsort(int* l, int* r) {
    for (int *i = l + 1; i < r; i++) {
        int* j = i;
        while (j > l && *(j - 1) > *j) {
            swap(*(j - 1), *j);
            j--;
        }
    }
}

```

**Shell**

**caralash**

**(Shellsort).** Saralashdabir xil fikrdan foydalanishni o‘ylab ko‘ring, taroqsimonsaralash va qo‘shishorqalisaralash uchun uning o‘llaymiz.

Masofatlash uchun kiritaylik vatanlaylik.

So‘ngramassivelementlaris inflarga ajratiladi,

shuningga o‘zgar masofaning bir necha elementlarini kiradi.

Har birisinfni qo‘shib, tartiblash yo‘libilantartiblaylik.

Taroqsimonsaralashdan farqli o‘laroq, optimal masofalarni o‘plamiani qimas.

Turlitaxminlarga ega bo‘lgan bir necha ketma-ketliklar mavjud bo‘lsin.

Shell saralashdabirinchielementketma-ketligi massiv uzunligiga teng,

har bir keyingi element ketma-ketligi oldingisining yarim o‘lchamiga tengdir.

Hibbard ketma-ketligi  $2n-1$  uchun murakkabligi eng yomon holat  $O(n^2)$  dir,

Sedgwick ketma-ketligi  $O(n^{4/3})$  uchun (ifodao‘zgaruvchan xususiyatli) va

Pratt (ikkivauchdarajali elementlar) ketma-ketligi  $O(n \log 2^n)$

murakkabligi eng yomon holat  $O(n^{1.5})$  dir.

Bubarchaketmaketliklarfaqatmassivhajmigaqadabhisoblabvakichikuchun ko'proqishlatishkerak

(aksholdaufaqtqo'shimchalaribilantartiblashtiriladibo'ladi).

Qo'shimchatadqiqotqildimva  $s_i = a * s_{i-1} + k * s_{i-1}$  shaklidaturliketmaketliklar (buqismanempirik siurketma - ketlikdailhomlanib,

elementlarkichiksoniuchunengyaxshimasofaketmaketliklardanbiri)

sinovdano'tkazdim. Engyaxshiketmaketliklara = 3, k = 1/3, a = 4, k = 1/4 vaa = 5, k = -1/5 koeffitsentlaribilanmaqsadgaerishishmumkin.

8.5-dastur. Saralashniamalgaoshirish.

1- variant

```
void shellsort(int* l, int* r) {
    int sz = r - l;
    int step = sz / 2;
    while (step >= 1) {
        for (int *i = l + step; i < r; i++) {
            int *j = i;
            int *diff = j - step;
            while (diff >= l && *diff > *j) {
                swap(*diff, *j);
                j = diff;
                diff = j - step;
            }
        }
        step /= 2;
    }
}
```

2- variant

```
void shellsorthib(int* l, int* r) {
    int sz = r - l;
    if (sz <= 1) return;
    int step = 1;
    while (step < sz) step <<= 1;
    step >>= 1;
    step--;
    while (step >= 1) {
        for (int *i = l + step; i < r; i++) {
            int *j = i;
```

```

        int *diff = j - step;
        while (diff >= 1 && *diff > *j) {
            swap(*diff, *j);
            j = diff;
            diff = j - step;
        }
    }
    step /= 2;
}
}

```

3- variant

```

int steps[100];
void shellsortsedgwick(int* l, int* r) {
    int sz = r - l;
    steps[0] = 1;
    int q = 1;
    while (steps[q - 1] * 3 < sz) {
        if (q % 2 == 0)
            steps[q] = 9 * (1 << q) - 9 * (1 << (q
/ 2)) + 1;
        else
            steps[q] = 8 * (1 << q) - 6 * (1 << ((q
+ 1) / 2)) + 1;
        q++;
    }
    q--;
    for (; q >= 0; q--) {
        int step = steps[q];
        for (int *i = l + step; i < r; i++) {
            int *j = i;
            int *diff = j - step;
            while (diff >= 1 && *diff > *j) {
                swap(*diff, *j);
                j = diff;
                diff = j - step;
            }
        }
    }
}

```

```
}
```

#### 4- variant

```
void shellsortpratt(int* l, int* r) {
    int sz = r - l;
    steps[0] = 1;
    int cur = 1, q = 1;
    for (int i = 1; i < sz; i++) {
        int cur = 1 << i;
        if (cur > sz / 2) break;
        for (int j = 1; j < sz; j++) {
            cur *= 3;
            if (cur > sz / 2) break;
            steps[q++] = cur;
        }
    }
    insertionsort(steps, steps + q);
    q--;
    for (; q >= 0; q--) {
        int step = steps[q];
        for (int *i = l + step; i < r; i++) {
            int *j = i;
            int *diff = j - step;
            while (diff >= l && *diff > *j) {
                swap(*diff, *j);
                j = diff;
                diff = j - step;
            }
        }
    }
}
```

#### 5- variant

```
void myshell1(int* l, int* r) {
    int sz = r - l, q = 1;
    steps[0] = 1;
    while (steps[q - 1] < sz) {
        int s = steps[q - 1];
        steps[q++] = s * 4 + s / 4;
    }
}
```

```

q--;
for (; q >= 0; q--) {
    int step = steps[q];
    for (int *i = l + step; i < r; i++) {
        int *j = i;
        int *diff = j - step;
        while (diff >= l && *diff > *j) {
            swap(*diff, *j);
            j = diff;
            diff = j - step;
        }
    }
}

```

6- variant

```

void myshell2(int* l, int* r) {
    int sz = r - l, q = 1;
    steps[0] = 1;
    while (steps[q - 1] < sz) {
        int s = steps[q - 1];
        steps[q++] = s * 3 + s / 3;
    }
    q--;
    for (; q >= 0; q--) {
        int step = steps[q];
        for (int *i = l + step; i < r; i++) {
            int *j = i;
            int *diff = j - step;
            while (diff >= l && *diff > *j) {
                swap(*diff, *j);
                j = diff;
                diff = j - step;
            }
        }
    }
}

```

7- variant

```

void myshell3(int* l, int* r) {

```

```

int sz = r - 1, q = 1;
steps[0] = 1;
while (steps[q - 1] < sz) {
    int s = steps[q - 1];
    steps[q++] = s * 4 - s / 5;
}
q--;
for (; q >= 0; q--) {
    int step = steps[q];
    for (int *i = l + step; i < r; i++) {
        int *j = i;
        int *diff = j - step;
        while (diff >= l && *diff > *j) {
            swap(*diff, *j);
            j = diff;
            diff = j - step;
        }
    }
}
}
}

```

### Iearxiksaralash

(Tree

sort). Binarqidiruviearxiyasigaelementlarnikiritamiz.

Barchaelementlarkiritilgandanso'ng,

iearxiyadantartiblanganmassivolishkifoya.

Agarsizqizil-

qorakabimuvozanatliiearxiyadanfoydalansangiz,

murakkabligiengyomon,

o'rtachavaengyaxshiholatlarda  $O(n \log n)$ .

Amalgaoshirishda multiset konteynerfoydalanadi.

8.6-dastur. Saralashniamalgaoshirish.

```

void treesort(int* l, int* r) {
    multiset<int> m;
    for (int *i = l; i < r; i++)
        m.insert(*i);
    for (int q : m)
        *l = q, l++;
}

```

### Navsaralash

(Gnome

sort). Algoritm qo'shishorqalisaralashgao'xshash.

Ko'rsatgichnijoriyelementgako'rsatamiz,

agaruavvalgisidankattabo'lsayokiubirinchiabo'lsako'rsatgichnito'g'rihola  
tgako'chiramiz,  
aksholdajoriyvaoldingielementlarnio'zgartiramizvachaptomongaharakatl  
antiramiz.

8.7-dastur. Saralashniamalgaoshirish.

```
void gnomesort(int* l, int* r) {  
    int *i = l;  
    while (i < r) {  
        if (i == l || *(i - 1) <= *i) i++;  
        else swap(*(i - 1), *i), i--;  
    }  
}
```

**Tanlashorqalisaralash**

**(Selection**

**sort).**Navbatdagiiterasiyabo'yichamassivdagiminimumnijoriyelementda  
nkeyintopamizvakerakbo'lsaubilano'zgartiramiz. Shundayqilib, i-  
iterasiyadankeyinbirinchi i elementlaroz'joylaridaqoladi.  
murakkabligiengyaxshi, o'rtachavaengyomonholatlarda $O(n^2)$ .  
Busaralashniikkiyo'lbilanamalgaoshirilishimumkinliginiunutmang  
-  
minimalvauningindekstutib, yokishunchakiularnoto'g'ritartibdabo'lsa,  
ko'ribchiqilayotganvajoriyelemento'rinalmashtirishkerak.  
Birinchiusulbiroztezroq.

8.8-dastur. Saralashniamalgaoshirish.

```
void selectionsort(int* l, int* r) {  
    for (int *i = l; i < r; i++) {  
        int minz = *i, *ind = i;  
        for (int *j = i + 1; j < r; j++) {  
            if (*j < minz) minz = *j, ind = j;  
        }  
        swap(*i, *ind);  
    }  
}
```

**Piramidakabisaralash**

**(Heapsort).**Tanlashorqalisaralashg'oyasinirivojlantirilganvarianti.

Piramidashaklidagimalumotstrukturasidanfoydalanaylik.

Elementlarniqo'shishvaminimuminichiqarishorqali $O(\log n)$ ,  $O(1)$   
minimuminiolishimkoniniberadi. Shundayqilib,  $O(n \log n)$   
murakkabligiengyomon, o'rtachavaengyaxshiholatlarda. C++ da  
priority\_queue konteynerimavjudbo'lsa-da,

bukonteynerjudasekinbo'lganiuchunPiramidagayo'natirilganyangito'plamsinfniamalgaoshirdim.

8.9-dastur. Saralashniamalgaoshirish.

Heap sinfi

```
template <class T>
class heap {
private :
    vector<T> h;
    void SiftUp(int a) {
        while (a) {
            int p = (a - 1) / 2;
            if (h[p] > h[a]) swap(h[p], h[a]);
            else break;
            a--; a /= 2;
        }
    }
    void SiftDown(int a) {
        while (2 * a + 1 < n) {
            int l = 2 * a + 1, r = 2 * a + 2;
            if (r == n) {
                if (h[l] < h[a]) swap(h[l], h[a]);
                break;
            }
            else if (h[l] <= h[r]) {
                if (h[l] < h[a]) {
                    swap(h[l], h[a]);
                    a = l;
                }
                else break;
            }
            else if (h[r] < h[a]) {
                swap(h[r], h[a]);
                a = r;
            }
            else break;
        }
    }
public:
```



```

heap():n(0){}
int n;
int size() {
    return n;
}
int top() {
    return h[0];
}
bool empty() {
    return n == 0;
}
void push(T a) {
    h.push_back(a);
    SiftUp(n);
    n++;
}
void pop() {
    n--;
    swap(h[n], h[0]);
    h.pop_back();
    SiftDown(0);
}
void clear() {
    h.clear();
    n = 0;
}
T operator [] (int a) {
    return h[a];
}
};

```

Heap cinfiaosidasaralash

```

void heapsort(int* l, int* r) {
    heap<int> h;

    for (int *i = l; i < r; i++) h.push(*i);
    for (int *i = l; i < r; i++) {
        *i = h.top();
    }
}

```

```

        h.pop();
    }
}

```

**Tezkorsaralash (quicksort).** Baʼzimosyozuvlarelementinitanlaylik. Shundansoʻngchapdankichikboʻlganbarchaelementlarni, kattaboʻlganlariniesaoʻngtomongaharakatlantiramiz. Takrorlanuvchiqismlarningharbiridanrekursivchaqirishmumkin. Natijadatartiblanganmassivgaegaboʻlamiz, chunkiharbirkattaelementdanoldinmassivdankichikboʻlganharbirelement joylashtirilgan. Murakkabligi:  $O(n \log n)$  oʻrtachavaengyaxshiholda,  $O(n^2)$  engyomonholda. Malumotelementinotoʻgʻritanlangandaengyomonreytinggaerishiladi. Bualgoritmniamalgaoshirishbutunlaystandarthisoblanadi, birvaqtningoʻzidachapgavaoʻnggaborib, elementlarnijufttopish, bundachapelementmosyozuvlaritanlangandankattaekanliginivaoʻngkichik, ularalmashtiriladi. Softezkortartiblashdantashqari, elementlarsonikamboʻlgandatartiblashnijoylashtirishga, taqqoslashdahamsaralashishtiroketdi. Qoʻshishorqalisaralashvazifauchunmosboʻlganengyaxshisaralashdirvado imiysinovorqalitanlanadi.

#### 8.10-dastur. Saralashniamalgaoshirish.

##### Birinchi variant

```

void quicksort(int* l, int* r) {
    if (r - l <= 1) return;
    int z = *(l + (r - l) / 2);
    int* ll = l, *rr = r - 1;
    while (ll <= rr) {
        while (*ll < z) ll++;
        while (*rr > z) rr--;
        if (ll <= rr) {
            swap(*ll, *rr);
            ll++;
            rr--;
        }
    }
    if (l < rr) quicksort(l, rr + 1);
    if (ll < r) quicksort(ll, r);
}

```

##### Ikkinchi variant

```

void quickinssort(int* l, int* r) {
    if (r - l <= 32) {
        insertionsort(l, r);
        return;
    }
    int z = *(l + (r - l) / 2);
    int* ll = l, *rr = r - 1;
    while (ll <= rr) {
        while (*ll < z) ll++;
        while (*rr > z) rr--;
        if (ll <= rr) {
            swap(*ll, *rr);
            ll++;
            rr--;
        }
    }
    if (l < rr) quickinssort(l, rr + 1);
    if (ll < r) quickinssort(ll, r);
}

```

**Birlashtirish orqalisaralash (Merge sort).** Bo‘lish-boshqarish paradigmasi asosida tartiblash. Massivni ikkigabo‘lamiz, qismlarni rekursiv ravishda tartiblang va keyin birlashtirish protsedurasini bajarib aring. Birinchi qismning joriy elementiga, ikkinchi sikkini qismning joriy elementiga ikki takrorlikni qo‘llang. Bu ikki elementdan minimumni tanlab, javobga joylashtirish va minimumga mos keladigan ko‘rsatgichni siljiting. Birlashtirish  $O(n)$ , barcha  $\log n$  darajalari uchun ishlaydi, shuning uchun murakkabligi  $O(n \log n)$ . Bu oldindan vaqtinchalik masiv yaratish va vazifa argument sifatida o‘tishis amaralibo‘ladi. Busaralash rekursiv, hamdatez va shuning uchun elementlar soni kambo‘lgan kvadratiko‘tish mumkin.

8.11-dastur. Saralashni amalga oshirish.

Asosiy funksiya

```

void merge(int* l, int* m, int* r, int* temp) {
    int *cl = l, *cr = m, cur = 0;
    while (cl < m && cr < r) {
        if (*cl < *cr) temp[cur++] = *cl, cl++;
    }
}

```

```

        else temp[cur++] = *cr, cr++;
    }
    while (cl < m) temp[cur++] = *cl, cl++;
    while (cr < r) temp[cur++] = *cr, cr++;
    cur = 0;
    for (int* i = l; i < r; i++)
        *i = temp[cur++];
}

```

#### Birinchi varinat

```

void _mergesort(int* l, int* r, int* temp) {
    if (r - l <= 1) return;
    int *m = l + (r - l) / 2;
    _mergesort(l, m, temp);
    _mergesort(m, r, temp);
    merge(l, m, r, temp);
}

void mergesort(int* l, int* r) {
    int* temp = new int[r - l];
    _mergesort(l, r, temp);
    delete temp;
}

```

#### Ikkinchi varinat

```

void _mergeinsort(int* l, int* r, int* temp) {
    if (r - l <= 32) {
        insertionsort(l, r);
        return;
    }
    int *m = l + (r - l) / 2;
    _mergeinsort(l, m, temp);
    _mergeinsort(m, r, temp);
    merge(l, m, r, temp);
}

void mergeinsort(int* l, int* r) {
    int* temp = new int[r - l];
    _mergeinsort(l, r, temp);
    delete temp;
}

```

**Sanashorqalisaralash (Counting sort).**  $r$  -  $l$  hajlimassiv yaratilgan, massivning  $l$  - minimalva  $r$  maksimalelementi hisoblanadi. Shunday qilib, massiv orqali bir element sonini hisoblash. Endi qiymatlarni massivdan o'qib, har bir sonni  $n$  echa marta bo'lganini yozish giz mumkin. murakkabligi  $O(n + r)$  bo'ladi. Bualgoritmlar qaror qilish uchun o'zgartirish giz mumkin: buning uchun, keyingi soni joyini aniqlash kerak bo'lib, vachapdan o'ngga original massiv orqali borib, to'g'ri joyda elementni qo'yib, va o'rnini birga oshirish lozim. Busaralash sinovdan o'tkazilmagan, testlari yetarli darajada kattaraq amllar uchun mavjud, chunki yangi massiv yaratish ancha xotiratalab qiladi. Biroq, bualgoritmlar masalaga qarab foydalib o'linadi.

**Bloklar qalisaralash (Bucket sort).** Bualgoritmlar savatvach o'taksaralash sifatida tanilgan algoritmdir. Massivning  $l$  minimalva  $r$  maksimalelementi bo'lsin. Elementlarni bloklarga ajratamiz, birinchi blokda  $l$  dan  $l + k$  gacha, ikkinchisida  $l + k$  dan  $l + 2k$  gacha va hokazo elementlarni o'z ichiga oladi, buyruqda  $k = (r - l) / n$ ,  $n$  bloklar soni. Umuman olganda, agar bloklar soni ikkiga teng bo'lsa, bualgoritmlar tez korsaralashga ayylanadi. Ushbu bualgoritmlarning murakkabligi aniqlanmas, ma'lumotlarni gajamurojaat vaqt va bloklar soni gajabog'liq. Muvaffaqiyatli ma'lumotlar bo'yicha ish vaqt ichi ziqlik ekanligi aytilib o'tiladi. Ushbu bualgoritmlar niamalga oshirish gajayin vazifalardan biri ekanligini isbotlangan. Buning uchun hachki yangi massivlarni yaratish, ularni rekursiv saralash va ularni bir-biriga birlashtirish orqali amalga oshirish giz mumkin. Biroq, buyruqdashuv judasekin va men qoniqmadim. Samarali amalga oshirishda bir necha g'oyalardan foydalaniladi:

- 1) yangi massivlarni yaratmaymiz. Buning uchun sana o'rqalisaralash texnikasidan foydalanamiz: har bir blokda gajelementlarni, prefiks summalarni va shu tariqahar bir elementning massivdagio'rnini sanaymiz.
- 2) Bo'sh bloklardan boshlamaymiz. Bo'sh bo'lmagan bloklarning indekslarini alohida massivda kiritish va ularni boshlash orqali saralash.

3) Massiv tartiblanganligini tekshiramiz. Agar hal ham minimal va maksimal topish uchun biro' tish qilish kerak, chunki, bu ishlayotgan vaqti yomonlashmaydi, lekin elementlar original massiv bir xil tartibda yangi bloklari joylashtirilgan, chunki u qisimantartiblashtirilgan ma'lumotlarni saralashni tezlashtirish uchun algoritmi beradi.

4) Algoritmi judanoqulay bo'lgani uchun, u juda kam sonli elementlar bilan juda samarasiz. Joylashtirish gao'tis tartibida ishni taxminan 10 marta gachatezlashtiradi.

Bu faqat qancha bloklarni tushunishga qolmoqda. Randomize testlarda quyidagi ballarni olishga muvaffaq bo'lgan: 1500 elementlari uchun 107 bloklariva 3000 uchun 108ta. Murakkablik formulasini topolmadik, ish vaqti bir necha martayomonlashadi.

8.12-dastur. Saralashni amalga oshirish.

```
void _newbucketsort(int* l, int* r, int* temp) {
    if (r - l <= 64) {
        insertionsort(l, r);
        return;
    }
    int minz = *l, maxz = *l;
    bool is_sorted = true;
    for (int *i = l + 1; i < r; i++) {
        minz = min(minz, *i);
        maxz = max(maxz, *i);
        if (*i < *(i - 1)) is_sorted = false;
    }
    if (is_sorted) return;
    int diff = maxz - minz + 1;
    int numbuckets;
    if (r - l <= 1e7) numbuckets = 1500;
    else numbuckets = 3000;
    int range = (diff + numbuckets - 1) /
numbuckets;
    int* cnt = new int[numbuckets + 1];
    for (int i = 0; i <= numbuckets; i++)
        cnt[i] = 0;
    int cur = 0;
    for (int* i = l; i < r; i++) {
```

```

        temp[cur++] = *i;
        int ind = (*i - minz) / range;
        cnt[ind + 1]++;
    }
    int sz = 0;
    for (int i = 1; i <= numbuckets; i++)
        if (cnt[i]) sz++;
    int* run = new int[sz];
    cur = 0;
    for (int i = 1; i <= numbuckets; i++)
        if (cnt[i]) run[cur++] = i - 1;
    for (int i = 1; i <= numbuckets; i++)
        cnt[i] += cnt[i - 1];
    cur = 0;
    for (int *i = 1; i < r; i++) {
        int ind = (temp[cur] - minz) / range;
        *(1 + cnt[ind]) = temp[cur];
        cur++;
        cnt[ind]++;
    }
    for (int i = 0; i < sz; i++) {
        int r = run[i];
        if (r != 0) _newbucketsort(1 + cnt[r - 1],
1 + cnt[r], temp);
        else _newbucketsort(1, 1 + cnt[r], temp);
    }
    delete run;
    delete cnt;
}
void newbucketsort(int* l, int* r) {
    int *temp = new int[r - 1];
    _newbucketsort(l, r, temp);
    delete temp;
}

```

**Bitlisaralash(Radix sort).**  
 raqamlisaralashsifatidanilganalgoritmdir.  
 Busaralashningikkiversiyasimavjudbo‘lib,

Shuningdek,  
 ular,

qandaydir sonlarsistemasida (masalan, binar – ikkilik) sonniifodalashdan foydalanish fikridan boshqa umumiylikka ega emas.

Birinchi usuli. LSD (least significant digit, kichik muhim son). Har bir sonni binar shaklda ifodalaylik.

Algoritmning har bir qadamidasonlarnishunday tartiblaymizki, ular  $k$  doimiy bo‘lgan birinchi  $k$  \*  $i$  bitlar bo‘yicha tartiblan sin.

Buta‘rifd anshunday xulosakelib chiqadiki,

har bir qadamda elementlarni yangi  $k$  bitlar bo‘yicha uzluksiz tartiblash kifoyadir. Shumaqsadda,

sanashorqalisaralashideldir (agarmuvaffaqiyatlisinidoimiy tanlasangiz  $2k$  xotiravavaqt ko‘p kerak emas). Murakkabligi  $O(n)$ ,

agar sonlar fiksi rang katta likdada ebfaraz qilsak va aksholda ikki sonni taqqo slash vaqt birligibo‘yicha bajariladideb hisoblash mumkin bo‘lmaydi.

Amalga oshirish juda oddiy.

8.13-dastur. Saralashni amalga oshirish.

```
int digit(int n, int k, int N, int M) {
    return (n >> (N * k) & (M - 1));
}
void _radixsort(int* l, int* r, int N) {
    int k = (32 + N - 1) / N;
    int M = 1 << N;
    int sz = r - l;
    int* b = new int[sz];
    int* c = new int[M];
    for (int i = 0; i < k; i++) {
        for (int j = 0; j < M; j++)
            c[j] = 0;
        for (int* j = l; j < r; j++)
            c[digit(*j, i, N, M)]++;
        for (int j = 1; j < M; j++)
            c[j] += c[j - 1];
        for (int* j = r - 1; j >= l; j--)
            b[--c[digit(*j, i, N, M)]] = *j;
        int cur = 0;
        for (int* j = l; j < r; j++)
            *j = b[cur++];
    }
    delete b;
    delete c;
}
```



```

}
void radixsort(int* l, int* r) {
    _radixsort(l, r, 8);
}

```

Ikkinchisuli. MSD (most significant digit, engmuhimraqam). Aslidaqandaydirbloknisaralashalgoritmigao‘xshaydi. Shublokdateng k bitlisonlarbo‘ladi. Murakkabligi LSD versiyasidagidek. Amalgaoshirishbloksaralashjudao‘xshaydi, lekinoddiy. Bu LSD versiyasiniamalgaoshirishdagibelgilanganraqamli vazifasidanfoydalanadi

#### 8.14-dastur. Saralashniamalgaoshirish.

```

void _radixsortmsd(int* l, int* r, int N, int d,
int* temp) {
    if (d == -1) return;
    if (r - l <= 32) {
        insertionsort(l, r);
        return;
    }
    int M = 1 << N;
    int* cnt = new int[M + 1];
    for (int i = 0; i <= M; i++)
        cnt[i] = 0;
    int cur = 0;
    for (int* i = l; i < r; i++) {
        temp[cur++] = *i;
        cnt[digit(*i, d, N, M) + 1]++;
    }
    int sz = 0;
    for (int i = 1; i <= M; i++)
        if (cnt[i]) sz++;
    int* run = new int[sz];
    cur = 0;
    for (int i = 1; i <= M; i++)
        if (cnt[i]) run[cur++] = i - 1;
    for (int i = 1; i <= M; i++)
        cnt[i] += cnt[i - 1];
    cur = 0;
    for (int *i = l; i < r; i++) {

```

```

        int ind = digit(temp[cur], d, N, M);
        *(l + cnt[ind]) = temp[cur];
        cur++;
        cnt[ind]++;
    }
    for (int i = 0; i < sz; i++) {
        int r = run[i];
        if (r != 0) _radixsortmsd(l + cnt[r - 1], l
+ cnt[r], N, d - 1, temp);
        else _radixsortmsd(l, l + cnt[r], N, d - 1,
temp);
    }
    delete run;
    delete cnt;
}
void radixsortmsd(int* l, int* r) {
    int* temp = new int[r - l];
    _radixsortmsd(l, r, 8, 3, temp);
    delete temp;
}

```

### **Bitoniksaralash**

**(Bitonic**

**sort).**Massivelementlariniparalelsaralashalgoritmibo‘lib, tarmoqlardafoydalaniladi. Bualgoritmningg‘oyasishundaniboratki, massivBitonketma – ketlikka-avvalortib, keyinkamayibboradiganketma-ketlikkaaylanadi. Buni quyidagichasamaralitartiblashtirishmumkin: massivniikkiqismgaajratiladi, ikkitamassivyaratiladi, barchaelementlarniikkiqismdanharbirtegishlielementlarbirinchimassivga minimalvaikkinchisigamaksimaltengqo‘shiladi. IkkitaBitonketma-ketlikolinishi, ularningharbirinibirxilusuldarekursivravishdatartiblashmumkin, so‘ngraikkitamassivnibirlashtirishmumkin (birinchisiningharqaysielementiikkinchisiningharqaysielementidankamy okitengbo‘lganligiuchun). MassiviniBitonketma-ketlikkaaylantirishuchunquyidagilarnibajarishlozim: agarmassivikkielementdaniboratbo‘lsa, unishunchakibajarishmumkin, aksholdamassivniikkigaajratib, algorithmnibo‘lakmassivlardanrekursivchaqiriladi, keyinbirinchiqisminitartibbilan, ikkinchisiniteskaritartibdasaralab, bir-birigabirlashtiriladi. Shubhasiz, natijadabitonalketma-ketlikniolanadi.

Murakkabligi  $O(n \log^2 n)$ , chunki Biton ketma-ketligini qurishda  $O(n \log n)$  uchun ishlaydigan tartiblashlardan foydalandik va darajalarning umumiy soni  $i \log n$  edi. Bundantashqari, massiv hajmi kunning darajasiga karralibo'lishi kerak, shuningdek, elementlar bilano'ldirish kerak bo'lishi mumkin.

8.15-dastur. Saralashni amalga oshirish.

```

void bitseqsort(int* l, int* r, bool inv) {
    if (r - l <= 1) return;
    int *m = l + (r - l) / 2;
    for (int *i = l, *j = m; i < m && j < r; i++,
        j++) {
        if (inv ^ (*i > *j)) swap(*i, *j);
    }
    bitseqsort(l, m, inv);
    bitseqsort(m, r, inv);
}

void makebitonic(int* l, int* r) {
    if (r - l <= 1) return;
    int *m = l + (r - l) / 2;
    makebitonic(l, m);
    bitseqsort(l, m, 0);
    makebitonic(m, r);
    bitseqsort(m, r, 1);
}

void bitonicsort(int* l, int* r) {
    int n = 1;
    int inf = *max(l, r) + 1;
    while (n < r - l) n *= 2;
    int* a = new int[n];
    int cur = 0;
    for (int *i = l; i < r; i++)
        a[cur++] = *i;
    while (cur < n) a[cur++] = inf;
    makebitonic(a, a + n);
    bitseqsort(a, a + n, 0);
    cur = 0;
    for (int *i = l; i < r; i++)
        *i = a[cur++];
}

```

```
delete a;  
}
```

**Timsortsaralash (Timsort).** Bugibridsaralashhisoblanadi, qo‘shishorqalisaralashvabirlashtirishorqalisaralashalgoritmlaridanfoydalanib yaratilgan.

Massivelementlarinibirnechakichikmassivlargaajratamizvaundagielementlarnitartiblanganholdamassivostigakengaytiramiz.

Tartiblanganmassivlarustidasamaraliishlashidanfoydalanib, massivostilarnijoylashtiribtartiblashyo‘libilantartiblaymiz.

Keyinchalikmassivostilarnitaxminantengkattalikdaolib, birilashtirish (merge) saralashdagikabibirlashtiramiz (aksholdasaralashvaqtikvadratgayaqinlashadi).

Buninguchunmassivostilarniinvariantsaqlab, stekkasqlashqulay — tepadanuzoqlashgansarikattalashibboradivauchinchimassivostiningo‘lchamiularningkattaliklariyig‘indisidankattayokitengbo‘lgandaginayuqoriqismidamassivostilarnibirlashtiradi. murakkabligi $O(n)$

engyaxshiholatdava $O(n \log n)$  o‘rtachavaengyomonholatda.

Amalgaoshirisharzimasvaungaqaat‘iyishonchyo‘q, lekinujudayaxshisaralashvaqtiko‘rsatdi.

8.16-dastur. Saralashniamalgaoshirish.

```
void _timsort(int* l, int* r, int* temp) {  
    int sz = r - l;  
    if (sz <= 64) {  
        insertionsort(l, r);  
        return;  
    }  
    int minrun = sz, f = 0;  
    while (minrun >= 64) {  
        f |= minrun & 1;  
        minrun >>= 1;  
    }  
    minrun += f;  
    int* cur = l;  
    stack<pair<int, int*>> s;  
    while (cur < r) {  
        int* c1 = cur;  
        while (c1 < r - 1 && *c1 <= *(c1 + 1))  
            c1++;  
        int* c2 = cur;
```

```

while (c2 < r - 1 && *c2 >= *(c2 + 1))
c2++;
if (c1 >= c2) {
    c1 = max(c1, cur + minrun - 1);
    c1 = min(c1, r - 1);
    insertionsort(cur, c1 + 1);
    s.push(make_pair(c1 - cur + 1, cur));
    cur = c1 + 1;
}
else {
    c2 = max(c2, cur + minrun - 1);
    c2 = min(c2, r - 1);
    reverse(cur, c2 + 1);
    insertionsort(cur, c2 + 1);
    s.push(make_pair(c2 - cur + 1, cur));
    cur = c2 + 1;
}
while (s.size() >= 3) {
    pair<int, int*> x = s.top();
    s.pop();
    pair<int, int*> y = s.top();
    s.pop();
    pair<int, int*> z = s.top();
    s.pop();
    if (z.first >= x.first + y.first &&
y.first >= x.first) {
        s.push(z);
        s.push(y);
        s.push(x);
        break;
    }
    else if (z.first >= x.first + y.first)
{
        merge(y.second, x.second, x.second
+ x.first, temp);
        s.push(z);
        s.push(make_pair(x.first + y.first,
y.second));

```

```

        }
        else {
            merge(z.second, y.second, y.second
+ y.first, temp);
            s.push(make_pair( z.first +
y.first, z.second ));
            s.push(x);
        }
    }
    while (s.size() != 1) {
        pair<int, int*> x = s.top();
        s.pop();
        pair<int, int*> y = s.top();
        s.pop();
        if (x.second < y.second) swap(x, y);
        merge(y.second, x.second, x.second +
x.first, temp);
        s.push(make_pair( y.first + x.first,
y.second ));
    }
}

void timsort(int* l, int* r) {
    int* temp = new int[r - l];
    _timsort(l, r, temp);
    delete temp;
}

```

Yuqorida 16  
tasaralashalgoritmni hosil qilish va tayyodasturiy fragmentlar keltirildi.

**Saralashalgoritmni itaqqoslash.** Buning uchun barcha yaratilgan saralashalgoritmni bir faylga joylashtiramiz.

Bularni itaqqoslash uchun elementlar soni va bat bilan oshirib boramiz va natijalar sonli, grafik keltiramiz.

8.1-jadval. Saralashalgoritmni ro'yxati.

No	Saralashalgoritmning nomi	funksiyasi
1	Havosharchakabisaralash	void bubblesort(int* l, int* r)
2	Shakersaralash	void shakersort(int* l,

		int* r)
3	Taroqsimon caralash	void combsort(int* l, int* r)
4	Qo'shishorqalisaralash	void insertionsort(int* l, int* r)
5	Shell caralash	void shellsort(int* l, int* r) void shellsorthib(int* l, int* r) void shellsortpratt(int* l, int* r) void myshell_one(int* l, int* r) void myshell_two(int* l, int* r) void myshell_three(int* l, int* r)
6	Iearxiksaralash	void treesort(int* l, int* r)
7	Navsaralash	void gnomesort(int* l, int* r)
8	Tanlashorqalisaralash	void selectionsort(int* l, int* r)
9	Piramidakabisaralash	void heapsort(int* l, int* r)
10	Tezkorsaralash	void quicksort(int* l, int* r) void quickinsort(int* l, int* r)
11	Birlashtirishorqalisaralash	void mergesort(int* l, int* r) void mergeinsort(int* l, int* r)
12	Blokorqalisaralash	void newbucketsort(int* l, int* r)
13	Bitlisaralash (LSD)	void radixsort(int* l, int* r)
14	Bitlisaralash (MSD)	void radixsortmsd(int* l,

		int* r)
15	Bitoniksaralash	void bitonicsort(int* l, int* r)
16	Timsortsaralash	void timsort(int* l, int* r)

Bualgoritmlarnivaqtbo'yichataqqoslashuchuntasodifiysonlarbilanm  
assivnito'ldiramiz, tasodifiysonlarnihamguruhlargaaajratamiz,  
xonalarbo'yicha, massivelementlarsoniniham 100 tadan 100000  
tagachatanlabolamiz.

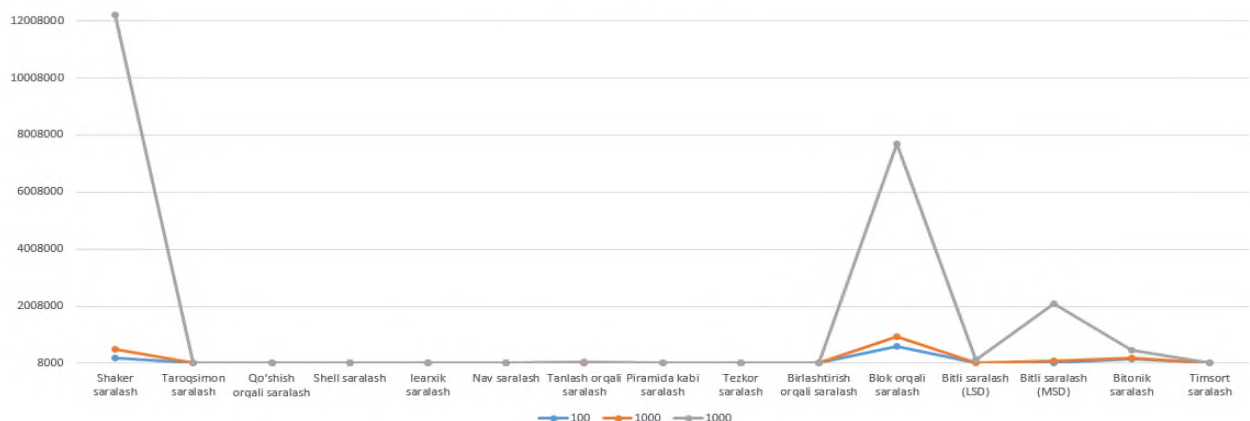
**Birinchitest.** Massivelementlarisoni 100, 1000, 10000  
tavatasodifiysonlarningenkattasi 1000:

Elementlar soni:100 Random eng katta soni:1000  bubblesort(a,b) => 189949 shakersort(a,b) =>9912 combsort(a,b) => 10017 insertionsort(a, b) => 9962 shellsort(a,b) => 9996 shellsorthib(a,b ) => 9983 shellsortpratt(a ,b) => 10001 myshell_one(a,b) => 10013 myshell_two(a,b) => 10000 myshell_three(a, b) => 9983 treesort(a,b) => 600071	Elementlar soni:1000 Random eng katta soni:1000  bubblesort(a,b) => 509917 shakersort(a,b) =>8523 combsort(a,b) => 9949 insertionsort(a, b) =>12012 shellsort(a,b) =>12536 shellsorthib(a,b ) => 9983 shellsortpratt(a ,b) => 10039 myshell_one(a,b) =>10026 myshell_two(a,b) =>9951 myshell_three(a, b) => 10056 treesort(a,b) => 940049	Elementlar soni:10000 Random eng katta soni:1000  bubblesort(a,b) => 12229978 shakersort(a,b) => 10000 combsort(a,b) => 19984 insertionsort(a, b) =>25365 shellsort(a,b) => 20009 shellsorthib(a,b ) => 10026 shellsortpratt(a ,b) => 40028 myshell_one(a,b) => 10044 myshell_two(a,b) => 9996 myshell_three(a, b) => 20044 treesort(a,b) => 7690211
---	---	--



gnomesort(a,b) =>6510	gnomesort(a,b) => 10044	gnomesort(a,b) =>120041
selectionsort(a,b) => 29963	selectionsort(a,b) => 79987	selectionsort(a,b) => 2090445
heapsort(a,b) => 160055	heapsort(a,b) => 190026	heapsort(a,b) => 460513
quicksort(a,b) => 10013	quicksort(a,b) => 10001	quicksort(a,b) => 10462
quicksort(a,b) => 9979	quicksort(a,b) => 9992	quicksort(a,b) =>10020
mergesort(a,b) => 10120	mergesort(a,b) => 20005	mergesort(a,b) => 29924
mergeinsort(a,b) => 9975	mergeinsort(a,b) => 10004	mergeinsort(a,b) => 9979
newbucketsort(a,b) => 20014	newbucketsort(a,b) => 19967	newbucketsort(a,b) => 19959
radixsort(a,b) => 10060	radixsort(a,b) => 30057	radixsort(a,b) => 19980
radixsortmsd(a,b) => 19992	radixsortmsd(a,b) => 20031	radixsortmsd(a,b) => 20249
bitonicsort(a,b) => 29988	bitonicsort(a,b) => 69974	bitonicsort(a,b) => 220502
timsort(a,b) => 149973	timsort(a,b) => 190021	timsort(a,b) => 140443

Saralash algoritmlarini taqqoslash  
(massivdagi eng katta son 1000)



8.1-rasm. Saralash algoritmlarini taqqoslash.

Ushbu rasmdek keltirilgan taqqoslash natijalariga Shaker, bitli, blokorqali, (MSD),

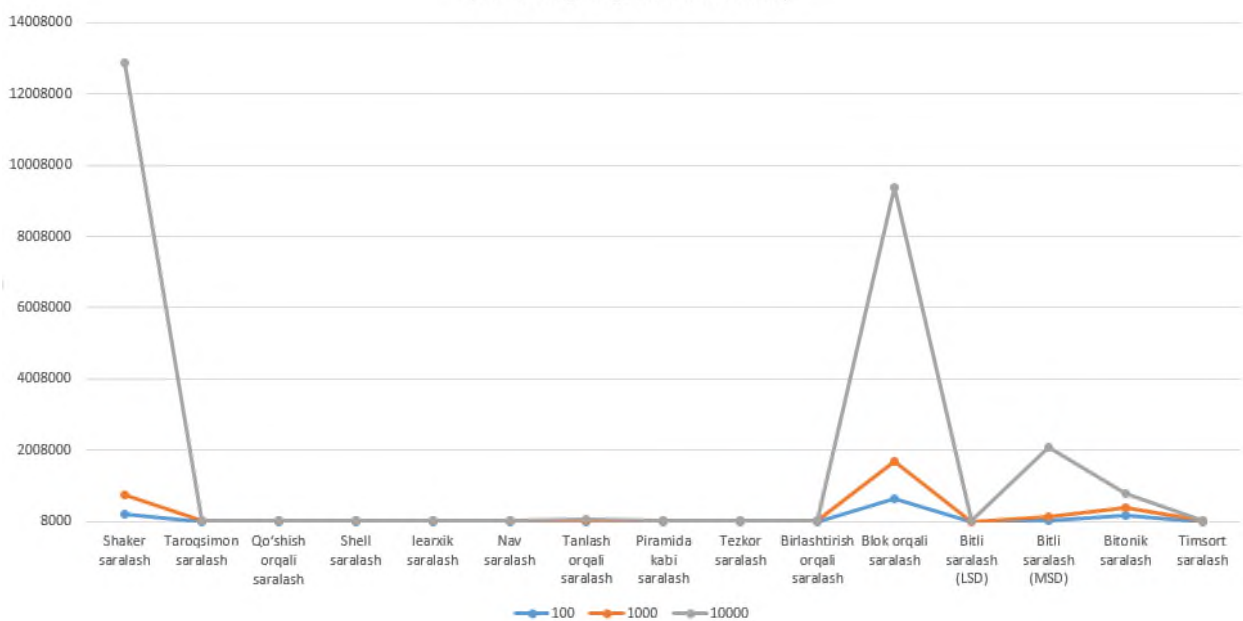
bitoniksaralashalgoritmlariningelementlarningsoninio‘zgarishbilansarala shuchunsarflanadiganvaqtnio‘zgarishinikuzatishimizmumkin.

**Ikkinchitest.**Massivelementlarisoni 100, 1000, 10000 tavatasodifiysonlarningengkattasi 10000:

Elementlar soni:100 Random eng katta soni:10000	Elementlar soni:1000 Random eng katta soni:10000	Elementlar soni:10000 Random eng katta soni:10000
bubblesort(a,b) => 230011	bubblesort(a,b) => 509982	bubblesort(a,b) => 12139952
shakersort(a,b) => 9889	shakersort(a,b) => 9956	shakersort(a,b) => 10056
combsort(a,b) => 10018	combsort(a,b) => 10009	combsort(a,b) => 20001
insertionsort(a, b) => 9562	insertionsort(a, b) => 9986	insertionsort(a, b) => 10009
shellsort(a,b) => 10018	shellsort(a,b) => 9992	shellsort(a,b) => 20001
shellsorthib(a,b ) => 10000	shellsorthib(a,b ) => 10021	shellsorthib(a,b ) => 20018
shellsortpratt(a ,b) => 8452	shellsortpratt(a ,b) => 9992	shellsortpratt(a ,b) => 40006
myshell_one(a,b) => 10022	myshell_one(a,b) => 10025	myshell_one(a,b) => 10017
myshell_two(a,b) => 9988	myshell_two(a,b) => 10360	myshell_two(a,b) => 10005
myshell_three(a, b) => 12001	myshell_three(a, b) => 15201	myshell_three(a, b) => 20018
treesort(a,b) => 660039	treesort(a,b) => 1030011	treesort(a,b) => 7700494
gnomesort(a,b) => 7563	gnomesort(a,b) => 8624	gnomesort(a,b) => 11254
selectionsort(a, b) => 29997	selectionsort(a, b) => 99992	selectionsort(a, b) => 1950542
heapsort(a,b) => 170016	heapsort(a,b) => 210079	heapsort(a,b) => 400520
quicksort(a,b)	quicksort(a,b)	quicksort(a,b)

=> 9562 quicksort(a,b) ) => 9996 mergesort(a,b) => 10069 mergeinsort(a,b) ) => 8996 newbucketsort(a, b) => 20060 radixsort(a,b) => 10005 radixsortmsd(a,b) ) => 10013 bitonicsort(a,b) => 30014 timsort(a,b) => 179974	=> 10009 quicksort(a,b) ) => 10005 mergesort(a,b) => 10000 mergeinsort(a,b) ) => 9996 newbucketsort(a, b) => 20040 radixsort(a,b) => 10018 radixsortmsd(a,b) ) => 9954 bitonicsort(a,b) => 59934 timsort(a,b) => 150012	=> 10210 quicksort(a,b) ) => 10004 mergesort(a,b) => 30318 mergeinsort(a,b) ) => 20502 newbucketsort(a, b) => 19954 radixsort(a,b) => 30075 radixsortmsd(a,b) ) => 29680 bitonicsort(a,b) => 229981 timsort(a,b) => 149973
--	---	--

Saralash algoritmlarini taqqoslash  
(massivdagi eng katta son 10000)



## 8.2-rasm. Saralash algoritmlarini taqqoslash.

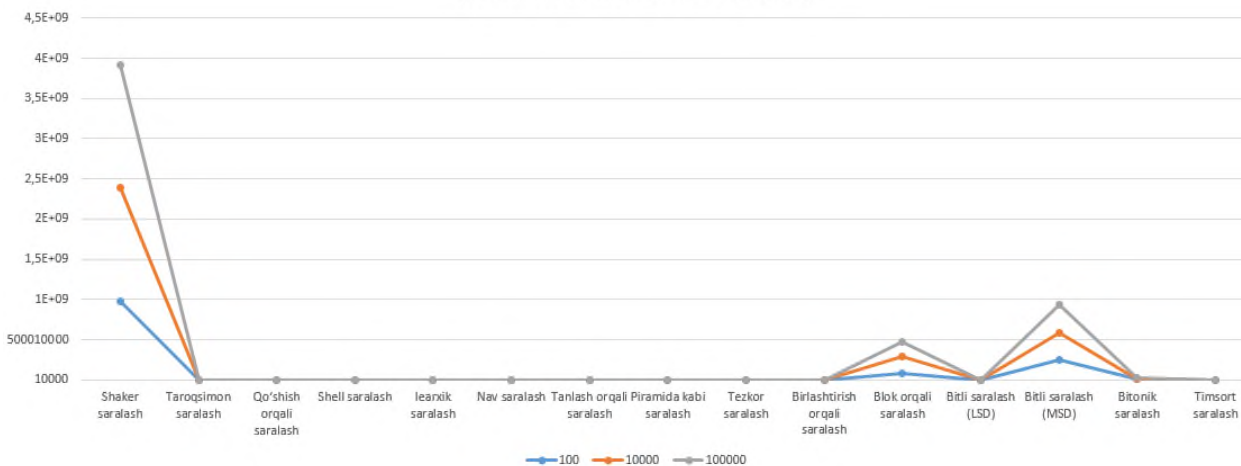
Ushbu rasmda keltirilgan taqqoslash natijalariga Shaker, blok orqali, bitli (MSD), bitonik saralash algoritmlarining elementlarining sonini o'zgarish bilan saralash uchun sarflanadigan vaqtning o'zgarishini kuzatishimiz mumkin.

**Uchinchi test.** Massiv elementlar soni 100000  
tavarasodifiy sonlarning eng kattasi 100, 10000, 100000:

Elementlar soni:100000 Random eng katta soni:100	Elementlar soni:100000 Random eng katta soni:10000	Elementlar soni:100000 Random eng katta soni:100000
bubblesort(a,b) => 977222383 shakersort(a,b) => 10035 combsort(a,b) => 190531 insertionsort(a, b) => 10514 shellsort(a,b) => 110523 shellsorthib(a,b ) => 119988 shellsortpratt(a ,b) => 439987 myshell_one(a,b) => 59994 myshell_two(a,b) => 69969 myshell_three(a, b) => 60028 treesort(a,b) => 81039998 gnomesort(a,b) => 10299 selectionsort(a, b) => 249475206 heapsort(a,b) => 7190050 quicksort(a,b) => 340165 quickinssort(a,b ) => 209933 mergesort(a,b)	bubblesort(a,b) => 1413069355 shakersort(a,b) => 20103 combsort(a,b) => 589990 insertionsort(a, b) => 30142 shellsort(a,b) => 340080 shellsorthib(a,b ) => 299988 shellsortpratt(a ,b) => 1020010 myshell_one(a,b) => 150166 myshell_two(a,b) => 159905 myshell_three(a, b) => 140058 treesort(a,b) => 212009993 gnomesort(a,b) => 20014 selectionsort(a, b) => 338029149 heapsort(a,b) => 6460020 quicksort(a,b) => 229943 quickinssort(a,b ) => 110074 mergesort(a,b)	bubblesort(a,b) => 1522791244 shakersort(a,b) => 10022 combsort(a,b) => 420097 insertionsort(a, b) => 20130 shellsort(a,b) => 240105 shellsorthib(a,b ) => 239879 shellsortpratt(a ,b) => 950139 myshell_one(a,b) => 140276 myshell_two(a,b) => 150114 myshell_three(a, b) => 139913 treesort(a,b) => 173539873 gnomesort(a,b) => 29920 selectionsort(a, b) => 346130116 heapsort(a,b) => 7080178 quicksort(a,b) => 200168 quickinssort(a,b ) => 110095 mergesort(a,b)

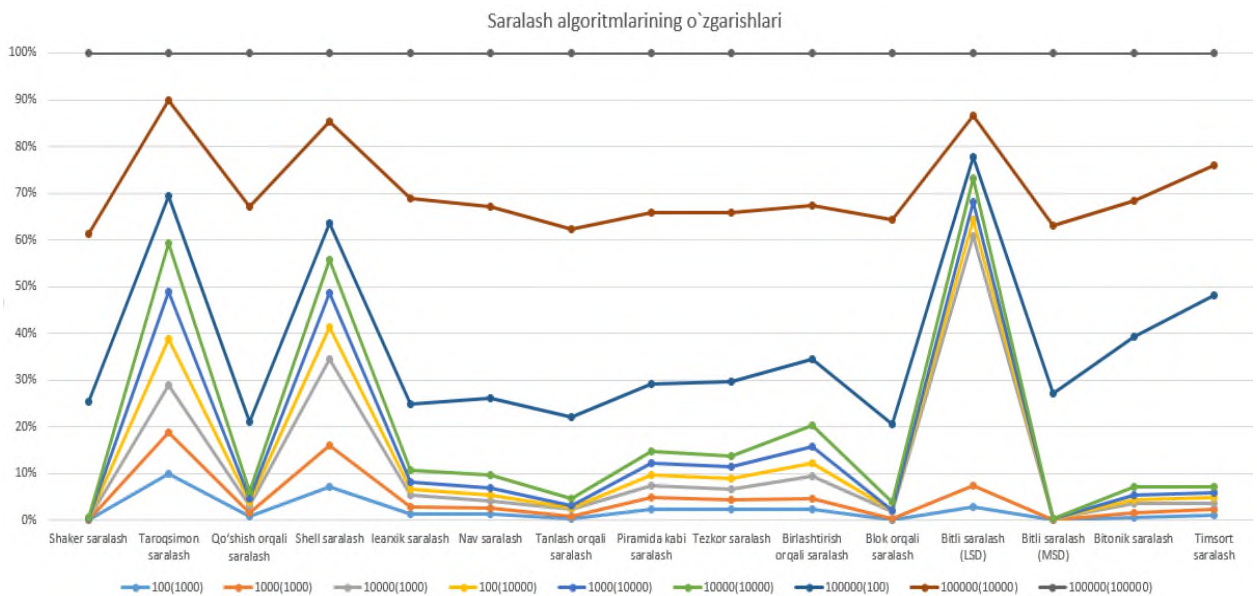
<pre> =&gt; 439970 mergeinsort(a,b) ) =&gt; 259845 newbucketsort(a, b) =&gt; 119831 radixsort(a,b) =&gt; 349931 radixsortmsd(a,b) ) =&gt; 320020 bitonicsort(a,b) =&gt; 4389857 timsort(a,b) =&gt; 469881 </pre>	<pre> =&gt; 520012 mergeinsort(a,b) ) =&gt; 340191 newbucketsort(a, b) =&gt; 110005 radixsort(a,b) =&gt; 400045 radixsortmsd(a,b) ) =&gt; 429952 bitonicsort(a,b) =&gt; 5839897 timsort(a,b) =&gt; 450090 </pre>	<pre> =&gt; 500015 mergeinsort(a,b) ) =&gt; 289851 newbucketsort(a, b) =&gt; 150140 radixsort(a,b) =&gt; 349846 radixsortmsd(a,b) ) =&gt; 570139 bitonicsort(a,b) =&gt; 5079891 timsort(a,b) =&gt; 360411 </pre>
--	--	--

Saralash algoritmlarini taqqoslash  
(massiv elementlar soni 100000)



### 8.3-rasm. Saralash algoritmlarini taqqoslash.

Ushbu rasmda keltirilgan taqqoslash natijalariga Shaker, blok orqali, bitli (MSD) saralash algoritmlarining elementlarning qiymatini o'zgartirganda saralash uchun sarflanadigan vaqtning o'zgarishini kuzatishimiz mumkin.



8.4-rasm. Saralashalgoritmlarining o'zgarishi.

Ushbu tahlil grafikda algoritmlarning massiv elementlar va qiymatlarini o'zgartirganimizda o'zlarini qanday tutishlar foiz miqdorida o'zaro nisbatlar keltirilgan. Masalan,

tanlash orqali saralash o'zini juda yaxshi tutganligini ko'rishimiz mumkin, ya'ni o'zgarishlarda ham uning uchun sarflanadigan vaqt uzluqli o'sgan.

Bu algoritmlarning o'rganishdan maqsad, ular asosida murakkablik kichik, ishlash vaqtida kam bo'lgan saralash algoritmlarini yaratish, masalaning mohiyatidankelib chiqqan holda qachon va qanday qilib foydalanishni belgilab olishimiz mumkin.

**Judam katta sonlar bilan ishlash.** Katta sonlar bilan ishlash uchun asosiy tip va uning xotira dan necha bayte galashini bilish muhim hisoblanadi. Tiplarni yaxshi bilsangiz kerak.

Ammo C++ tilining rivojlanirilgan variantlaridaturlitiplar mavjud. Tiplardan foydalanishga birmisol keltiramiz. Masalan, byte tipi bo'lsin, xotiradan 1 bayte galaydigan. Bu tipga  $a=200$ ,  $b=100$  bo'lsanatiya, 44 bo'ladi. Qanday? Buning uchun avvalo byte ning diapozonini, ya'ni  $[0,255]$  bilish shart. Agar tipning eng yuqori qiymatigachi qannati java 1 ta 0 ni ketmaket qo'yilsa 300 chiqib keladi  $(255+44+1(0))=300$ . Har qanday tipo'zning diapozonidoirasidahisoblashlar nibajaradi. Avvalo C++ standartlarigamostiplarni keltiramiz.

8.2-jadval. Tiplar va ularning xususiyatlari.

No	Tip nomi	bayt	Ikkinchi nomi	Diapozoni
1	int	4	signed	$[-2\ 147\ 483\ 648, 2\ 147\ 483\ 647]$

№	Tipnomi	bayt	Ikkinchinomi	Diapozoni
2	Unsigned int	4	Unsigned	[0, 4 294 967 295]
3	__int8	1	char	[-128, 127]
4	unsigned __int8	1	Unsigned char	[0, 255]
5	__int16	2	Short	[-32 768, 32 767]
6	unsigned __int16	2	UnsignedShort	[0, 65 535]
7	__int32	4		[-2 147 483 648, 2 147 483 647]
8	unsigned __int32	4		[0, 4 294 967 295]
9	__int64	8	Long	[-9 223 372 036 854 775 808, 9 223 372 036 854 775 807]
10	unsigned __int64	8		[0, 18 446 744 073 709 551 615]
11	bool	1		[false] yoki [true]
12	char	1		[-128,127]
13	char	1		[-128 , 127]
14	unsigned char	1		[0, 255]
15	short	2		[-32 768, 32 767]
16	unsigned short	2	unsigned short int	[0, 65 535]
17	long	4	long int	[-2 147 483 648, 2 147 483 647]
18	unsigned long	4	unsigned long int	[0, 4 294 967 295]
19	long long	8	__int64	[-9 223 372 036 854 775 808, 9 223 372 036 854 775 807]
20	unsigned long long	8	__int64	[0, 18 446 744 073 709 551 615]
21	enum	null		
22	float	4		[3,4E +/- 38 (7 belgi)]
23	double	8		[1,7E +/- 308 (15 belgi)]
24	long double	8		[1,7E +/- 308 (15 belgi)]
25	wchar_t	2	__wchar_t	[0, 65 535]

Texnikaning rivojlanishi dasturlashtirish usullaridagi eng yondashuvlarning amalga oshirishga sabab bo'lmoqda. Masalan, tiplarning diapazonini bilish uchun ikki darajasi garraliqilib tuzib chiqmoqdalar. Bujuda oson. Buyuqoridagi jadvaldan ham ko'rish mumkin.

Tiplarning ichida eng kattasi bu long tipidir, uning ikki lani long long eng kattasi bo'lishi mumkin, ammo bu lardan foydalanish uchun \_\_int64 tipi ham yaratilgan. Bu lardan foydalanish qulaydir. Yuqoridagi jadvaldan bu ko'rishimiz ham, o'zimiz dasturiy o'zbek tushirishimiz ham mumkin. Odatdastur chiqarilgan quyidagi fragmentlardan foydalanishadi:

```
#define LL long long
#define uLL unsigned long long
//...
uLL b = 0;
```

Ammo bunday kattasonlar bilan ishlashni C++ standartining o'ziga qaratib qaratmaydi. Bunga ikkita misol keltiramiz.

8.17-dastur. Kattasonlar chegarasini tekshirish.

```
int main()
{
    cout << ULLONG_MAX << endl;
    __int64 a, b = 0; // 20 xona
    cin >> a >> b;
    if ((a+b) > ULLONG_MAX)
        cout << "Chegaradan chiqdi" << endl;
    else
        cout << "Natija:" << a+b << endl;

    system("pause");
    return 0;
}
```

Agar dasturga 20 xona lisonni (< ULLONG\_MAX) kiritsangiz xato ishlaydi. Nima sababdan, sanoq sistemaning xatosini ko'rish mumkin. Shuningdek, ikkinchi 64 darajasine chagateng? Albatta ULLONG\_MAX chagatengdir. Endi yanabir gibrid tip kiritamiz, long double vabuo'raqali  $2^{64}$  darajasini hisoblaymiz.

```
#define LD long double
// ..
cout << ULLONG_MAX << endl;
```



```
LD f = powl(2,64);  
//cout << f << endl;  
printf("%.0Lf\n",f);
```

Dasturfragmentini ishlatib,  
4taxonasigae‘tiborbering.

natijaningoxirgi

Tiplarkombinatsiyalarniyordamidakattasonlarxatolibo‘lganligiuchunama  
liymatematikaningturlimasallarini echganda aniqnechtaxonaaniqligioldin  
danbelgilabolinadi. Masalan, quyidagidasturga qarang.

8.18-dastur. Tiplargabog‘liqkattasonlardanfoydalanish

```
#include "stdafx.h"  
#include <iostream>  
#include <vector>  
#include <math.h>  
using namespace std;  
int main(){  
    double n=0,a=0;  
    cin>>a>>n;  
    vector<double>arr;  
        arr.push_back(pow(a,n));  
    for(int i=0; i<arr.size(); i++)  
        printf("%.0Lf\n", arr[i]);  
    system("pause");  
    return 0;  
}
```

8.18-dastur. Output

```
1- test:{  
son kirit: 2  
    darajasini kirit:50  
1125899906842624  
}  
2- test:{  
son kirit: 2  
    darajasini kirit:64  
18446744073709552000  
}  
3- test:{  
son kirit: 5
```



}

Dasturixtioriysoningixtioriyydarajasinihisoblashuchunmo'ljallan  
gan, ammomhammasiningoxirgiraqamlarigae'tiborbersangiz, 0  
larbilanto'ldirilgan.

Buesayanabirborhisoblashdasanoqsistemasiningxatoliginiko'rsatadi.

Xulosashundaniboratbo'ladi, 20  
xonagachayaxshixisoblashamalgaoshirilishimumkin,  
ammobuxonadanboshlabxatoliklarninazoratqilibbo'lmaydi.

Kattasonlarbilanishlaganda,  
yondashuvniboshqachaamalgaoshirishimumkin,  
masalanoqimdankiruvchiraqamlarnimassivelementlardebqarabuniustida  
mustaqilarifmetikamallarnibajarishimumkin.

Harbirarifmetikamalnioldindan  
C++daamalgaoshirishdasturfragmentiniyozib, Avvalo,  
alohidako'ribchiqishimumkin. cheksizuzoqraqamniifaqatdinamikmassivsifatidafodalashimumkinliginitu  
shunishingizkerak.

Lekinraqamlardinamikmassivsifatidataqdimetilsaham,  
ba'zicheklovlarbo'ladi. Masalan, bundaysonning (massivning)  
uzunligikompyuterxotirasimiqdoribilanchegaralanganbo'ladi.

Shunihamtushunishkerakki,  
qo'shishvaayirishamallaridanfoydalanishdanatijaoperandlargaqaraganda  
kompyuterxotirasidako'proqjoyegallaydi.

**Kattasonlarningqo'shish.**Arifmetikadaishlatiladiganarifmetikqo's  
hishamaliniko'ribchiqamiz.

Ushbuoddiyarifmetikamaluchunalgoritmhayratlanarlidarajadaoddiydir,  
dasturfragmentigaqarang:

```
// birinchi g'oya
if (size_a > size_b)
    length = size_a + 1;
else
    length = size_b + 1;
// ikkinchi g'oya
for (int ix = 0; ix < length; ix++)
{
    b[ix] += a[ix];
    b[ix + 1] += (b[ix] / 10);
    b[ix] %= 10;
}
```

```
// uchinchi g'oya
if (b[length - 1] == 0)
length--;
```

Dasturfragmentida qo'shadigan raqamlarni taxminan a va b massivlarida yoziladi. Raqamlar uzunligi size\_a va size\_b o'zgaruvchilarda saqlanadi,

lekin har qanday bosh qo'zgaruvchilardan foydalanishingiz mumkin.

Birinchi g'oyadan imauchun birifshart operatoribor, nimauchunubuyerda? – degansavoltug'ililadi.

Bu fragment blokida yig'indin natijasida olinadigan sonning maksimal uzunligini aniqlaymiz. Axir, qo'shiladigan raqamlar turli uzunlikda, birikattava ikkinchi siki kichikroq va har bir raqam gamoskelishi uchun xotira ajratishimiz kerak.

Algoritm,

matematikadarslarida o'rgatilgan usul bo'yicha amalga oshiriladi:

birinchi oxiridaboshlab, eng kichik razriyaddan raqam kiritish, natijada summani ajratish va darhol keyingiraqamga qo'shish va o'ngabo'ling andabutun qismini olish.

Natijasida sonining birinchi razriyadivakeyingira zryaduchun qo'shiladigan son bo'ladi, albatta, bor bo'lsa. Asosiy narsa soni b massivga saqlanadi va oxiridachi qishilozim.

Bunda massivni razryadlarini tenglashtirish kerakligini unutmang.

Yuqoridagi dastur fragmentiga o'z g'oyalarimizni qo'shib dasturni keltirami z:

8.19-dastur. Kattasonlarni qo'shish.

```
// Created by MBBahodir
#include "stdafx.h"
#include <iostream>
#include <string>
using namespace std;
int main(){
    string a, b, a1, b1;
    cin >> a >> b;
    int leng = 0;
    // birinchi g'oya
    if (a.length() > b.length())
        leng = a.length()+1;
    else
        leng = b.length()+1;
```

```

// dasturchining g'oyasi
int k = 0;
while(k < leng){
    if( k < leng - a.length() ) a1 += '0';
    if( k < leng - b.length() ) b1 += '0';
    k++;
}
a1 += a;
b1 += b;
// ikkinchi g'oya
int temp = 0;
for (int ix = leng-1; ix >= 0 ; ix--)
{
    int temp_a = a1[ix] - 48;
    int temp_b = b1[ix] - 48; temp_b = temp_b
+ temp;
    temp = 0;
    temp_b += temp_a;
    temp += (temp_b / 10);
    temp_b %= 10;
    a1[ix] = (char)temp_a + 48;
    b1[ix] = (char)temp_b + 48;
}
// uchinchi g'oya
if(b1[0] == '0') {
    b1.erase(0,1);
}
cout << b1 << endl;
system("pause");
return 0;
}

```

#### 8.19-dastur. Output

```

a=984654313156311315315464876546487946456465431311132
15318653186316632158632183218693168316853541867931484
3164131654315313315315365315315331563131831853
b=54646454646465465465455464546546546546546546546
54644654654646431543186464564879876489794648994648979

```

```
4897979874897979879797987489648364584364854684
natija=1531118859620965969970020341093034493003011977
85767869963307840963063701818647783573044806648190862
5804638062111529213293195113352804963696147496686537
```

Dasturnitahlilqilingvaamaldasinabko‘ring.

**Kattasonlarniayirish.** Ikkinchiengko‘pishlatiladiganarifmetikamal ayirishdir.

- Endikattasonlarniayirishalgoritmlariniyozishnio‘rganishgaqaraymiz.

Sonlarimiz a va b massivlardasaqlanadidebfarazqilamiz, mosravishda m va n busonlarninguzunliklaribo‘lsin. Raqamlarnirazryadlarinitenglashtirishniunutmang (yuqoridagidek). Albatta, qaysisonkattaekanliginibilsak, vazifasoddalashadi. Lekinbunibilmasligimizmumkin.

Keyinavvalsonlarningqaysibirikattaekanliginitopishimizkerak.

Olinganraqamningbelgisinianiqlashuchunbukerak.

Boshqachaqilibaytganda, agarbirinchiraqamikkinchisidankambo‘lsa, javobishorali (-) belgisiniko‘rsatadi. Shundayqilib, algoritmningbirinchiqisiminiyozishniboshlaymiz, ya‘nikattasonnianiqlaymiz. Algoritmshundayko‘rinadi:

```
int k = 3; // agar k == 3 sonlar bir xil
length = size_a;
if (size_a > size_b){
    length = size_a;
    k = 1; // agar k == 1 birinchi son
ikkinchisidan uzun
}
else
    if (size_b > size_a) {
        length = size_b;
        k = 2; // agar k == 2 ikkinchi son
birinchisidan uzun
    }
    else // agar ular teng bo‘lsa, kattasini
aniqlash lozim
        for (int ix = 0; ix < length;) //
razryad bo‘yicha tekshirish
        {
            if (a[ix] > b[ix]) // birinchi
sonni razryadi katta bo‘lsa
```

```

        {
            k = 1; // birinchi son
ikkinchisidan katta
            break; // chiqish
        }

        if(b[ix] > a[ix]) // ikkinchi sonni
razryadi katta bo'lsa
        {
            k = 2; // ikkinchi son
birinchisidan katta
            break; // chiqish
        }
    }

```

Endiyozganlarimnitushuntiraman. Avval k o'zgaruvchi 3 gao'rnatilganiniko'rishmumkin. Algoritmningbuqismida k o'zgaruvchitekshirishnatijasihisoblanadi. Agarsonlartengbo'lsa, k=3 gatengbo'libqoladi, agarbirinchisiikkinchisidankattabo'lsa, k= 1 qiymatini, ikkinchisibirinchisidankattabo'lsa, k= 2 qiymatinioladi. Length o'zgaruvchikattasonuzunliginingqiymatinioladi. Endibualgoritmningamaradorliginiasoslashgao'tamiz.

Sonlarnitaqqoslashikkibosqichdaamalgaoshadi. Birinchidan, raqamlaruzunliklarnitaqqoslaymiz: qaysiraqamuzunroqligi, sonlاربirxiluzunlikdabo'lsa, uholdataqqoslashgao'tamiz. Sonlarnitartibbilantaqqoslashniengkattarazryadlardanboshlaymiz, shuninguchunsonningkattasinianiqlaymiz.

Buesabirinchiqismningmohiyativamurakkabligibildiradi.

Endialgoritmningikkinchiqismi — ayirmagao'tamizvauquyidagicha:

```

int difference (int *x, int *y, int *z, int
length){
    for (int ix = 0; ix < (length - 1); ix++){
        if (ix < (length - 1)){
            x[ix + 1]--;
            z[ix] += 10 + x[ix];
        } else
            z[ix] += x[ix];
        z[ix] -= y[ix];
        if (z[ix] / 10 > 0){
            z[ix + 1]++;
        }
    }
}

```

```

        z[ix] %= 10;
    }
}
return 0;
}

```

Ayirmaning o'zi uchun funksiyani yozish qulay, chunki unda ikki hol uchun ikki ta algoritmni yozish shart emas, birinchi son ikkinchisidan kattabo'lgandava aksincha massiv kattason, y massiv kichiksonva z massiv natijahisoblanadi. Algoritm juda oddiy: har bir raqam uchun eng yuqori raqamdan ayirishni hisobga olgan holda 10 niqo'shamiz. Buraqamlar ayirmasini soddalashtirish uchun -1 amalga oshiriladi.

Bu amalfaqat massivning oxirgi raqariyadibo'lsa amalga oshiriladi (son birinchi uchun). Sonlarni ayirgandanso'ng z massividagi son hosil bo'ladi. Javob z massiviga teskari usulida yoziladi. Protse-durani quyidagicha qirish kerak:

```

if (k == 1) difference(a,b,c, length); // birinchi
son ikkinchisidan katta
if (k == 2) difference(b,a,c, length); // ikkinchi
son birinchisidan katta

```

Endi javob shuteskaritartibda massiv dasaqlanadi.

Bu esa kattasonlarni ayirishni amalga oshirishni hal qilib beradi.

**Kattasonlarni ko'paytirish.** Bu algoritm masalalarni yechishda oldin gi ikki tasiga qaraganda keng tarqalgan.

Bevosita algoritmning o'zi gao'taylik.

E'tiboringizga algoritmni taqdim etaman:

```

length = size_a + size_b + 1;
for (int ix = 0; ix < size_a; ix++)
    for (int jx = 0; jx < size_b; jx++)
        c[ix + jx - 1] += a[ix] * b[jx];
for (int ix = 0; ix < length; ix++){
    c[ix + 1] += c[ix] / 10;
    c[ix] %= 10;
}
while (c[length] == 0)
length--;

```

Bu algoritmni quyidagicha ko'ribchi qamiz va qanday ishlashini aniqlash uchun qushunishga harakat qilaylik. Avvaliga a va b massivlardabir xil teskarishaklda ikki tasonbor. Sonlar uzunliklari size\_a va



size\_b o'zgaruvchilardasaqlanadi. length  
o'zgaruvchinatijaviysonuzunliginianiqlaydi.

Uberilgansonlaruzunliklariningyig'indisigayokibuyig'indidanbittagakatt  
atengbo'ladi.

Lekinolingansonninganiquzunliginibilmaganimizuchunuzunroquzunlikn  
i, ya'niikkinchivariantniolamiz. Endiqiyinbo'lmaganhisob-  
kitoblardanso'ng, raqamlarniko'paytiraylik.  
Maktabdao'rganganizmizbo'yichaularniko'paytiramiz. Buninguchun,  
ikkitakrorlanishishlatamiz, biri size\_a uchun, vaboshqa size\_b uchun.  
Shutufaylimassivdagisonniyozishdamassivningharbiryacheykasidaolinga  
nsonningbirraqaminiolamiz.

Oxirgitakrorlanishnatijasidasonininganiquzunliginitopishuchunzarurbo'l  
gansonitaxminuzunligihaqiqiyqismidankattabo'lishimumkin.

Javobsmassivdasaqlanadi.

Dasturlashtilidaamalgaoshirilgandaunitushunishosonroq.

Judakattaraqamlarniko'paytirishimkerak, algoritmiyordamida,  
lekino'zgaruvchilartiplarinimoslashtiraolmaydi.

Bucheklovniqandaybartarafetishmumkinliginitipgaqarabtanlabolishmum  
kin.

```
long int Multi(int A, int B) {
    string Astr = to_string(A);
    string Bstr = to_string(B);

    unsigned long int a, b, c, d, sizeA =
Astr.size(), sizeB = Bstr.size();

    if (sizeA == 1 || sizeB == 1) return A * B;

    unsigned long int n = max(sizeA, sizeB);

    a = A / static_cast<long int>(pow(10, n / 2));
    b = A % static_cast<long int>(pow(10, n / 2));
    c = B / static_cast<long int>(pow(10, n / 2));
    d = B % static_cast<long int>(pow(10, n / 2));

    unsigned long int p1 = Multi(a, c);
    unsigned long int p2 = Multi(b, d);
    unsigned long int p3 = Multi(a, d) + Multi(b,
c);
```

```

return pow(10, n) * p1 + pow(10, n / 2) * p3 +
p2;
}

```

Buhechkingasiremas, butunsonlarmaksimalvaminimalqiymatlaribo'yichachegarasigaega, turliarxitekturalardaturlisha. Masalan, int tipidagibutunsonuchununingqiymatlariorali*g*'i -2147483646 dan 2147483647 gacha. Harbiryo'nalishda 2 milliardbutunsonbo'libtuyuladi, lekinhaqiqiykriptografiyayokimashinalio'qitish, ehtimolliknazariyasiyokizamonaviymatematikagakirganingizdanson'ng, bujudakichikekanliginitushunasiz. Buholatlarda, kattaraqamlarbilanishlashsinflaridanfoydalanishmumkin.

**Large Integer** sinfiningqo'llanilishi. Ko'psonlihisoblashlarstandartma'lumotturlarining cheklanishidantashqarigachiqish, cheksizkattasonlarbilanishlash, hajmifaqatmashinaninghisoblashkuchibilancheklanadi.

Bungaqandayerishishmumkin?

Engmantiqqiyyo'limassivyozihsivastandarttipraqamlarizchilmassivuchun maxsustarздаaylantirish.

Misoluchun, 123456789123456789 raqaminiqandaysaqlashingizmumkin, u int tipigamoskelmaydi. Uniqo'yishuchun, int tipidagimassivolishmumkin, arr[0] = 123456, arr[1] = 789123, arr[2] = 456789, bundayyozihsnimaxsusullaridebbo'ladi. Shuningdek, to'g'rikiritilsa, qo'shish, ayirish, ko'paytirishvabo'lishamallariniyozihsimumkin.

Endimaxsus Bignumber sinfiniamalgaoshirishniboshlaymizvaqo'shish, ayirishuchun. Businfbilanishlashdaizohlarbilandasturfragmentinito'liqkeltirilgan. Dasturfragmentigaqarabizohlarnio'qishingizmumkin.

Kattasonniinitsializasiyaqilish. Quyidagichakiritishishlarisifatida Bignumber(string str) sinfkonstruktorimavjud. Satrningoxiridanboshlabjoriyuzunlikdagi substringlartanlanadi, raqamlargaaylantiriladivaqismlarvektorigaberiladi.

Satrdamanfiyishorabor-

yo'qligigaqarabsinfningshoxususiysohasiinitsializatsiyaqilinadi.

Ba'zannormallashtirishdaziddiyatlarbo'lmasligiuchunsinfningsbarchaob'ektlariuchunbirxilbo'lganstatikdoimiydonishlatiladi.

Kattasonlarni qo'shish. + operatorini qaytayuklaymiz, bu esa o'z navbatida ikki kattasonni qo'shadigan `_plus()` usulini chaqiradi. Bundan tashqari, har ikki raqamlar qismlarini, massiv bir xil hajmli qilish kerak. Bundan keyin esa natijani `_normalization()` funksiyasi bilan normallashtiring.

Kattasonlarni ayirish. Chiqarish qo'shish uchun xuddi shunday ishlaydi, — operatorini qaytayuklanadigan `_minus()` usulidan foydalaniladi.

Kattasonni qo'yingan chiqarish. Chiqish << operatorini qaytayuklashdankelib chiqadi, lekin chiqishning o'zidan oldin sonni `_normalization()` funksiyasi bilan normallashtirish kerak. Ubarcha nol bo'laklarni, bo'laklardagi manfiy sonlarni tozalab, sonni qo'yingan berish kerak.

Sinfining dastur fragmentini keltiramiz, unda izohlab yozilgan:

```
#include "stdafx.h"
#include <iostream>
#include <vector>
#include <algorithm>
#include <iomanip>
#include <string>
using namespace std;
// manfiy sonlar uchun bo'linish
int my_div(int num, int diver) {
    if ((num < 0) && (num % diver))
        return num / diver - 1;
    else
        return num / diver;
}
// manfiy raqamlar uchun modul olish
int my_mod(int num, int diver) {
    if ((num < 0) && (num % diver))
        return num % diver + diver;
    else
        return num % diver;
}
// Katta sonlar sinfi, katta sonli va uzun
arifmetikani saqlash usulini tavsiflaydi
class BigNumber {
private:
```

```

vector<int> chunks;
int sign;
// Katta sonlarni saqlashni sozlash
// takrorlashgadi massiv elemaentlari soni
static const int BASE = 2;
// BASE ga bog'liq (BASE10 = 10^BASE),
normallashtirish uchun
static const int BASE10 = 100;
// xizmatchi funksiyalar
BigNumber _plus(BigNumber &a);
BigNumber _minus(BigNumber &a);
void _normalizationSigns();
void _normalizationChunks();
void _resize(int newsize);
public:
BigNumber operator + (BigNumber &num);
BigNumber operator - (BigNumber &num);
friend ostream & operator << (ostream &os,
BigNumber &num);
int getBASE() {
    return this->BASE;
}
// Konstruktor, bir massivni katta songa
o'zgartiradi
BigNumber(string str) {
    int i;
    if (BASE != 1) {
        // BASE bo'yicha belgilar massivi
        oxiriga yoziladi
        for (i = str.size() - BASE; i >= BASE -
1; i -= BASE) {
            chunks.push_back(stoi(str.substr(i,
BASE)));
        }
    }
    else {
        for (i = str.size() - BASE; i >= BASE;
i -= BASE) {

```

```

        chunks.push_back(stoi(str.substr(i,
BASE)));
    }
}
// son belgisini aniqlash
if (str[0] == '-') {
    sign = -1;
    if (i + BASE - 1 != 0) {
        chunks.push_back(stoi(str.substr(1,
i + BASE - 1)));
    }
}
else {
    sign = 1;
    chunks.push_back(stoi(str.substr(0,
i+BASE)));
}
}
// Konstruktor, parametrsiz, musbat sonlar
uchun
    BigNumber() {
        sign = 1;
    }
};
// massiv o'lchamini o'zgartirish
void BigNumber::_resize(int newSize) {
    chunks.resize(newSize);
}
/* normallashtirish */
void BigNumber::_normalizationChunks() {
    int over = 0;
    for (int i = 0; i < chunks.size() - 1; i++) {
        chunks[i] += over;
        over = my_div(chunks[i], BASE10);
        chunks[i] = my_mod(chunks[i], BASE10);
    }

    chunks[chunks.size() - 1] += over;

```

```

    // qayta ishlash
    if (chunks[chunks.size() - 1] / BASE10) {
        over = my_div(chunks[chunks.size() - 1],
BASE10);
        chunks[chunks.size() - 1] =
my_mod(chunks[chunks.size() - 1], BASE10);
        chunks.push_back(over); // qoldiq bilan
yangi yacheyka yaratish
    }
    return;
}
// chop qilish uchun normallashtirish
void BigNumber::_normalizationSigns() {
    // manfiy bo'lsa
    if (chunks[chunks.size() - 1] < 0) {
        sign = -sign; // ishorasini o'zgartirisha
        chunks[0] = BASE10 - chunks[0]; //
normallashtirish
        for (int i = 1; i < chunks.size(); i++) {
            chunks[i] = (BASE10 - chunks[i] - 1) %
BASE10;
        }
    }
    // 0 larni o'chirish
    int i = chunks.size() - 1;
    while (chunks[i] == 0) {
        if (i == 0) {
            sign = 1;
            return;
        }
        chunks.pop_back();
        i--;
    }
    return;
}
// qo'shish funksiyasi
BigNumber BigNumber::_plus(BigNumber &num) {
    BigNumber res;

```

```

        res.sign = this->sign;
        for (int i = 0; i < this->chunks.size(); i++) {
            res.chunks.push_back(this->chunks[i] +
num.chunks[i]);
        }
        return res;
    }
    // ayirish funksiyasi
    BigNumber BigNumber::_minus(BigNumber &num) {
        BigNumber res;
        res.sign = this->sign;
        for (int i = 0; i < this->chunks.size(); i++) {
            res.chunks.push_back(this->chunks[i] -
num.chunks[i]);
        }
        return res;
    }
    // katta sonlarni qo'shish uchun + operatori
    BigNumber BigNumber::operator + (BigNumber &num) {
        BigNumber res;
        // hajmni aniqlash
        if (this->chunks.size() > num.chunks.size()) {
            num._resize(chunks.size());
        }
        else {
            (*this)._resize(num.chunks.size());
        }
        // son ishorasini aniqlash
        if (sign == num.sign) {
            res = (*this)._plus(num);
        }
        else {
            res = (*this)._minus(num);
        }
        res._normalizationChunks();
        return res;
    }
    // katta sonlarni ayirish uchun - operatori

```

```

BigInteger BigInteger::operator - (BigInteger &num) {
    BigInteger res;
    // hajmni aniqlash
    if (this->chunks.size() > num.chunks.size()) {
        num._resize(chunks.size());
    }
    else {
        (*this)._resize(num.chunks.size());
    }
    // son ishorasini aniqlash
    if (sign != num.sign) {
        res = (*this)._plus(num);
    }
    else {
        res = (*this)._minus(num);
    }
    res._normalizationChunks();
    return res;
}
// << operatorini qayta yuklash
ostream & operator << (ostream &os, BigInteger &num)
{
    num._normalizationSigns();
    if (num.sign == -1) {
        os << '-';
    }
    os << num.chunks[num.chunks.size() - 1];
    for (int i = num.chunks.size() - 2; i >= 0; i--
) {
        os << setw(num.getBASE()) << setfill('0')
<< num.chunks[i];
    }
    return os;
}

```

Sinfni ishlatishgadamurkeltiramiz:

8.20-dastur. BigInteger sinfidan foydalanish.

```

int main() {
    BigInteger n1("-

```





OpenSSL kattasonlar bilan ishlash uchun ajralmas mexanizmi, funksiyalaribor. Agarkutubxonabo'lsa, unday quyidagicha foydalanish mumkin:

```
#include "stdafx.h"
#include <climits>
#include <openssl/bn.h>
#include <stdio.h>

int main() {
    BN_CTX *ctx = BN_CTX_new(); // hisoblash uchun
    massiv
    BIGNUM *mybignum = nullptr; // ko'paytirilishi
    kerak bo'lgan son
    BIGNUM *mul = BN_new(); // natija

    BN_dec2bn(&mybignum, "18446744073709551615"); //
    2^64-1
    BN_mul(mul, mybignum, mybignum, ctx);

    // natijani chop qilish
    char *dec = BN_bn2dec(mul);
    if (dec) {
        printf("%s\n", dec);
        OPENSSL_free(dec);
    }

    // xotirani tozalash
    BN_free(mybignum);
    BN_free(mul);
}
```

Bu C++danko'proq C kodidir. Bu istisnalar bilan birmuhitda bilan ishlash uchun xavfli va xotira oqib ketishi mumkin.

Gcrypt. Libgcrypt kutubxonasi GnuPG loyihasi asosida yaratilgan va shifrlash bilan ishlash uchun asosiy funksiyalarni, shu jumladan kattasonlar bilan ta'minlaydi. Bu kutubxonadan quyidagicha foydalanish mumkin:

```
#include <gcrypt.h>
```

```

#include <cassert>

int main() {
    unsigned long long mybignum =
18446744073709551615ull;
    gcry_mpi_t max_ul = gcry_mpi_new(64);
    gcry_mpi_t mul = gcry_mpi_new(128);

    size_t scanned = 0;
    gcry_mpi_scan(&max_ul, GCRYMPI_FMT_USG, &mybignum,
sizeof(unsigned long long), &scanned);
    assert(scanned==sizeof(unsigned long long) &&
"failed to scan the whole number");

    gcry_mpi_mul(mul, max_ul, max_ul);

    gcry_mpi_dump(mul);

    gcry_mpi_release(mul);
    gcry_mpi_release(max_ul);
}

```

Muammolarhalihambirxil - C kodi, undaresurlarnio‘zimizyaratishimizmumkin.

GMP.

Avvalgiikkikutubxonakriptografiyadafoydalanishuchunishlatiladi. GMP faqatraqamlarbilanishlaydiganmaxsuskutubxona.

Uningyordamidaquyidagichadasturtuzishmumkin:

```

#include <stdio.h>
#include <gmp.h>

int main() {
    mpz_t mybignum;
    mpz_t mul;

    mpz_init_set_str(mybignum, "18446744073709551615",
10);
    mpz_init(mul);

```

```

mpz_mul(mul, mybignum, mybignum);
gmp_printf("%Zd\n", mul);

mpz_clear(mybignum);
mpz_clear(mul);
}

```

Boost.Multiprecision.

Bukutubxonakattasonlarningo‘zibilanarifmetikaniamalgaoshirmaydi, faqatularbilanishlashuchunqulay, C++ interfeysinita‘minlaydi, uchinchitomonkutubxonalaridanhisob-kitoblaruchun backend sifatidafoydalanadi.

Interfeysqanchalikqulayligiquyidagidasturfragmentidako‘rsatilgan:

```

#include <boost/multiprecision/cpp_int.hpp>
#include <iostream>

int main() {
    using namespace boost::multiprecision;
    int128_t mybignum = 18446744073709551615ull;
    cout << mybignum * mybignum << endl;
}

```

Bundaytez-

tezuchraydiganmasalalaruchunkutubxonalaridanfoydalanishdaxotiravabajarishjudamushkul. C++ standartlashtirishbo‘yichaMilliyishchiguruhdanrusishlabchiquvchilaribir xilishonishadi, shuninguchunstandartkutubxonagamustaqilravishdawide integer tiplarini kiritishtaklifibilanchiqqan. Ulardandasturdaquyidagichafoydalanishmumkin:

```

#include <wide_integer>
#include <iostream>

int main() {
    auto mybignum = 18446744073709551615_int128;
    std::cout << mybignum * mybignum << std::endl;
}

```

Megabaytxotiraoladiganraqamlarbilanishlashniistaysizmi?

Hechqandaymuammoyo‘q, 100!nihisoblashdasturini keltiramiz:

```

#include <wide_integer>

```

```

#include <iostream>

using int_mb = wide_int<1024*1024>;

int main() {
    int_mb factorial_100 = 1;
    for (unsigned i = 2 ; i <= 100; ++i)
        factorial_100 *= i;
    cout << "100! = " << factorial_100 << endl;
}

```

Yuqoridagilargaasoslanib, tadqiqotlarnatijasidabiztomondantaklifqilinadigankattasonlarbilanishlash gamo‘ljallanganBigIntkutubxonaga 2002 yildaishlabchiqganmizvauniturlimamlakatlargafoydalanishuchunberilgan.

Internetdahammodifikatsiyalanganvariantlariniyokioriginalo‘zinitopishin gizmumkin. Ushbukutubxonanito‘liqligichakeltiramiz:

```

#include <stdc++.h>

using namespace std;

typedef int64_t ll;
typedef long long ll;

#define EL printf("\n")
#define pb push_back
#define FOR(i,l,r) for (int i=l;i<=r;i++)
#define FORD(i,r,l) for (int i=r;i>=l;i--)

const int base = 1e9;
typedef vector<int> BigInt;

void Set(BigInt &a) {
    while (a.size() > 1 && a.back() == 0)
        a.pop_back();
}

void Print(BigInt a) {

```

```

    Set(a);
    printf("%d", (a.size() == 0) ? 0 : a.back());
    FORD(i,a.size()-2,0) printf("%09d", a[i]); EL;
}

BigInt Integer(string s) {
    BigInt ans;
    if (s[0] == '-') return ans;
    if (s.size() == 0) {ans.pb(0); return ans;}
    while (s.size()%9 != 0) s = '0'+s;
    for (int i=0;i<s.size();i+=9) {
        int v = 0;
        for (int j=i;j<i+9;j++) v = v*10+(s[j]-
'0');
        ans.insert(ans.begin(),v);
    }
    Set(ans);
    return ans;
}

BigInt Integer(char c[]) {
    string s = "";
    FOR(i,0,strlen(c)-1) s = s + c[i];
    return Integer(s);
}

BigInt Integer(ll x) {
    string s = "";
    while (x > 0) s = char(x%10+'0') + s, x /= 10;
    return Integer(s);
}

BigInt Integer(int x) {
    return Integer((ll) x);
}

void operator >> (istream &in, BigInt &a) {
    string s;

```

```

    getline(cin, s);
    a = Integer(s);
}

void operator << (ostream &out, BigInt a) {
    Print(a);
}

bool operator < (BigInt a, BigInt b) {
    Set(a);
    Set(b);
    if (a.size() != b.size()) return (a.size() <
b.size());
    FORD(i,a.size()-1,0)
        if (a[i] != b[i]) return (a[i] < b[i]);
    return false;
}

bool operator > (BigInt a, BigInt b) {
    return (b < a);
}

bool operator == (BigInt a, BigInt b) {
    return (!(a < b) && !(b < a));
}

bool operator <= (BigInt a, BigInt b) {
    return (a < b || a == b);
}

bool operator >= (BigInt a, BigInt b) {
    return (b < a || b == a);
}

bool operator < (BigInt a, int b) {
    return (a < Integer(b));
}

```

```

bool operator > (BigInt a, int b) {
    return (a > Integer(b));
}

bool operator == (BigInt a, int b) {
    return (a == Integer(b));
}

bool operator >= (BigInt a, int b) {
    return (a >= Integer(b));
}

bool operator <= (BigInt a, int b) {
    return (a <= Integer(b));
}

BigInt max(BigInt a, BigInt b) {
    if (a > b) return a;
    return b;
}

BigInt min(BigInt a, BigInt b) {
    if (a < b) return a;
    return b;
}

BigInt operator + (BigInt a, BigInt b) {
    Set(a);
    Set(b);
    BigInt ans;
    int carry = 0;
    FOR(i,0,max(a.size(), b.size())-1) {
        if (i < a.size()) carry += a[i];
        if (i < b.size()) carry += b[i];
        ans.pb(carry%base);
        carry /= base;
    }
    if (carry) ans.pb(carry);
}

```



```

    Set(ans);
    return ans;
}

BigInt operator + (BigInt a, int b) {
    return a + Integer(b);
}

BigInt operator ++ (BigInt &a) { // ++a
    a = a + 1;
    return a;
}

void operator += (BigInt &a, BigInt b) {
    a = a + b;
}

void operator += (BigInt &a, int b) {
    a = a + b;
}

BigInt operator && (BigInt a, BigInt b) {
    Set(a);
    Set(b);
    BigInt ans;
    int carry = 0;
    FOR(i,0,a.size()-1) {
        carry += a[i] & (i < b.size() ? b[i] : 0);
        if (carry < 0) ans.pb(carry+base), carry =
-1;
        else ans.pb(carry), carry = 0;
    }
    Set(ans);
    return ans;
}

BigInt operator && (BigInt a, int b) {
    return a && Integer(b);
}

```

```

}

void operator -- (BigInt &a) { // --a
    a = a && 1;
}

void operator -= (BigInt &a, BigInt b) {
    a = a + b;
}

void operator -= (BigInt &a, int b) {
    a = a && b;
}

BigInt operator * (BigInt a, BigInt b) {
    Set(a);
    Set(b);
    BigInt ans;
    ans.assign(a.size()+b.size(), 0);
    FOR(i,0,a.size()-1) {
        ll carry = 0ll;
        for (int j=0;j<b.size() || carry > 0;j++) {
            ll s = ans[i+j] + carry +
(11)a[i]*(j<b.size()?(11)b[j]:011);
            ans[i+j] = s%base;
            carry = s/base;
        }
    }
    Set(ans);
    return ans;
}

BigInt operator * (BigInt a, int b) {
    return a * Integer(b);
}

void operator *= (BigInt &a, BigInt b) {
    a = a * b;
}

```

```

}

void operator *= (BigInt &a, int b) {
    a = a * b;
}

BigInt operator / (BigInt a, BigInt b) {
    Set(a);
    Set(b);
    if (b == Integer(0)) return Integer("-1");
    BigInt ans, cur;
    FORD(i,a.size()-1,0) {
        cur.insert(cur.begin(), a[i]);
        int x = 0, L = 0, R = base;
        while (L <= R) {
            int mid = (L+R)>>1;
            if (b*Integer(mid) > cur) {
                x = mid;
                R = mid-1;
            }
            else
                L = mid+1;
        }
        cur = cur && Integer(x-1)*b;
        ans.insert(ans.begin(),x-1);
    }
    Set(ans);
    return ans;
}

BigInt operator / (BigInt a, int b) {
    Set(a);
    BigInt ans;
    ll cur = 0ll;
    FORD(i,a.size()-1,0) {
        cur = (cur*(ll)base + (ll)a[i]);
    }
}

```

```

        ans.insert(ans.begin(), cur/b);
        cur %= b;
    }
    Set(ans);
    return ans;
}

void operator /= (BigInt &a, BigInt b) {
    a = a / b;
}

void operator /= (BigInt &a, int b) {
    a = a / b;
}

BigInt operator % (BigInt a, BigInt b) {
    Set(a);
    Set(b);
    if (b == Integer(0)) return Integer("-1");
    BigInt ans;
    FORD(i, a.size()-1, 0) {
        ans.insert(ans.begin(), a[i]);
        int x = 0, L = 0, R = base;
        while (L <= R) {
            int mid = (L+R)>>1;
            if (b*Integer(mid) > ans) {
                x = mid;
                R = mid-1;
            }
            else
                L = mid+1;
        }
        ans = ans && Integer(x-1)*b;
    }
    Set(ans);
    return ans;
}

```

```

int operator % (BigInt a, int b) {
    Set(a);
    if (b == 0) return -1;
    int ans = 0;
    FORD(i,a.size()-1,0)
        ans = (ans*(base%b) + a[i]%b)%b;
    return ans;
}

void operator %= (BigInt &a, BigInt b) {
    a = a % b;
}

void operator %= (BigInt &a, int b) {
    a = a % Integer(b);
}

BigInt gcd(BigInt a, BigInt b) {
    Set(a);
    Set(b);
    while (b > Integer(0)) {
        BigInt r = a%b;
        a = b;
        b = r;
    }
    Set(a);
    return a;
}

BigInt lcm(BigInt a, BigInt b) {
    return (a*b/gcd(a,b));
}

BigInt sqrt(BigInt a) {
    BigInt x0 = a, x1 = (a+1)/2;
    while (x1 < x0) {
        x0 = x1;
    }
}

```

```

        x1 = (x1+a/x1)/2;
    }
    return x0;
}

BigInt pow(BigInt a, BigInt b) {
    if (b == Integer(0)) return Integer(1);
    BigInt tmp = pow(a, b/2);
    if (b%2 == 0) return tmp * tmp;
    return tmp * tmp * a;
}

BigInt pow(BigInt a, int b) {
    return pow(a, (Integer(b)));
}

int log(int n, BigInt a) { // log_n(a)
    Set(a);
    int ans = 0;
    while (a > Integer(1)) {
        aC++;
        a /= n;
    }
    return ans;
}

```

Ushbukutubxonadanfoydalanishjudaosonbo‘libhisoblanadi.

```

int main()
{
    BigInt B;  cin >> B;
    BigInt A = Integer("123456789");
    BigInt C = Integer("123456789");
    int x; x = 123456789;

    if (B <= A) cout << A - B;
    else {
        cout << "-";
        cout << B - A;
    }
}

```

```

cout << A + B; Print(A + x);
cout << A * B; Print(A * x);
cout << A / B; Print(A / x);
cout << A % B; printf("%d\n", A % x);

C = ++A; ++B; C += B + x;
Print(A); Print(B); Print(C);

cout << max(A,B);
cout << min(A,B);

cout << gcd(A,B);
cout << lcm(A,B);

cout << sqrt(A);
printf("%d %d %d\n", log(2,A), log(10,B),
log(5,C));

A = Integer(28); x = 80;
cout << pow(A,B);
cout << pow(A,x);

return 0;
}

```

Turlimatematikamallarnibajaradiganmaxsussinlarvakutubxonalar  
io‘rganganbo‘lsangizkerak. Kattasonlarbilanishlashhamxuddishunday.  
Ixtiyoriymasalagayondashuvasosidakattasonlarbilanishlovchimaxsussin  
larniyaratishmumkin. Ammo, shunihaminobatgaolishkerakki,  
ularbilanishlagandaxotiraninghajmi, birtomonlamafunksiyalar,  
tubsonlarikabitestlashlardano‘tkazibfoydalanishmaqсадgamuvofiq.

## NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1. Saralashalgoritmlariningqaydayturlarinibilasiz?

2. Havosharchakabisaralashalgoritminingtasnifinivamurakkabli giniayting?
3. Massivningtartiblanmaganqisminianiqlashuchunikkita begin va end ko'rsatikichniqo'llabquvvatlaydisaralashalgoritminibilasizmi?
4. "Toshbaqalar" saralashalgoritmlarinisanabbering?
5. Murakkabligio'rtachavaengyomonholuchun $O(n^2)$  vaengyaxshiuchun $O(n)$  bo'lgansaralashalgoritminiayting?
6. Shell caralashalgoritminitasniflabbering?
7. multiset konteynerdanqaysisaralashalgoritmidafoydalanishmumkin?
8. Qo'shishorqalisaralashgao'xshashkabisaralaydiganalgoritmlarnisanabbering?
9. Tanlashorqalisaralashg'oyasinirivojlantirilganvariantibuqaysialgoritmi?
10. Massivniikkigabo'lasiz, qismlarnirekursivravishdatartiblaysizvakeyinbirlashtirishprotsedurasinib ajarasiz. Buqaysisaralashalgoritmi?
11. Bitlisaralashalgoritmlarningqandayvariantlaribor?
12. Bitonketma-ketliknimanianglatadi vaqaysisaralashalgoritmdaishlatiladi.
13. Saralashalgoritmlarininimauchuntaqqoslanadi?
14. Tasodifiysonlarbilanto'ldirilganmassivnisaralashsasodifiysonlarce hegarasigabog'liqmi?
15. Tasodifiysonlarningengkattasi 10000
- bo'lgandao'ziniengyomontutudigansaralashalgoritminiayting?
16. Massivelementlarisoni 100000
- bo'lgandao'ziniyomontutudigansaralashalgoritmlariniayting?
17. Kattasonlarbilanishlashbilanishlashnimauchunkerak?
18. Nimauchunsonlargadiapozontushunchasikiritilgan?
19. Sonlarustidaarifmeikamallarnibajargadaxatonatijanimauchunchiqishinitushuntiribbering?
20. \_\_int64tipiningdiapozoniniayting?
21. Kattasonlarchegarasinitekshirishmumkinmi?
22. 20 xonaaniqlikdaishlovchitipniqandaytip?
23. Kattasonlarbilanishlagandaxatoliklarqachonchiqibkeladiyokinece haxonadankeyin. Nimauchunvamisolkeltiring?
24. Kattasonlarnimostiylardanfoydalanmasdanhamarifmetikamallarnibajarishmumkinmi, qandayvanimauchun?



25. Kattasonlarnimassivgaolib, qo'shishamalinibajarishg'oyasiniayting?
26. Kattasonlarnimassivgaolib, ayirishamalinibajarishdagidastlabkiasosiyg'oyanimadaniborat?
27. Kattasonlarnimassivostilargaolibajarishmumkinmi?
28. Kattasonlaruchunturlioperatorlarnianiqlashmumkinmivaqanday?
29. OpenSSL qandaykutubxonavanimauchunishlatiladi?
30. Libcrypt kutubxonasiqandaykutubxonavanimauchunishlatiladi?
31. GMP qandaykutubxonavanimauchunishlatiladi?
32. Boost.Multiprecision kutubxonasinimasibilanustun?
33. C++

standartlashtirishbo'yichaMilliyishchiguruhdanrusishlabchiquvchilarista ndartkutubxonagamustaqilravishdaqandaytiplarini kiritishtaklifibilanchiq qan?

34. using int\_mb = wide\_int<1024\*1024>; dasturfragmentinitushuntiribbering?

35. BigIntkutubxonasihaqidanimanibilasizvaundagi stoi funksiyasiningvazifasinitushuntiribbering.



## AMALIY KO'NIKMA VA MALAKALARNI ANIQLASH HAMDA RIVOJLANTIRISH UCHUN ASSISMENT TOPSHIRIQLARI.

<b>BIRINCHI ASSISMENT TOPSHIRIG'I</b>	
<p>Murakkabsaralashalgoritmlarigaoidberilganquyidagidasturbo'yichaberilgantopshiriqlarkeraklifragmentlariasosidabajaring.</p> <p> Bundadasturdagiba'zio'zgartirishlarnitopishorqalitopshiriqlarbosqichma bosqichamalgaoshiriladi.</p>	
dastur	topshiriqlar
<pre>void shakersort(int* l, int* r) {     int sz = r - l;     if (sz &lt;= 1) return;     bool b = true;</pre>	<p>1. Nechtasaralashalgoritmiberilgan.</p> <hr style="border: 0; border-top: 1px solid black; margin: 5px 0;"/> <hr style="border: 0; border-top: 1px solid black; margin: 5px 0;"/> <p style="text-align: center;">—</p>

<pre> int* beg = l - 1; int* end = r - 1; while (b) {     b = ;     beg++;     for (int* i = beg; i &lt; end; i++) {         if (*i &gt; *(i + 1)) {             swap(*i, *(i + 1));             b = true;         }     }     if (!b) break;     end--;     for (int* i = end; i &gt; beg; i--) {         if (*i &lt; *(i - 1)) {             swap(*i, *(i - 1));             b = true;         }     } } void insertionsort(int* l, int* r) {     for (int *i = l + </pre>	<p>2. shakersort(int* l, int* r)funksiyasidandasturdafoydalanibko‘ring.</p> <hr/> <hr/> <hr/> <hr/> <hr/>
	<p>3. insertionsort(int* l, int* r)funksiyasidandasturdafoydalanibko‘ring.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
	<p>4. shellsort(int* l, int* r)funksiyasidandasturdafoydalanibko‘ring.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>

```

1; i < r; i++) {
    int* j = i;
    while (j > l
|| *(j - 1) > *j) {
        swap(*(j
- 1), *j);
        j--;
    }
}
void shellsort(int*
l, int* r) {
    int sz = r - l;
    float step = sz /
2.0;
    while (step >= 1)
{
    for (int *i =
l + step; i < r; i++)
{
        int *j =
i;
        int *diff
= j - step;
        while
(diff >= l && *diff >
*j) {
            swap(*diff, *j);
            j =
diff;
            diff
= j - step;
        }
    }
    step /= 2;
}
}

```

5. Shellsaralashalgoritmi uchun yangi variant yaratning.

---



---



---



---



---



---

6. Satrlarni saralash algoritmini tuzing.

---



---



---



---




---



---

7. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. \_\_\_\_\_
8. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.

### IKKINChIASSIMENTTOPShIRIG‘I

	<p>Saralashalgoritmlaritaqqoslashgaoidberilganquyidagidasturbo‘yichaberilgantopshiriqlarkeraklifragmentlariasosidabajaring.</p> <p>☝ Bundadasturdagiba‘zio‘zgartirishlarnitopishorqalitopshiriqlarbosqichma — bosqichamalغانoshiriladi.</p>
dastur	topshiriqlar
<pre>void bubblesort(int* l, int* r) void shakersort(int* l, int* r) void combsort(int* l, int* r) void insertionsort(int* l, int* r) void shellsort(int* l, int* r) void shellsorthib(int* l, int* r) void shellsortpratt(int* l, int* r) void myshell_one(int* l, int* r) void myshell_two(int* l, int* r) void myshell_three(int* l, int* r) void treesort(int* l, int* r) void gnomesort(int*</pre>	<p>1. Nechtasaralashalgoritmiberilgan.</p> <p>_____</p> <p>_____</p> <p>—</p> <p>2. shakersort(int* l, int* r)va myshell_one(int* l, int* r)funksiyalarnitaqqoslashdasturinituzing.</p> <p>_____</p> <p>—</p> <p>_____</p> <p>_____</p> <p>3. newbucketsort(int* l, int* r)varadixsortmsd(int* l, int* r)funksiyalarnitaqqoslashdasturinituzing.</p> <p>_____</p> <p>—</p> <p>_____</p> <p>_____</p> <p>_____</p>

<pre> l, int* r) void selectionsort(int* l, int* r) void heapsort(int* l, int* r) void quicksort(int* l, int* r) </pre>	<hr/> <hr/> <hr/> <hr/>
<pre> void quickinssort(int* l, int* r) void mergesort(int* l, int* r) void mergeinssort(int* l, int* r) void newbucketsort(int* l, int* r) void radixsort(int* l, int* r) void radixsortmsd(int* l, int* r) </pre>	<p>4. 2</p> <p>tasaralashalgoritmlarinitasidifiyengkatasoni 100000 bo'lgan 100000 taelementdaniboratmassivlarnitaqqoslang.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
<pre> voidbitonicsort(int* l, int* r) void timsort(int* l, int* r) </pre>	<p>5. Shellva <span style="float: right;">3</span></p> <p>tasaralashalgoritmlarinitasidifiyengkatasoni 100000 bo'lgan 100000 taelementdaniboratmassivlarnitaqqoslang.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
<p>6. Dasturdajamibo'lib, nechao'zgartirishkiritildi. _____</p> <p>7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.</p>	

## UChINChIASSIMENTTOPShIRIG'I

🚩	<p>Judahamkattasonlarbilanishlashgaoidberilganquy idagidasturbo'yichaberilgantopshiriqlarkeraklifragmentlariasosidabajaring.</p> <p>👉 Bundaydasturdagiba'zio'zgartirishlarnitopishorq alitopshiriqlarbosqichma — bosqichamalganoshiriladi.</p>
---	--

dastur	topshiriqlar
<pre>// Birinchidastur. #include "stdafx.h" #include &lt;iostream&gt; #include &lt;vector&gt; #include &lt;math.h&gt; using namespace std; int main(){     double n=0,a=0;     cin&gt;&gt;a&gt;&gt;n;      vector&lt;double&gt;arr;      arr.push_back(pow(a,n));     for(int i=0; i&lt;arr.size(); i++)          printf("%.0Lf\n", arr[i]);     system("pause");     return 0; } // Ikkinchidastur. // Created by MBBahodir #include "stdafx.h" #include &lt;iostream&gt;</pre>	<p>1. Dasturkachonxatoqisblashniamalgaoshiriradi.</p> <p>_____</p> <p>_____</p> <p>—</p>
	<p>2. Vektordanboshqakonteynerdanfoydalanishmumkinmi. Dasturdaishlatibko'ring</p> <p>_____</p> <p>—</p> <p>_____</p> <p>—</p> <p>_____</p>
	<p>3. Aynannimauchunprintf("%.0Lf\n", arr[i]);foydalanilgan.</p> <p>_____</p> <p>—</p> <p>_____</p> <p>—</p> <p>_____</p> <p>—</p> <p>_____</p> <p>—</p> <p>_____</p>



<pre>                 {                     int temp_a = a1[ix] - 48;                     int temp_b = b1[ix] - 48; temp_b = temp_b + temp;                     temp = 0;                     temp_b += temp_a;                     temp += (temp_b / 10);                     temp_b %= 10;                     a1[ix] = (char)temp_a + 48;                     b1[ix] = (char)temp_b + 48;                 } // uchinchi g'oya if(b1[0] == '0') {      b1.erase(0,1); } cout &lt;&lt; b1 &lt;&lt; endl; system("pause"); return 0; } </pre>	<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
---	---

6. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. \_\_\_\_\_
7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.

<b>TO‘RTINCHI ASSISMENT TOPSHIRIG‘I</b>	
	<p><b>Large Integer</b></p> <p>sinfiningqo‘llanilishigaoidberilganquyidagidasturbo‘yichaberilgantopshiriqlarkeraklifragmentlari asosidabajaring.</p> <p> Bundasturdagiba‘zio‘zgartirishlarnitopishorq</p>



alitoqshiriqlar bosqichma bosqich amalga oshiriladi.	
dastur	topshiriqlar
<pre>// qism dastur bool operator &lt; (BigInt a, BigInt b) {     Set(a);     Set(b);     if (a.size() != b.size()) return (a.size() &lt; b.size());     FOR(i, a.size()- 1, 0)         if (a[i] != b[i]) return (a[i] &lt; b[i]);     return false; }  bool operator &gt; (BigInt a, BigInt b) {     return (b &lt; a); }  bool operator &lt;= (BigInt a, BigInt b) {     return (a &lt; b    a == b); }</pre>	<p>1. Dastur nechta taqqoslash operatori keltirilgan.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <p>2. operator <code>==</code> ni <code>BigInt</code> tiplarini taqqoslash funksiyasini yozing</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <p>3. operator <code>&lt;</code> ni <code>BigInt</code> va <code>int</code> tiplarini taqqoslash funksiyasini yozing</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>

<pre> bool operator &gt;= (BigInt a, BigInt b) {     return (b &lt; a    b == a); }  bool operator &gt; (BigInt a, int b) {     return (a &gt; Integer(b)); }  BigInt max(BigInt a, BigInt b) {     if (a &gt; b) return a;     return b; }  BigInt operator + (BigInt a, int b) {     return a + Integer(b); }  void operator += (BigInt &amp;a, BigInt b) {     a = a + b; } </pre>	<p>4. 2          BigInttiplarinikichiginitopuvchiminif          unksiyasiniyozing</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
	<p>5. operator ++niyozing.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
<p>6. Dasturdajamibo‘lib, nechao‘zgartirishkiritildi. _____</p> <p>7. Shudasturninganaloginiyaratishsizgamustaqilvazifadir.</p>	

## 3-BOB. VISUAL C++ MUHITIDA DASTURLASH.

### 3.1. Visual C++ muhitidasturlash.

📖 Integrallashganishlabchiqarishmuhiti, ularniyaratilishtarixi, ribovlanishbosqichlari, yagonatillivako‘ptilliintegrallashganmuhitlar, Vizual Studio variantlarivaimkoniyatlari, .NET frameworkningimkoniyatlari, Visual C++ da Windows ilovalarniyaratish, Muhitdamenyularvauskunalarpaneli, Form xususiyatlarivahodisalaridanfoydalanishbo‘yichanazariyhamdavanamun alarikeltirilganbo‘lib, nazariybilimlarniasoslashuchundasturlartuzibko‘rsatilgan.

bilimlarnimustahkamlashuchun 30  
tanazariysavolvaamaliyko‘nikmavamalakalarnirivojlantrishuchun 10  
assistenttopshirig‘iberilgan.

Butopshiriqlarnibajarishmavzunimustahkamlashuchunxizmatqiladi.

✍️ **Kalitso‘zlar.** IDE, Visual C++, user interface, OYD, .NET, (plug-ins, add-ins, add-ons, abstrakt Windows Toolkit (AWT), MSDN, Visual Studio Interdev-ASP (Active Server Pages), GUI, ilova, loyiha, Windows Forms application, .NET Framework, Buyruqlarvaasboblarni, Form xususiyatlari, hodisalari, MessageBox.

📑 **Bilishshartbo‘lgantushunchalar.** Tiptushunchasi, sinfvasinfob‘ekti, funksiyavako‘rsatkich, kutubxonayaratishvadasturlashgaoiddastlabkitushunchalarhamdaC++tili niqo‘llabquvvatlovchimuhitdaishlashnibilishlozim.

📖 **Bilibolasiz.** Visual C++ muhitidasturlash, integrallashganishlabchiqarishmuhiti, toolkit, toolbox, Integrallashganmuhitlartarixi, Turbo muhitlar, GNU Emacs, Ob‘ektgayo‘naltirilgantil, Team Foundation Server (TFS), kengaytmalarniishlabchiqishmexanizmi (plug-ins, add-ins, add-ons), Yagonatillivako‘ptilliintegrallashganmuhitlar, NetBeans integrallashganmuhiti, Java dasturlashhaqida, Microsoft Visual C++, ASP texnologiyasi, Visual C++ da Windows ilovalarniyaratish, Muhitdamenyularvauskunalarpaneli, Buyruqlarvauskunalarpaneli, MS Visual Studio muhitininginteraktivtugmalari, Form xususiyatlarivahodisalari, MessageBox sinfining show funksiyalarinio‘rganishingizmumkin.

### REJA

1. Visual C++ muhitidasturlash.
2. Visual C++ da Windows ilovalarniyaratish

3. Visual C++ muhitidamenyularvauskunalarpaneli.

4. Form xususiyatlarivahodisalarini

## KIRISH

Bugungikundadasturlashmuhitlarivaturliko'payibbormoqda. Shuvaqtgacha console rejimdadasturlashni o'rganish bo'lsangiz kerak. Ammoso'ngi 35 yilichida foydalanuvchilarga oynalidasturlashmuhitlarida foydalanib kelinmoqda. Bunday dasturlarni yaratish uchun dasturlashning barcha asoslariga tayyanib, foydalanuvchi uchun qulay vizual dasturlash muhitlari yaratilgan. Ularda birinchi navbatda foydalanuvchining UI (user interface) yokidasturlashning foydalanuvchi bilan muloqot oynasini loyihalashtiriladi. Buning uchun tayyor sinflar (komponentalar) mavjud bo'lib, ularning xususiyatlar, funksiyalarivahodisalarini bilan ishlash shu juda qulay qilib, vizual foydalanish mumkin. Vizual foydalanish degan niko'ri turib ishlatishtirish degan ma'noni beradi. Bu vizual dasturlash bo'yicha ilk ma'lumotlarni keltiramiz. Deyarli barcha dasturlash tillari uchun vizual dasturlash muhitlari mavjud.

**Visual C++ muhitidasturlash.** Dasturlashni integrallashgan ishlab chiqarish muhitiningi lovalar itushunchasi, uning model xususiyatlari, rivojlanish tarixi, ularning eng ko'p ishlatilganlarini bilan tanishish lozim.

Integrallashgan ishlab chiqarish muhiti (Integrated development environment - IDE) - dasturlashni ishlab chiqarish hayot siklining barcha asosiy funksiyalarini qo'llab-quvvatlovchi umumiy interaktiv grafik qobig'iga egabo'lgan dasturlarni ishlab chiqarish va tekshirish vositalarini majmui, dastur matnini (kod) yozish va tahrirlash, kompilyatsiya, bajarish, xatolarni tuzatish, profilaktikavaboshqalarni bajarishimkoniyat beradi.

Integrallashgan muhitdan foydalanish dasturlashni ishlab chiqarishning mumkin bo'lgan yondashuvlaridan biridir. Shu bilan bir qatorda, UNIX tizimining oldingi anaviy yondashuvi, funktsionallik bilan bog'liq bo'lgan vositalar (toolkit, toolbox) to'plamidan foydalanishga asoslangan, lekin Integral interaktiv muhitga vababo'lgan (UNIX tizimining dastlabki versiyalarida) buyruqsatir rejimida (buyruqsatiri interfeysi) amalga oshirilgan. Albatta, Integrallashgan muhitdan foydalanish ishlab chiqaruvchi uchun ancha qulay bo



etishvaxatolarniqaytatuzatish.

Integralmuhitdanchiqmasdanvabubosqichlaroddiyfunksiya tugmachalaribilannazoratqilindivaharqandayalohidavositalarnianiqchaqi rishnitalabqilmaydi. Yuqorikompilyatsiya tezligiham Turbo muhitlarjozibadorsifatiaylandi. Turbo

Paskalningbirinchiversiyalaridakompilyatsiya birinchixatodanoldinamalgaoshirilganbo'lsa-da, barchaxatolarnitopishvatashxislashuchunavvalgisinio'ratganingizdansa 'ngkompilyatsiyaniqaytaishgatushirishingizkerak edi, ammobufunksiya tugmalaridanfoydalanibdarholamalgaoshirildi.

Turbo muhitlarto'liqikkilik Kodi, loyihakodikompilyatsiyasiuchunajralmas build mexanizmibor edi, shuningdek, make (F9) rejimi, klassik UNIX make foydafaoliyatigao'xshash, faqattahrirlangankodmodullar recompiling uchunamalgaoshirilgan.

Turbo Paskalmuhitidagi engmuhim yangilik - kirishtiliningob'ektgayo'naltirilgantushunchalar (sinf, ob'ekt) bilankengaytmasivamustaqilkompilyatsiya birligig'oyasinio'zidamujassamlashtirganbirlikkonstruksiyasi (moduli) bo'lgan. OYDuchunqo'llab - quvvatlashversiyadapaydobo'lgan. Keyinchalikbug'oyalar Windows platformasida, Borland integrallashganmuhitlarning yangiversiyalarida - Borland Pascal va Delphi (Paskalg'oyalariniOYDbilanbirgalikdarivojlantiruvchi Borland dasturlashtili) ishlabchiqildi.

Zamonaviyintegrallashganmuhitlarningasosiyxususiyatlari. Endiintegrallashgandasturiyta'minotishlabchiqishmuhitlariningasosiyxususiyatlarinumulashtiraylik. Ularningharbiri quyidagiqismlarga ega:

- Funksiya tugmalaridankengfoydalanganholdamuhitdanchiqmasdanboshqabarchak componentlarnichaqirishgaimkonberuvchi yagonainteraktivqobiq;

- Dasturfragmentlarini yozishvatahrirlashuchunmatnmuharriri;

- Qo'llab-quvvatlashtiziminiqirish, ya'niamalgaoshirilayotganmanbatilidankompilyatorvabittabajariladigankod (yuklash moduli) daob'ektikkilikkodlarininglinkerini'zichigaolganmanbakodidanloyihalar nituzish; linker operatsiontizimningstandartkomponentisifatida yokiushbumuhituchunmaxsusishlatiladi;

- Buyruqlarmajmui yordamidamuhitda debugging dasturlariuchun Debugger: birjoydao'ratish; berilgantartibdato'xtatish;

o'zgaruvchilar qiymatlarini tasavvur (yoki, pastdarajada, registr larini vaxotirahududlari) qilish;

-  
Integrallashgan muhitdagi zamonaviy matn muharrirlari kodni avtomatik bajarilishini ta'minlash (kodni tugatish), muharrir muhitdagi joriy yozilgan kodni sintaktik to'g'ri va uning davomini bo'lishim mumkinligini ta'lab qilinadi, misol uchun, yopilish qavsi yo'qligi, nuqta-vergul va usul nomlari, tafovutlar usul ma'lum birlashtirish bo'lgan bo'lsavaboshqalar.

-  
Integrallashgan muhitlarning zamonaviy versiyalarida quyidagi xususiyatlar ham qo'shilgan (komponentlar):

- Profiler -  
integrallashgan muhitni nazorat ostidagi dasturni bajarish natijasida olingan statistik ma'lumotlarni jamlash va ta'hlil qilish vositasi: protseduralarga (usullarga) chaqiriqlar soni, dasturni bajarish vaqtida ishlatiladigan xotirani miqdori va boshqalar.

- Refactoring -  
kodni takomillashtirish maqsadida muhitdagi dasturlarning tizimligi uruhlim o'zgarishlarini, ularning funksiyalarini ta'biqatub o'zgarishlarsiz amalga oshirish vositalari. Odatda, bunday harakatlarni o'z ichiga oladi, misol uchun, uning ta'rifini barcha foydalanish usulini o'zgartirish, uning argumentiga qo'shib, asinash try/catch blok oldindan e'tiborsiz istisno bilan band qilish va hokazo.

- Unit test generator - test modul uchun umumiy test yaratish uchun vosita (birliklari) - usullari yoki protseduralar - argument xususiyatlar turlini mumkin birikmalar bilan; odatda misollar integral Java muhitida JUnit vositasi va Visual Studio o'xshash NUnit vositasi bo'lib;

- Manba kodini boshqarish tizimi (source code control system) yoki mavjud versiya tizimlaridan biri (CVS, RCS, Mercurial, Visual SourceSafe va boshqalar) bilan muhitni integrallashgan qilish vositasi.) - dasturlarni qo'llab-quvvatlash daloyihakodifayllarining versiyasini nazorat qilishni qo'llab-quvvatlash;

- Dasturlarning jamoaviy rivojlanishini qo'llab-quvvatlash vositalari (jamoaviy ish) - dastur hayot siklibosqichlari (talab va xususiyatlar, dizayn, amalga oshirish, sinov), dasturchilar jamoasi a'zolarini o'rtasida rivojlanish vazifalarini taqsimlash,

loyihamenejeritomonidanvazifanibajarishninazoratqilish. Visual Studio muhitidabukomponentbirinchibo‘libTeam Foundation Server (TFS) debnomlanganva Visual Studio 2013 versiyasidanboshlab, ubulutliinterfeysifatidaamalgaoshiriladiva Visual Studio Online debataladi.

- Kodtahlilvositalari - uningsemantikto‘g‘riligi: odatdaijropaytidaaniqlanadiganxatolarningayrimenturlariningyo‘qligi, masalan, erishibbo‘lmaydigansharoitlar; zarurtekshirishlarvaxavfsizlikruxsatlariningyo‘qligivaboshqalar.

Professional

qiziqishlartufaylijudamuhimbo‘lganushbuxususiyatlarhaqidaqo‘shimcha ma‘lumotolishuchun "Visual Studio 2013 vauningishonchlivaxavfsizhisoblash (ishonchlihisoblash) uchunimkoniyatlari" gaqarang . Ushbuimkoniyatlar 2002 dashakllantirilganishonchlivaxavfsizhisoblash (trustworthy computing) sharoitivatamoyillarigamoskeladi. Microsoft korporatsiyasivautomonidanhayotgaizchiltatbiq etilmoqda. Zamonaviymuhitlargadasturdaginazoratoqimigrafiginingsiklomatiksoni, sinflarningilashishdarajasi (o‘zarobog‘liqligi) vashukabimurakkabliginitavsiflovchimetriklarbo‘yichakodnitahlilqilish vositalarihamkiradi;

- Hosilbo‘lganbinarkodnivizuallashtirishvositalari-usullar, o‘zgaruvchilar, ularningnomlarivaboshqalar. Misoluchun, Visual Studio muhitida, platforma .NET Common Intermediate til yagonaoraliq (ikkilik) kodnitasavvurqilishimkoniniberadi;

- Kod elementlari, nomlarinialmashtiruvchiobfuskatsionvositalar - sinflar, usullar, maydonlarvaboshqalar. Noma‘lumlarbilan, "tasodifiy", "chalkash" nomlariqiyin desompilel bo‘ldai, ikkilikkodnio‘rganishuchun, koddamavjud yangig‘oyalarnijoriyqilish, ruxsatsizistaganhakerlartomonidankodni "Hack"kaqarshihimoya qilishuchun, yokihujumlarnitashkillashtirish, zararlimaqsadlaruchununio‘rnatish;

- Standartkodvositalariasosidaturlixildasturiyloyihalarvaechimlarni yaratishniqo‘llab-quvvatlash; kengaytmalarniishlabchiqishmexanizmi (plug-ins, add-ins, add-ons). Zamonaviydasturiyta‘minotniishlabchiqishdaba‘zanturlixililovalarvavosi talar - konsol (oddiy) ilovalar, veb-ilovalarvaveb-xizmatlar, mobil ilovalar, bulutliilovalarvaboshqalarni yaratishkerak. Buturdagiharbirmanbakodifayllarma‘lumbirtuzilishnirivojlantirishnitala



bqiladi, shuningdek belgilash konfiguratsiya fayllar sifatida, misol uchun, kod xavfsizlik ruqsatini, veb-konfiguratsiyalar va hokazo.

Zamonaviy integrallashgan muhitlar manbako disha blonlarini taqdim etish va loyiha uchun zarur konfiguratsiya

fayllarini avtomatik ravishda ishlab chiqarish orqali turli xil loyihalarni yaratishni avtomatlashtiradi.

Dasturlashni tayyor kod shablonlarini ishlatmasdan tasavvur qilish qiyin, bu esa muqarrar ravishda xatolarga yo'l qo'yadi. Misol uchun,

uqo'ldaloyiha ning ajralmas qismibo'lgan faylni yaratishni unutish, yokikodba'zimuhim parchako'zdanyo'qotish juda oson (masalan,

moskelmaydiganko'rsatkicho aruchun, lekinungabog'langanko'rsatkichnita'minlash uchun emas).

Shuning uchun,

integrallashgan muhitlartomonidan loyihalarni turli turdagi qo'llab-quvvatlash, ayniqsa, muhim ahamiyatga ega. Bundantashqari,

zamonaviy integrallashgan muhitlardamumkin bo'lgan loyihalarni majmui kengaytiriladi, ya'ni, tuzuvchiz arur bo'lsa, muhitgaloyiha ning

yangituri niki ritingiz mumkin. Misol uchun, aspect yo'naltirilgan dasturlash vositasi amalga oshirishda Aspect.net Visual Studio

integrallashgan muhit kengaytmalarisifatida, tegishlikodshablonibilan loyiha - aspect yangituri joriy qilgan.

- UML (yagona model lashtirish tili)

model lashtirish tilidasturlar strukturasi ni model lashtirishni qo'llab-quvvatlash. UML ning zamonaviy versiyasi (2.x)

tilharxiliturdagidasturlarvategishli diagrammalarni qurish model larini ta'minlaydi. Bundantashqari, UML

dastur ishlab chiqish va ishlab chiquvchilaro'rtasida o'zaro hamkorlik uchun faoliyat model lari ishlab chiqish qo'llab-quvvatlaydi.

Zamonaviy integrallashgan muhitlar UML tilidan foydalanishni ikki yo'l bilan qo'llab-quvvatlaydi:

model ni vamanbako didantegishli diagrammani hosil qilish va, aksincha, ishlab chiqilgan model dan manbakodini (shablonini) hosil qilish.

**Yagona til va ko'ptili integrallashgan muhitlar.** Dastlab

yagona manbatilidasturlash uchun integrallashgan muhitlar ishlab chiqilgan (masalan, Turbo Paskal-Borland ning Paskal kengaytmasi dasturlash uchun).

- Shubilanbirga,

bunday monotilidagi integrallashgan muhitlarni ko'ptili muhitlarga aylantirish bo'yicha bosqichma-bosqich tendensiya yuz berdi,

chunkiturlitillardaloyihalarniishlabchiqishuchuno‘xshashtamoyillarvame xanizmlarqo‘llaniladivaturlitillarda

yozilgandasturbo‘laklarinibittaloyihadaishlatishhamba‘zanqulay.

Masalan, loyihada, shujumladan, yagonamaqsaduchun, masalan, C#, yanaqayta yozishuchun emas, balki, shuninguchun, (masalan, C) oldingitilda yozilganmeroskodniishlatishuchun yaratilgan.

- taniqli NetBeans integrallashganmuhitdastlab Java dasturlashuchun Pragada Charlz universitetitalabasiloyihasisifatida yaratilgan. Hozirgikunda NetBeans muhiti C va C++ daloyihalarniishlabchiqishgako‘makberuvchi C / C++development pack komponentasiniamalgaoshiruvchikuchliko‘ptilliintegrallashganmuhitgaa ylandi.

- Visual Studio - environment.net azaldanko‘ptillimuhitsifatida yaratilgan.

Ishlabchiquvchilarishlabchiqilganloyihaningtegishliqismlariuchun engqulaytilni

(tillarni)

tanlashlarivaloyihanitegishlitillardanbittaikkilikoraliq

SIL

kodigatuzishorqaliolinganikkilikkomponentlardan

(assemblies)

to‘plashlariuchun Microsoftningprinsipialsozlamalari. Keyinroq Visual studioningushbuqulayxususiyativauniqu‘llab-

quvvatlovchitillarmajmuasiga yaqindantanishibchiqamiz.

- Integrated development environment (IDE) - dasturishlabchiqishhayot sikliningbarchaasosiyvazifalariniqo‘llab-

quvvatlovchiumumiyinteraktivgrafikqobig‘iga

egabo‘lgandasturlarniishlabchiqishvaxatolarniqaytatahriqilishvositalari majmuidir.

Integratsiyalashgan Microsoft Visual Studio muhiti 1995 yilgakeilib,

ommagakengtaqaladi,

bunomostidamahsulotningbirinchiversiyasichiqarildi. Buoldin, 1990-yillarboshida, Microsoft yuqoridarajadagitillardadasturlashqo‘llab-quvvatlashuchunalohidamahsulotlarbilanbog‘liqbo‘lgan:

- Microsoft Visual C++ - Visual C++ dainteraktivdasturlashmuhiti, Microsoft tomonidanishlabchiqilganvaamalgaoshirilgan, C++ tiliningkengaytmasihisoblanadi. Visual C++ hali Visual Studio barchaversiyalaridadunyodagi engmashhurvakengtarqalgandasturlashtili;

- Visual Basic-Microsoft tomonidanishlabchiqilganvaamalgaoshirilganasosiytilningob’ektgayo‘na ltirilgankengaytmasi,

tilningsoddaligini ob'ektga o'zgartirish uchun birlashtirilgan. 1990-yillarda Visual Basic GUI qulaylik haqida gapirib va

boshqaruv dasturlarini ishlab chiqish uchun foydalanish afzal bo'lgan; - Microsoft Visual FoxPro-Visual FoxPro tilida interaktiv dasturlash muhiti - kichik firma Fox Software tomonidan dastlab Foxbase

nomi ostida ishlab chiqilgan protsessual dasturlash elementlari bilan ob'ektga o'zgartirish.

Ko'pgina foydalanuvchilar uchun butilning xususiyati SQL asosidagi dasturlash muhiti - kichik firma Fox Software tomonidan dastlab, xususan FoxPro da SQL so'rovlarini dasturlashga yordam bergan;

- Microsoft Visual Sourcesafe-Microsoft kod versiyasini auzorati zimit tomonidan ishlab chiqilgan, keyinchalik Visual Studio muhiti bilan integratsiyalangan.

Ushbu mahsulotlarning barchasi 1995-dan "birmuhitda" Visual Studio nom bilan birlashtirildi. Yangi muhitga 4.0 versiyasi deb nom berildi, ehtimol yuqoridagilardan biri muhitning komponentlarini bir nechta yillarda avomida alohida shakldagi qilingan.

Ushbu versiyada ham Visual Studio juda mashhur bo'ldi. Masalan, 1995-yilda yangi Java dasturlash texnologiyasini chiqargan Sun Microsystems Windows platformasi uchun GUI rivojlanishini qo'llab-quvvatlash uchun Visual Studio muhiti va platforma - mustaqil Java kutubxonalarini, abstrakt Windows Toolkit (AWT) ni amalga oshirish uchun unda amalga oshirilgan Visual C++ tilidan foydalangan.

1990-yillarda dasturchilari orasida vizual studiyalar muhiti mehbilando'stona "vizualka" nom bilan atalgan. Bubejiz emas: Visual Studio

dasturchilari rivojlanishida shonchli hamkorlangan, buuning interfeysining do'stonaligi va qulayligini tasdiqlaydi.

Taqqoslash uchun, o'shavaqtgacha kelib, o'sha firma SUN bir xil mashhur dasturiy muhitga ega emas edi. Java dasturlash uchun NetBeans muhiti biroz keyinroq - 1997 yilda, Charlz universiteti Praga dasturlash talabalari kichik guruhi ish natijasida paydobo'ldi. Visual Studio muhiti ning keng tarqalishida MSDN (Microsoft Developers ' Network)

gaobunabo'lishorqalimuhimrolo'ynadi, bu esa engso'nggi Microsoft dasturiyta'minotiniishlabchiqishgaimkon yaratdi.

**Vizual Studio 97 (5.0).** Visual Studio yangi (beshinchi) versiyasi, shuningdek, Visual Studio sifatidatanilgan 97 vakodlangan. Uning yangixususiyatlarivavositallari Visual Studio Interdev-ASP (Active Server Pages) texnologiyasiasosidainteraktivveb - saytlarniishlabchiqishvositasi - va Visual Java++ - "Microsoft"dan Java dasturlashmuhitibo'lgan.

ASP texnologiyasi - Microsoft kompaniyasining webdasturlashsohasidagijoyibrivojlanishi, 1994-1995-yillardapaydobo'lgan. Faolveb-sahifalarniishlabchiqishuchunmo'ljallangan-veb-so'rovlargajavobsifatidamijozbrauzerlariuchunHTML-sahifalarni yaratishuchunshablonlarga egabo'lgan. Buhaliko'psaytlardaishlatiladi. Texnologiya keyinchalikuningasosidaishlabchiqilgan ASP.NET, xususiyatlarigakengimkoniyatochibbergan.

Visual Java++ vositasigakelsak, umumanboshqaturdagihodisalarubilanbog'liq - ba'zida, afsuski, o'zdunyosidashungao'xshashnarsalarsodirbo'ladi, bu esakompaniyalarjudaraqobatbardoshbo'lishimuqarrarbo'lgan. Microsoft Sunning yangi Java texnologiyasibilantanihibchiqqanva Sun bilanraqobatdaungajavobsifatida Java tiliningo'zkengaytmasiniishlabchiqdivauni Visual Java++debnomlangan yangimuhitdaamalgaoshirdi. Biroq, afsuski, buharakatlarto'liq Sun tomonidanjoriyrasmiy Java amalgaoshirishartibibilanmoskelmadi. Shukungacha (Oracle hozir Java texnologiya egasibo'lsa-da, butartibaslidao'zgarmagan), Java nomirasmanro'yxatdano'tgansavdobelgisidir. Shuninguchun, yangi Java amalgaoshirishuchun, Java texnologiyasi egalikkompaniya litsenziya sotib, vaamalgaoshirishidoraviy Java standartlargajavobtekshirishuchun Java moslashuv Kit (JCK) debnomlangan yangi Java amalgaoshirishuchuntil, kutubxonalarvavertualmashinaamalgaoshirishuchuntestlarmaxsusmajmu inio'tishikerakbo'lgan. Faqatbutestlarninghammasito'liq 100% o'tsa, yangiamalgaoshirish "Java"debnomlanishhuquqinioladi. Afsuski, Microsoft tomonidan Visual Java++ niamalgaoshirishuchun Visual Java++ tizimidaamalgaoshirilgantikengaytmalaritufayli JCK testlaridano'tmadi. Buish, afsuski, ikkalakompaniya uchunham, Sun va Microsoft o'rtasidagiuzoqmuddatlihuquqiyjangda, Sun

tomonidan qoʻlgakiritilgan, natijada Microsoft mahsuloti Vj++ deb nomlangan (Java nomini ochiq-oydin eslatib oʻtmasdan). Bunday hodisalarni muqarrar boʻlsa-da, juda yoqimsiz, chunki har ikki tomon ham, baʼzan bir nechta yillarda vomi damashhur texnika mutaxassislar, bularning hammasiga koʻp vaqt, kuch va sogʻliqsarflashadi. Biroq, keyinchalik, 2004 yilda, Sun va Microsoft 10 yil muddat gʻai intellektual mulk boʻyicha mislik oʻrilmagan shartnomatuzish, bir shartnomakelgan.

Shunday qilib, Visual Studio 5.0 Visual C++, Visual Basic va VJ++ da integrallashgan dasturlash muhit boʻlib, u ASP texnologiyasi yordamida interaktiv Websaytlarni ishlab chiqish uchun Visual Studio Interdev vositasini ham oʻz ichiga oladi.

**Visual Studio 6.0 (1998).** Visual Studio keyingi versiyasi (6.0, kodlangan Aspen) 1998 yilda chop etildi. Visual C++, Visual Basic, vj++ va Interdev, (maksimal versiyada - Visual Studio Enterprise Edition) - allaqachon maʼlum komponentlarini yangi versiyalar bilan bir qatorda, quyidagi

yangi komponentlar ham vizual modellashtirish va dasturlashni taʼhlil qilishni qoʻllab-quvvatlash uchun kiritilgan:

1. Application Performance Explorer
2. Automation Manager
3. Microsoft Visual Modeler
4. RemAuto Connection Manager
5. Visual Studio Analyzer

Visual Studio 6.0 dasturlar bilan ishlayotgan uchun bittakoʻptilli virtual platforma oʻz ichiga oladi integrallashgan muhit, yangi versiyasini yaratish uchun asossifatida Microsoft tomonidan ishlatiladigan versiyasi - .NET.

**Visual Studio.net (2002).** Visual Studio 7.0 versiyasi, shuningdek, Visual sifatida tanilgan Studio.net vakodlangan Rainier, joriy etildi. Ushbu versiyada eng muhim yangilik boshqariladigan kod, umumiy tilni fratauzilmasi (SLI), umumiy oraliq til (SIL), umumiy turdagitizim (SP) asosida toʻliq turdagi nazorat va xavfsizlik bilan ish oʻnchil va xavfsiz dasturlashni taʼminlaydigan koʻptilli platformani amalga oshirishdir. .Net platformada barcha kompilyatorlar manba kodini tegishli til dantarjima qiladi (C#, C++, Visual Basic va hokazo.) bitta oraliq SIL kodining ikkilik formatiga - VMning postfiks yozuvi instructions.net -

Virtual ijrotizimi (VES). Taqdim etilgan: .Net ichidagimoslik-turlitillardantuzilgankodlarningilovalari; yagona metadata asosidagiturdaginazoratvaxavfsizlikmexanizmi; yagonako'ptilliistisnomuomalamexanizmio'zichigaoladi. Visual Studio.NET 2002 versiyasito'rtversiyalaridachop etildi (nashrlar):

*Academic, Professional, Enterprise Developer, Enterprise Architect*

Visual Studio muhitidaqo'llab-quvvatlanadigantillardahaqiqatdanhamajoyibo'zgarishvaqo'shimchalaramalgaoshirildi.

Avvalo, C# -yangidasturlashtiliishlabchiqildivaamalgaoshirildi, hozirgikunda engkuchli, zamonaviyvato'liq dasturlashtili.

Visual C++ va Visual Basicdamuhimkengaytmalaramalgaoshirildi. Aslidasintaksisdagiayrimfarqlargaqaramay, butillardeyarlio'xshashxususiyatlarnita'minlab, C#tiliga "teng" bo'libkelgan. Yangi .NET -mosversiyasi Visual C++ muvaffaqdeyiladi, C++ va yangi Visual Basic SOF-mosversiyasi Visual Basic.net deyiladi.

**Visual Studio.net 2003.**Keyingiyili, 2003, Visual Studio yangiversiyasi released.net kodinomi Everett ostidaishlabchiqardi. Birbutunsifatidamahsulotishlabchiqishnuqtainazaridan, buversiya muhimo'zgarishlarqilinmadi: yangiversiyasiningasosiy maqsadiversiyasi .net Framework 1.1 uchun yangilashbilanta'minlash edi. biroq, Visual Studio 2003 judamuhim yangilik mobil qurilmalar .net Compact Framework uchunversiya .net da amalgaoshirishva Visual Studio vositalari yordamida mobil qurilmalaruchundasturlarishlabchiqishuchunqo'llab-quvvatlash edi.

Oldingiversiyasikabi, Visual Studio 2003 to'rt nashrlardachopetildi: *Academic, Professional, Enterprise Developer, Enterprise Architect*. Buntantashqari, "maksimal" versiyada - Enterprise arxitektor - Visual Studio 2003 dasturarxitekturasinivasavvur UML diagrammalaruchunqo'llab - quvvatlash, shujumladan, boshqamashhur mahsulot -Microsoft Visio tarzidadasturiymodellashtirishvositalariqo'llab-quvvatlaydi.

Visual Studio.net 2005. Visual Studio 2005 integralashganmuhit (kodlangan Whidbey) keyingiversiyasi 2005 yilda e'lonqilindi.

Versiyaningasosiy yangiliklariparametrlima'lumotturlari (generics) gategishli. Ular C# 2.0 tilidagiversiyadava .net Framework 2.0 daamalgaoshirildi. Bubirvaqtningo'zidashungao'xshashxususiyatlari

Java 1.5 da amalga oshirildi, deb ta'kidlash qiziq, lekin generics mexanizmi Java nisbatan, yanada qulay, moslashuvchan va kengaytirilgan versiyada amalga oshiriladi .net va C# platforma asoslangan ta'kidlash lozim.

Manfaatdorlikni yaxshilash yanabir guruh ASP.NET -. NET platformasi gaxoskodning ishonchli ligi va xavfsizligi bilan birgalikda, veb-sahifalarni amalga oshirish va faollashtirish uchun mo'ljallangan ASP texnologiyasini ishlab chiqildi. Visual Studio 2005 versiyasi qo'shimchaloyihalaritushunchasini qo'llab-quvvatlash uchun birinchi marta ASP.NET Webxizmatlarini amalga oshirildi. NET dagi webdasturlash vositalari ushbu platforma va umumiy Webdasturlash talabari uchun alohida e'tiborgaloyiqligini unutmang. Faqatularni o'rganish va ishlatish uchun har bir dasturchi platformani yaxshilash kerak .NET va amalda o'zi ishlatish kerak.

Aspect yo'naltirilgan dasturlash tizimi rivojlantirish jamoasi uchun Aspect.net Visual Studio 2005 versiyasini alohida ro'lni o'ynadi. Visual Studio muhitining ushbu versiyasi uchun plug-in deb nomlangan Aspect.net Framework-integrallashgan muhit o'zining vizual uslubiga o'xshash bir uslubdagi jihatlarin iboshqarish uchun GUIdir. Interaktiv xususiyatlar va qulaylik Aspect.net foydalanuvchilartomonidan yuqorib aholandi va tizimning ushbu versiyasida Aspect.net dunyoning 26 mamlakatidatartalgan.

Vizual *plug-in* sifatida amalga oshirilgan boshqataniqliloyiha Studio.net 2005 loyihasini hisoblanadi. Knowledge.net - bilim vakillik orqali C# tilining kengaytmasi (ramkalar, qoida silsilasi, ontologies), qaysi bilim foydalanish bilan an'anaviy C # dasturlash uslubini birlashtirib beradi. Knowledge.net talabalar va magistrantlartomonidan bilimlardan foydalanadigan aqlli echimlarni ishlab chiqish uchun ishlatiladi.

Visual Studio.net 2008. 2008 yilda Visual Studio ning navbatdagi versiyasi ishlab chiqarildi. Microsoft kompaniyasining Visual Studio muhitining ikki versiya raqami Visual Studio 9 hisoblanadi. Ushbu raqamlar yoddatutish uchun foydalidir, chunki Microsoft xodimlariko'pincha Visual Studio rivojlanishini tasvirlashda ushbu ikki versiya raqamlaridan foydalanadilar (masalan, bloglarda). Visual Studio 2008 versiyasi Windows Vista uchun mo'ljallangan. Bu boshqamashhur Microsoft Office 2007 mahsulot versiyasiga mos keladigan ilg'or veb-dasturlash qo'llab-

quvvatlash vositalari o'z ichiga oladi. Versiya .NET Framework Visual Studio 2008 versiyasi uchun .NET Framework 3.5 kateqishli hisoblanadi.

VS 2008 versiyasi GUI-ni qo'llab-quvvatlash uchun Windows Presentation Foundation (WPF) API-dan foydalanadigan ilovalar uchun yangi VS 2008 debugger multithreaded dasturlardaxatolarni qaytatahlilash uchun yordam beradi.

VS 2008 versiyasi birinchi marta J# tilini qo'llab-quvvatlamadi, Microsoft Java tilining kengaytmasi, yuqorida aytib o'tilgan. Buning o'rniga foydalanuvchilarga Java tilini muhitidagi ilovalarni .Net muhitiga ko'chirish uchun muhitga kiritilgan Java Tilni o'zgartirgich Assistant (JLCA) vositasi taklif etildi.

Version VS 2008 hozirda Microsoft ichki mahsulotni qo'llab - quvvatlaydi Visual Studio muhit so'nggi versiyasi - Feniks, derleyiciler va til protsessorlari optimallashtirish rivojlantirish uchun bir vosita. Bu Microsoft tadqiqot va Microsoft mahsulot jamoasi qo'shma mahsulot hisoblanadi.

Biz tizimida maqsad ilovalar joylash jihatlari ni amalga oshirish uchun Feniks foydalanish Aspect.net, Feniks qulay yuqorida rajali ichki vakillik bor, chunki Feniks yuqorida rajali IR (HIR), jihatidan undasturo'zgarishlarni amalga oshirish uchun qulay. Feniks ham turlitizimida amalga oshiradi, bu ichki vakillik tomonidan ikkilik build .net ishlab chiqarish uchun qulay vositasini o'z ichiga oladi Aspect.net unda jihatlarni joylangan dan keyin olingan binar montaj hosil qilish. Feniks hozirda Microsoft tadqiqot Midori loyihasi uchun ishlatiladi, boshqariladigan kod muhit .net butunlay yozilgan yangi operatsion tizimi

**Visual Studio.net 2010.** Aprel oyida 2010, Visual Studio versiyasi edi. Sankt-Peterburgda Visual Studio 2010 ishga tushirilishi marosimida bir qator qiziqarli taqdimotlar va namoyishlar bo'lib o'tgan, jumladan, Visual Studio kompaniyasining takchil ishlab chiquvchisi, Microsoft menejeri Brayan Garri tomonidan hisobot taqdim etildi. Tizimni yoqishni taklif qilib Aspect.net studiyaning kelajakdagi versiyalarini etkazib berishda, chunki AOP tizimimiz taniqli va uning imkoniyatlari, uslub va AOP ni keng tarqatish uchun Visual Studio bilan integrallashganda rasigamoskeladimuqobil kundalik texnologiyalar dan bir sifatida (ubiquitous technologies), Visual Studio-da qo'llab-quvvatlanadigan texnologiyalarning eng keng to'plamiga kiritilgan. Shuning uchun, ayniqsa, zamonaviy versiyadan beri amal qiladi Aspect.net



Visual Studio so'nggi versiyasidan mos 2013. Tizim tahrirlash Aspect.net "Visual Studio 2013 da aspektli dasturlashdan (AOP) foydalanish" dako'rib chiqiladi.

Visual Studio (ichki versiyaraqami-10, kod nomi-Dev10) dagi eng muhim yangilik qo'llab - quvvatlanadigan tillarning "jentelmenlar to'plami" dabosh qatilning paydobo'lishi - f# funksional tili. Til muallifi Microsoft Research, Kembrij, buyuk Britaniyadan don Syme hisoblanadi. .net Framework va Visual Studioda F# tilini funksional dasturlash imkoniyatlarini ob'ektga yo'naltirilgan dasturlash, parallel dasturlash va boshqabarcha xususiyatlar bilan integratsiya vositalari bilan birlashtiradi. Bu, ayniqsa, yosh dasturchilar orasida ko'plab tarafdorlarni topgan juda istiqbolli tildir.

Dastlab, Visual Studio 2010 Phoenix mahsulotini va unga asoslangan optimallashtiruvchi kompilyatorni Visual C++ tilidan o'z ichiga olish rejalashtirilgan bo'lib, hosil qilingan kikkodni bajarishda Visual C++ kompilyatorining mavjud, avvalgi versiyasidan o'zib ketish kerak edi. Biroq, afsuski, bu rejalarga amalga oshirilmadi - Feniks jamoasi Visual Studio

loyiha sish boshqaruvi tomonidan belgilangan muddatlargajavob bermadi.

Visual Studio 2010 Ultimate - - "maksimal" versiyada UML diagrammalari yordamida loyihani tuzilishini aksettirish uchun vositalari, shuningdek, qulay skipped test guruhlar, loyihani manbako'zga o'tirishda sirini baholash uchun Testta'sirli komponent bor, chunki keraksiz o'tishtirishni o'z ichiga olish imkonini beradi. Bundan tashqari, debug tarixi va uning to'liq holatini eslash imkonini beradi. Intel trace bir debugger bor, shu jumladan multithreading tarixi.

Visual Studio 2010 bulunish o'z ichiga qo'llab-quvvatlaydi. Buning uchun Microsoft Azure loyihalarining yangi turlari qo'shildi. Ularni ishlatish uchun Visual Studio 2010-Microsoft Azure SDK uchun maxsus vosita o'rnatish kerak.

**Visual Studio 2012.** 2012-yil avgust oyida Visual Studio 2012 kodli Metro mavzusining versiyasi chiqarildi. Bu foydalanuvchi interfeysi yaxshilandi, yangi loyihani o'z ichiga olish vositasi ishlab chiqilgan va kod fragmentlarini ta'kidlab uchun ranglar qo'llab-quvvatlashi yaxshilandi. Microsoft ta'kidlaganidek, Visual Studio 2012 uchun manbako'zining umumiy hajmi 50 millionga etdi.

Shundayqilib, Microsoft Visual Studio integrated development muhitifaolinnovasionrivojlanishningnoyobnamunasidirvadasturiyta‘min otniishlabchiqishsohasidayangig‘oyalarvavositalariningbirbutunbirlashuv inio‘zidamujassamlashtiradi. 1995 yildanbuyon 25 yilmobaynida 19 taversiyachi qarilganbo‘lib, ularningharbiridayangidasturlashtillari, buyruq-satrdasturlariniishlabchiqishvositalarivaboshqako‘plabyangiliklarmavjud

Ba‘zibir kalitso‘zlarnibilibolishlozim:

9.1-jadval. Kalitso‘zlar.

.NET	-ISO xalqarostandartlaribilanstandartlashtirilganvabirnecha implementations egabo‘lganob’ektgayo‘naltirilgankop’tillidasturlashplatforma, - Microsoft.net, Microsoft tomonidanishlabchiqilgan
C#	-Microsoft kompaniyasitomonidanishlabchiqilganvafaolishlabchiqilganengyanngiob’ektgayo‘naltirilgandasturlashtili. NETplatformasiuchunasosiydasturlashtili
Visual Basic	- Microsoft tomonidanishlabchiqilganvaishlabchiqilganasosiytilningob’ektgayo‘naltirilgankengaytmalidasturlashtili
Visual C++	- dasturlashtili, Microsoft tomonidanishlabchiqilganvaishlabchiqilgan C++ tiliningkengaytmasi
Visual Fox Pro	-Microsoft tomonidanishlabchiqilganob’ektgayo‘naltirilgandasturlashtillaridambiri, SQL so‘rovlartilibilanmosma‘lumotlarbazalarigakirishuchunvositalarnio‘zichigaolgan
Visual Studio	- Microsoft kompleksrivojlantirilganintegrallashganmuhit.

**Visual C++ da Windows ilovalarniyaratish.** Windows ilovalarniyaratishgakirishdanoldinmuhitnio‘rnatishvauningimkoniyatlari gato‘xtalibo‘tamiz. Visual Studio standart o‘rnatishsida C++ tilinirivojlantirishuchunzarurelementlarniqo‘shish/o‘chirishimkoninibera

di. Masalan, an'anaviy C++ dasturlarini ishlab chiqish, C++ da mobil ilovalarni ishlab chiqish, C++ da Linux muhiti uchun vositalarni ishlab chiqish, C++ tilida o'yin ishlab chiqish va boshqa yanabir qatorim koniyatlarni beradi.

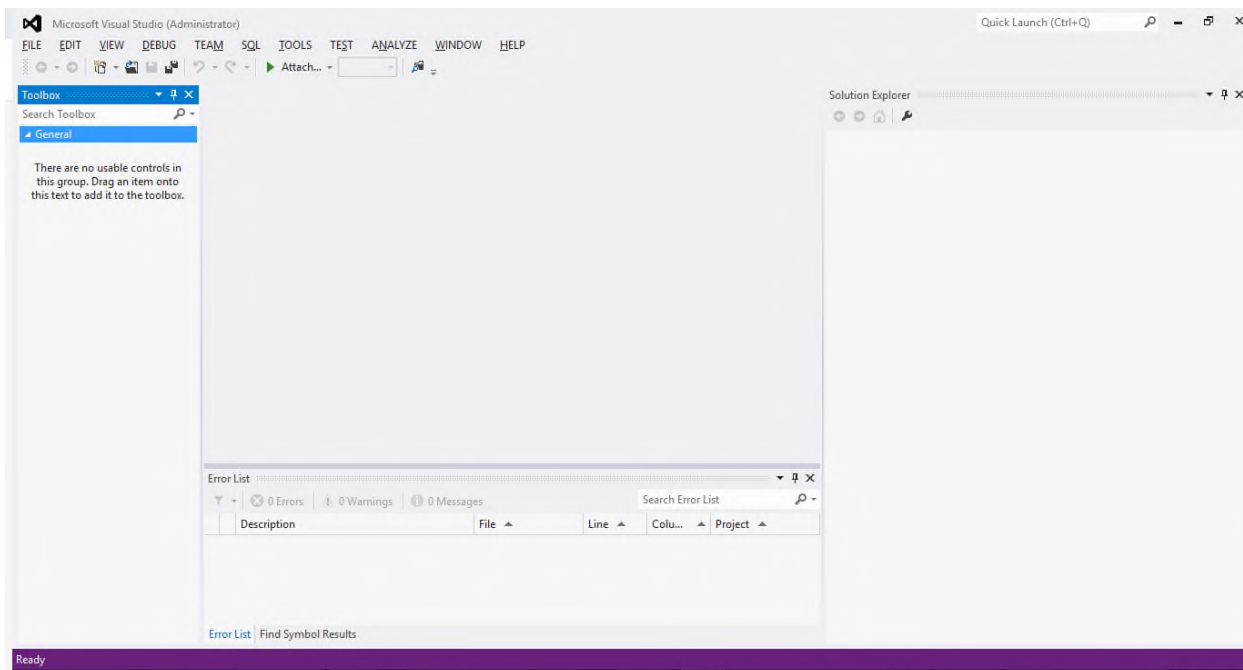
Visual C++ da Windows ilovalarni yaratish uchun avvalo, MS Visual Studio o'rnatish kerak. Kompyuterning texnik koniyatlaridankelib chiqqan holda, mos versiyatanlanadi. Mazkur versiyani o'rnatish, internet manbalaridan olingan bo'lsa, Odatda oddiy muhitlarni o'rnatiladi.

MS Visual Studio ning hamma versiyalarida Windows ilovalarni turlicha yaratish mumkin.

Ammo eng qulay va umumiy foydalanuvchi uchun mos bo'lgan interfeysni o'rnatish uchun har xil versiyalarga har xil sozlashlarni bajarish kerak. Masalan, MS Visual Studio 2012 da Windows ilovalarni yaratish uchun, Windows oynasitay yoki komponentni o'rnatish uchun, 9.1-rasmda keltirilgan fayllarni VCvcprojectsvcNET/ VC++ papkasiga nusxalash lozim.

Name	Date modified	Type	Size
backup	9/13/2012 4:50 PM	File folder	
MC++AppWiz.ico	6/11/2012 8:41 PM	Icon	10 KB
MC++AppWiz.vsz	12/12/2011 1:30 PM	Visual Studio Wiza...	1 KB
MC++ClassLib.ico	6/11/2012 8:41 PM	Icon	10 KB
MC++ClassLib.vsz	12/12/2011 1:30 PM	Visual Studio Wiza...	1 KB
MC++EmptyProj.ico	6/11/2012 8:41 PM	Icon	10 KB
MC++EmptyProj.vsz	12/12/2011 1:30 PM	Visual Studio Wiza...	1 KB
MC++WinApp.ico	6/11/2012 8:41 PM	Icon	10 KB
MC++WinApp.vsz	9/2/2012 5:08 PM	Visual Studio Wiza...	1 KB
MC++WinCtrlLib.ico	9/2/2012 11:15 PM	Icon	10 KB
MC++WinCtrlLib.vsz	9/2/2012 5:09 PM	Visual Studio Wiza...	1 KB
vcNET.vmdir	9/2/2012 10:20 PM	VSDIR File	1 KB

9.1-rasm. Windows ilovalarni yaratish uchun zaruriy fayllar. Shunday qilib, MS Visual Studio ni shartlashing, 9.2-rasmdagidek, integrallashgan muhit oynasini o'rnatish kerak.



## 9.2-rasm. Integrallashgan muhitoyyasi.

Muhitoyyasi 5 qismdan iborat. Ular quyidagilar:

1. Sarlavhasatri. Bundatizimlimenyu (chaptomondagitizimbelgisi), tizimvajoriyproektnomi, tizimlitugmachalar (yig'ishtirish, oynao'lchaminio'zgartirish, oynaniyopish)

2. Muhitningmenyusi.

3. Muhitninguskunalarpaleni (foydalanuvchitomonidankerakliuskunalarbilanto'ldirilishimumkin).

4. Muhitningishchimaydoni (Foydalanuvchiningjoriyproektuchunzarurkomponentalarvaularniboshqarish, kuzatishimkoniyatlarimavjud)

5. Muhitningholatsatri.

Yangiproektyaratilishibilanproektgamoskeraklivajoriyqilingankomponentalarfaollashadi. Windows ilovalarniyaratishuchunbirnechtavarinatlarmavjud.

Ularningengko'ptarqalganvariantimenyudanfoydalanibiyaratishdir.

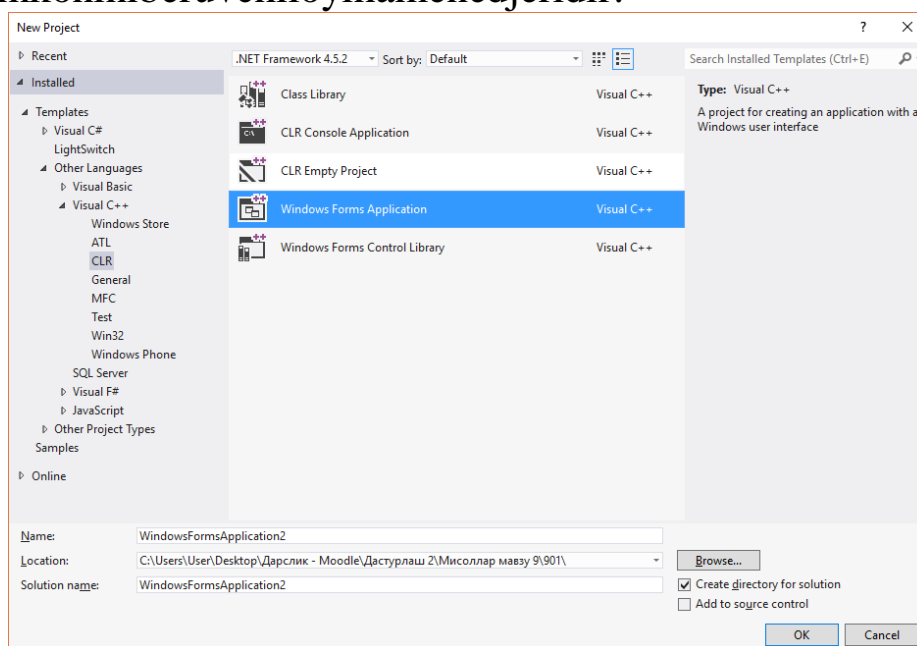
Windows ilovalarniyaratishuchun [File → New → Project] ketmaketligiyoki [Ctrl+Shift+N] tugmachalarmajmuasinibosishyetarli. Ekranga [new Project] nomlimuloqotoynasihosilbo'ladi (9.3-rasmgaqarang).

Birinchimarta MS Visual Studio ish gatushirgandaasosiytilnitlanlashniso'raydi.

AgarC++tilinitanlaganbo'lsangiz, avtomatikshutildayaratilishimumkinbo'lganloyihalarro'yxatihosilbo'ladi. Agaroldinboshqatilnitanlaganbo'lsangiz, [Other Languages] bandidan

Visual C++ tilinitanlashingizlozim. Visual C++ tilasosidayaratishmumkinbo‘lgankonseptualloyihalarro‘yxatichiqadivab ularningo‘zibirnechtaloyihalgabo‘lnadi (9.3-rasmningchaptomonidaqarang).

Visual C++ da Windows ilovalarniyaratishuchunavvalo [CLR] loyihamedjeritanlanadi. CLR - Common Language Runtime - birnechaqo‘llab-quvvatlanadigantillardaharqandayyozilgandasturlarijrosiniboshqaradidas turlashbo‘lib, ularnitillariharqandayyozilganumumiyob’ektgayo‘naltirilgansinflargaal mashishimkoniniberuvchiloyihamedjeridir.



9.3-rasm. Yangiloyihayaratishoynasi.

[CLR] loyihamedjeribirnechtaloyihalaryaratishimkoniniberadi:

1. Class Library – boshqaproektlargafoydalanishuchun CLR sinflarniyaratishimkoniniberadi.
2. CLR Console Application – CLR asosidakonsolloyihayaratishuchunishlatiladi.
3. CLR Empty Project - yangihechnimayozilmaganloyixalarasosidalokalloyihalarniyaratishuchun ishlatiladi.
4. Windows Forms application – Windows oynaliloyihalarniyaratishimkoniniberadi.
5. Windows Forms Control Library - Windows ilovalarniboshqarishuchunfoydalaniladigankutubxonalarniyaratishimkoniniberadi.

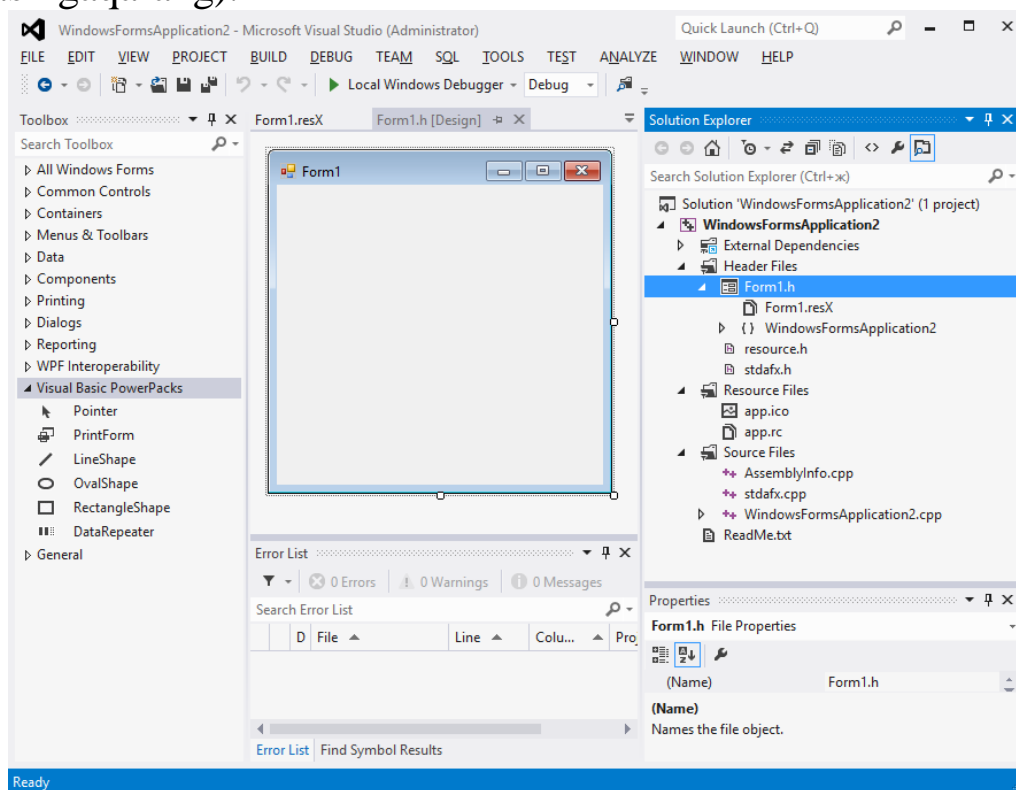
Harbiroyihano‘zo‘rnidayaratishfoydalanishmuzimdirvaharbirinitan laganingizdaularningizohioynaningo‘ngtomonidachiqadi.

[CLR] loyihamederidanfoydalanibloyihayaratishda .NET Framework ningvariantihammuhimrolo‘ynadi. Shuninguchun .NET Framework

larningimkoniyatlaribilanoldindanyaxshilabtanishibchiqishlozim. Harbir .NET Framework ningo‘ziningimkoniyatlariborvashuningdek Visual C++ da Windows ilovalarniyaratishuchun .NET Framework 4.5 yoki 4.5.2 variantlaridanfoydalanishnitavsiyaqilamiz. .NET Framework nitanlashloyihaturinitanlashoynasiningustidajoylashgan.

Loyihayaratishoynasini pastkiqismidaloyixanomi, joylashganjoyini, loyihaningishchipakkasiningnomiko‘rsatiladi.

Kerakliamallarnibajargandanso‘ng [ok] tugmasinibossangiz, birnechtaavtomatiksozlashdanso‘ngloyihaningilkko‘rinishihosilbo‘ladi (9.4-rasmgaqarang).



9.4-rasm. Oynaloyihaningdastlabkiko‘rinishi.

Oldinaytibo‘tilgandek, muhitningishchioynasinio‘zgardi. Uniodatdafoydalanuvchio‘ziningxoxshibo‘yichajoylashtirishimumkin.

Engaosiyelementlardanbiribu – Solition Explorer bo‘libhisoblanadi. Buloyihapapkasiniboshqarishuchunqulayqilibyaratilgan.

Bittalshchipakdabirnechtaloyihabo‘lishimumkin. Shuninguchun, uyerdanloyihanominitanlashkerakbo‘ladivaloyihauchunnkerakbo‘limlarr o‘yxatichiqadi:

1. External Dependencis – tashqi, tizimlikera klikutubxona, sinflar joylashadi.

2. Header Files – kutubxonalar fayllar.

3. Resource Files – loyixaresurs fayllari.

4. Source Files – dasturmanbakodlari yozilgan fayllar.

5. ReadMe – matnli fayl bo‘lib, unda loyiha uchun izohlarni yozib qolirishingiz mumkin.

Standart loyiha yaratilgan dastur yaratidgan joriy fayllarni o‘zgartirmang, bo‘lmas loyiha ishlamasligi mumkin.

Odatda juda ko‘p foydalanuvchilar loyihalarda vizuallashni yoqtiradilar, buning uchun kutubxonalar fayllarini majmasidan \*.h kengaytmali fayllardan foydalaniladi.

Unibos gandaloyiha ning kerakli oynasi vizual ko‘rinishda hosil bo‘ladi.

Solition Explorer dani kerakli faylni tanlab, ustiga sichqoncha ni bossangiz shu faylni tahrirlashimkoniyati hosil bo‘ladi.

Agar oldindan nomlar fazosi, sinflar, shablonlar bilan ishlagan bo‘lsangiz, ularning ham algoritmlari va funksiyalarini bilantani shishim mumkin.

Buning uchun kerakli chekni bosish kerak bo‘ladi.

Solition Explorer dani kerakli ob‘ekt (fayl) ni tanlaganigizda uning xususiyatlari (properties) va funksiyalari (events) larni sozlashim mumkin. Buning uchun dasturchiga properties oynasi yordam beradi. Ob‘ekt gamosholda ishlaydi.

Ishchi maydonning chaptomonida Solition Explorer dani tanlangan ob‘ekt uchun qurish, integrallashtirishim mumkin bo‘lgan uskunalar, metodlar va boshqa zaruriy xarakterga ega elementlar hosil bo‘ladi.

Ishchi maydonning aosiymarkazidalo yiha ni tahrirlashishlarini bajariladiv auning pastki qismidaxatolar, xabarlar, ogohlantirishlar va boshqa dasturchi uchun kerakli barcha yordamchi oynalar ni joylashtirishim mumkin. Shuningbilingki, ishchi oynani dasturchi o‘ziga moslab tayyorlab shablon ko‘rinishida qilib olishim mumkin.

**Muhitdamenyular va uskunalarning paneli.** MS Visual Studio muhitining oynasi boshqa dasturlarning menyularini kabijoylashgan bo‘lib, 13 tadan iborat.

Uskunalarning paleni dasturchining xoxishi bilan almashib turishim mumkin. Shuning uchun faqat menyularning vazifasi buyruqlarini o‘xtalamiz. Odatdamenyugasi sichqonchabilan murojaat qilinadi.

Agarsichqonchaishlamayqolsa, [Alt] tugmasinibosganda, menyunomlarigatagigachiqlanganharfpaydobo‘ladi. [Alt] tugmasinishuharfbilanbossangizmenyugamurojaatqilasiz.

[FILE] –

muhitdaloyihaustidaglobalhodisalar nibajarishuchunmo‘ljallanganasosiy menyulardanhisoblanadi. Undaquyidagiasosiybuyruqlarbor: yangiloyihayaratish, loyihani ochish, loyihabirorbiryangiloyihaqo‘shish, faylniyopish, loyihaniyopish, loyihafaylinivaloyihanito‘liqsaqlash, boshqaloyihalarniimprotqilish, oldinishlatilganloyihafayllari, muhitdanchiqish.

[EDIT] - muhitdatahrirlashishalar nibajarishgamo‘ljallangan.

[VIEW] – muhitniko‘rnishi, uskunalarnijoylashtirish, keraklisigamurojaatqilishgamo‘ljallangan.

[PROJECT] – loyihauchunelementlarniqo‘shish, boshqarish, sozlamalarnitayyorlashuchunishlatiladi.

[BUILD] – loyihaniqurish, qaytaqurish, tozalash, paramertlarinisozlashvaboshqaishlargamo‘ljallangan.

[DEBUG] – loyihanitekshirish, qadamba- qadamtekshirish, ishlatish, ishlashinitekshirishkabiamallarnibajarishuchunmo‘ljallangan.

[TEAM] – serverlargaulanishuchunmo‘ljallangan

[SQL] –

ma’lumotlarbazasiniboshqarishninglingvistika‘minotibilanishlashgaqaratilgan.

[FORMAT] –

muhitdaturliformatlashishlar nibajarishgamo‘ljallanganbuyruqlaribor.

[TOOLS] –

muhitdaturlisozlamalarniamalgaoshirishuchunqaratilganmenyu.

[TEST] –

loyihatestlarvahisobotlarniolishuchunbuyruqlarmajmuasijoylashganmenyu.

[ANALYZE] –

loiyhaningelementlariningo‘zarointegrallashganholatdaishlashinitahlilqilishbuyruqlari

[WINDOW] – muhitdaoynalarnijoylashtirish, boshqarishbuyruqlari

[HELP] –

muhitbo‘yichayordam, turlirasmiyyordammuhitlaargachiqishimkoniyatlariniberadiganbuyruqlar joylashgan.

**Buyruqlarvauskunalarpaneli.** Menyularvauskunalarpanelifoydalanuvchilar vspackage buyruqlargakirishuchunbiryo‘ldir. Buyruqlar-



xujjatnichopetish,  
ko‘rinishniyangilashyoki yangifaylyaratishkabi vazifalarnibajaruvchifunk  
siyalardir.

Menyularvauskunalarpanelifoydalanuvchilargabuyruqlarnitaqdimetishni  
ngqulaygrafikusullariega. Odatda,

tegishlibuyruqlarbirxilMenyudayokibirxiluskunalarpanelidaguruhlanadi.

Menyularodatdaintegrallashganmuhati (IDE)

yokiuskunalaroynasiningyuqoriqismidaketma-

ketguruhlenganbittaso‘zlisatrlarsifatidako‘rsatiladi.

Sichqonchaningo‘ngtugmasinibosgandamenyunihamko‘rsatishmumkin

— buholdaularkontekstmenyulardebataladi.

Bosgandanso‘ngmenyuochiladivabiryokibirnechabuyruqlarniko‘rsatadi.

Birorbuyruqbosilgandanso‘ngvazifalarbajarilishiyoki qo‘shimchabuyruql  
arbilan submenus ochilishimumkin. Ba‘ziomabopmenyunomlarifayl,  
ko‘rish, vaoyna, [Object window] bor.

Uskunalarpaneli - Toolbars

odatdatugmalari vaboshqaelementlardaniboratsatrlardajoylashgan,

masalan, ro‘yxatlarjamlanmasivamenyu controllers kabi.

Uskunalarpaneliboshqaruviningbuyruqbilanbog‘liqbarchaelementlarinita  
‘minlaydi. Uskunalarpanelitugmasinibosish,

ugabog‘liqbuyruqfaollashtiradi.

Uskunalarpanelitugmalariodatdachopetishbuyrug‘iuchun printer

kabiasosiybuyruqlarnitaklifqiluvchibelgilarnio‘zichigaoladi.

Ochiladiganro‘yxatdaro‘yxatdagiharbirelementboshqabuyruqbilanbog‘li  
q. Menyutekshiruvi - bugibriddir,

undanazoratningbirqismi uskunalarpanelitugmasivaikkinchisibosilgandaq  
o‘shimchabuyruqlarniko‘rsatadi.

Birbuyruqyaratishda, buninguchunbirhodisayaratishkerak.

Agarfaollashtirilganbo‘lsa,

buyruqaniqyokiyoqilganbo‘lsahodisabajarilishinibildiradi,

uningmatninio‘zgartirishimkoniniberadivabuyruqqategishlijavobta‘minl

aydi (yo‘naltiriladi). Ko‘phollarda, IDE IOleCommandTarget

interfeysiyordamidabuyruqlarjarayonlaribajariladi. Visual Studio

yo‘nalishidagibuyruqlarfoydalanuvchitanlovgaasoslanganbuyruqningildi

zkontekstidanboshlabva global

tanlovasosidatashqikontekstgachaierarxiyasifatidako‘rsatiladi.

Asosiymenyugaqo‘shilganbuyruqlarssenariylardafoydalanishuchunqulay

.

Yangi menyular va uskunalarmajmuasiani qilash uchun, Visual Studio buyruq jadvalda (.vst) fayllar qali ularni tasvirlash kerak. Visual Studio paketi shablon uchun bu faylni yaratadi, shuningdek kerakli elementlar shablon datanlangan har qanday buyruqlarni, uskunalar majmuasivamuharrirlarini qo'llab-quvvatlash gakerak.

**Visual Studio menyular va buyruqlar.** Microsoft Office dan farqli o'laroq, bir qancha alohida mahsulotlar paketidan iborat, Visual Studio ko'pmahsulotlar o'zicha o'ladi, qaysiki har bir global Visual Studio IDE buyruqlar o'z o'plamiga hissa qo'shadi. IDE kontekst asosida foydalanuvchi gamavjud funksiyalarni filtrlash orqalimingl ab buyruqlarning murakkabligini boshqaradi.

Loyihalash o'ynasida kodni tahrirlash o'ynasiga o'tish kab foydalanuvchi kontekst o'zgaranda, yangi kontekst bilan bog'liq bo'lmagan funksiyalar yo'q o'ladi. Shu bilan bir vaqtda, yangi funksional tegishli dinamik ma'lumotlar bilan birga o'chiladi, bunday Toolbox xususiyatlarini va parametrlarini sifatida qaraladi. Foydalanuvchi mavjud buyruqlar o'plamini almashtirishni sezmasligi kerak.

Foydalanuvchi chalg'itdiyoki buyruqlar paydoyokig'oyi bilan adashtirmaslik, foydalanuvchi interfeysi interfeysi sozlanishi kerak. Joriy foydalanuvchi kontekstida IDE sarlavhasida, xususiyatlar o'ynasida yokis ahifalarimuloqot o'ynasida bo'lgan kabibiryokib irnecha usul dako'rsatiladi.

Buyruq o'plamilar ulgamoslashuvchanligini ta'minlaydi. Visual Studio muhitiga xos bo'lgan yagona buyruq strukturalari asosiy menyuga asosiy buyruqlar satribo'lib, ularni sozlash va hatto yashirish mumkin. Boshqa buyruqlar majmuasii lovaning holatiga qarab paydobo'ladivayo'qoladi. Oynalar uskunasi va hujjat muharrirlari, shuningdek, ajralmas oynalar uskunalar majmuasio'zicha o'lishi mumkin.

menyular qachon mavjud bo'lsa, birgalikda buyruqlar va buyruq guruhlaridan foydalaning.

Buyruqlar odatda kontekst asosidan amoyishi etiladi, chunki, mavjud menyular va buyruq guruhlar yordamida buyruq tuzilishi kontekstida o'zgarishlar o'rtasidan isbatanbarqaror bo'lib qolishini ta'minlaydi. Buyruqlar guruhini qayta ishlatish va tegishli buyruqlar yaqinidagi buyruqlarni joylashtirish IDE murakkabligini kamaytiradi va yanada samimiylashtirib yaratadi.

Agaryangibuyruqaniqlashkerakbo‘lsa,  
mavjudbuyruqguruhigauniqo‘yibko‘rishmumkin.  
Agaryangiguruhaniqlashkerakbo‘lsa,  
yangiyuqoridarajalimenyuniyaratishvabog‘liqbuyruqguruhiyonidamenyu  
sidajoylashtirishmumkin.

Harbirbuyruquchunpiktogrammayaratishyaxshiemas.  
Buyruqbelgisiniyaratishnidiqqatbilano‘ylabko‘ring.  
Piktogrammafaqatquyidagibuyruqlaruchunyaratilishikerak:

\* joriyuskunalarpanelidako‘rsatiladiganlar.

\* foydalanuvchilartomonidanuskunalarpaneligaqo‘shish,  
maxsusmuloqotorqalimumkin.

\* Microsoftningmahsulotibirxilharakatbilanbog‘liqbuyruqlarga.

Klaviaturatugmalariniqo‘shishnicheklash.

Foydalanuvchilarningaksariyatibarchamavjudtugmalarningkichikqismini  
ishlatadi. Agarshubhangizbo‘lsa,

funksiyaniklaviaturatugmasibilanbog‘lamang.

Yangiyorliqlarniqo‘shishdanoldinfoydalanuvchilarguruhibilanishlash.

Komandalargastandartmenyujoyiniberish.

Jamoalarboshqalartomonidantuzilganvashungako‘ramo‘ljallanganbo‘lad  
i. Yashirinbuyruqdegannarsayo‘q. Barcha Visual Studio buyruqlari  
Tools > configure muloqotoynasida, buyruqoynasida,  
avtomatikiyakunlash, >> options >klaviaturamuloqotoynasidava  
development tools muhitida (DTE) ko‘rsatiladi.  
Buyruqlarnominivavositalarimajmuiniberishgaishonchhosilqilishkerakva  
foydalanuvchilarosonginaularnitopishmumkin, shundayqilib, .ctc  
faylyaratishkerak.

Ichkiuskunalarpanelidagiumumiybuyruqlarnitakrorlamang.

Buyruqlarnifoydalanuvchiningdiqqatmarkazigayaqinjoydajoylashtirishfo  
ydalidir.

Buningbiryo‘liuskunaoynasiyoki hujjatmuharririningyuqoriqismidao‘rnat  
ilganuskunalarpaneliniyaratishdir.

Uskunalarpaneligajoylashtirilganbuyruqlaroynadagikontentmaydonigax  
osbo‘lishikerak.

Buuskunamajmuasiningumumiybuyruqlarninusxaqilmang. Masalan,  
ajralmasuskunalarpanelidasaqlashbelgisinijoylashtirishkerakemas.

MS Visual Studio muhitininginteraktivtugmalari.

Asosiytugmalar

**Ctrl+Shift+P, F1** Buyruqlarpalitrasiniko‘rsatish

**Ctrl+P** Tezochish, faylgao‘tish...

<b>Ctrl+Shift+N</b>	Yangioynayaratish
<b>Ctrl+Shift+W</b>	Oynaniyopish
<b>Ctrl+,</b>	Foydalanuvchisozlamalari
<b>Ctrl+K Ctrl+S</b>	Klaviaturayorliqlar

#### Asosiy tahrirlashtugmalari

<b>Ctrl+X</b>	Buferga olish va o'chirish
<b>Ctrl+C</b>	Buferdanni saqlash
<b>Alt+ ↑ / ↓</b>	qatorni yuqoriga / pastgako'chirish
<b>Shift+Alt + ↓ / ↑</b>	Qatorni saqlash yuqoriga / pastgako'chirish
<b>Ctrl+Shift+K</b>	Qatorni o'chirish
<b>Ctrl+Enter</b>	Yangi qator joylashtirish
<b>Ctrl+Shift+Enter</b>	Yuqoridagi yangi qator joylashtirish
<b>Ctrl+Shift+\</b>	Moskeladigan qavsga o'tish
<b>Ctrl+] / [</b>	Qatorga abzas qo'shish / o'chirish
<b>Home / End</b>	Satrboshi / oxiriga o'tish
<b>Ctrl+Home</b>	Faylboshiga o'tish
<b>Ctrl+End</b>	Fayloxiriga o'tish
<b>Ctrl+↑ / ↓</b>	qator dani yuqoriga / pastga harakatlashtirish
<b>Alt+PgUp / PgDn</b>	sahifa bo'yicha yuqoriga / pastga harakatlashtirish
<b>Ctrl+Shift+[</b>	Operatortasmasini belgilash
<b>Ctrl+Shift+] ]</b>	Operatortasmasini belgilashni bekor qilish
<b>Ctrl+K Ctrl+[</b>	Tasmasini o'chirish
<b>Ctrl+K Ctrl+] ]</b>	Tasmani qaytarib olish
<b>Ctrl+K Ctrl+0</b>	Qatorlarga izoh qo'yish
<b>Ctrl+K Ctrl+J</b>	Qatorlardan izohni o'chirish
<b>Ctrl+K Ctrl+C</b>	Satrga izoh qo'shish
<b>Ctrl+K Ctrl+U</b>	Satrdan izoh o'chirish
<b>Ctrl+ Z</b>	Amalni bekor qilish

#### Navigatsiya tugmalari

<b>Ctrl+G</b>	Kerakliqatorga borish
<b>F8</b>	Navbatdagixatoga borish
<b>Shift+F8</b>	Navbatdagixatovaxabargaborish
<b>Ctrl+Shift+Tab</b>	Tahrirlash oynalariga o'tish
<b>Ctrl+M</b>	Fokustugunlarni o'zgartirish

#### Izlash va o'zgartirish tugmalari

<b>Ctrl+F</b>	Izlash
---------------	--------

<b>Ctrl+H</b>	O'zgartirish
<b>F3 / Shift+F3</b>	Izlashningkeyingisigao'tish
<b>Alt+Enter</b>	Izlashnatijalarinibelgilash

Dasturlashgako'maklashuvchitugmalar

<b>Ctrl+Space</b>	Tegishlifunksiyalarnichaqirish
<b>Ctrl+Shift+Space</b>	Tegishlifunksiyavariantlarinichaqirish
<b>Shift+Alt+F</b>	Dasturniformatlash
<b>Ctrl+E Ctrl+F</b>	Dasturfragmentiniformatlash
<b>Ctrl+Shift +.</b>	Dasturmatninikattalashtirish
<b>Ctrl+Shift +,</b>	Dasturmatninikichiklashtirish

Buinteraktivtugmachalardantashqario'rganibolishingizkerakbo'lga  
ntugmachalarjudako'p,  
ularnidasturlashdavomidafoydalanishgao'rganibolasizdeganumiddamiz.

**Form xususiyatlarivahodisalari.** Form  
buloyihayaratilganhosilbo'ladiganbirinchioynahisoblanadi.

Oynaningjoriyholatlarimavjud.

Ulariloyihaboshqaruvchisipanelidanko'rishmumkin. Oynauchunyangi  
form1 sinfiyaratiladi. Businf form sinfiningmerosxo'rihisoblanadi.

Uquyidagichaaniqlangan:

```
public ref class Form1 : public
System::Windows::Forms::Form{}
```

ShusinfningichidaInitializeComponentfunksiyasiborvauformaniiyar  
atilishjavobberadi.

Uningdastlabkisozlamalariquyidagifragmentdakeltirilgan:

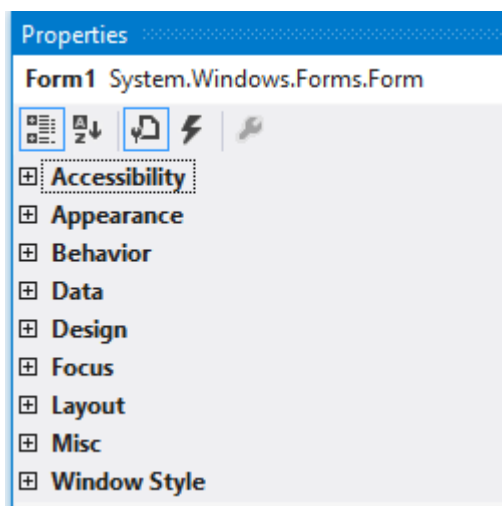
```
this->AutoSizeDimensions = System::Drawing::SizeF(6,
13);
this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
this->ClientSize = System::Drawing::Size(284, 261);
this->Name = L"Form1";
this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterScre
en;
this->Text = L"Form1";
this->ResumeLayout(false);
```

Imkoniyatqadarbudasturfragmentlarigao'zgartirishkiritishshartema  
s, bularnihammasinivizualbajarishmaqsadgamuvofiq.

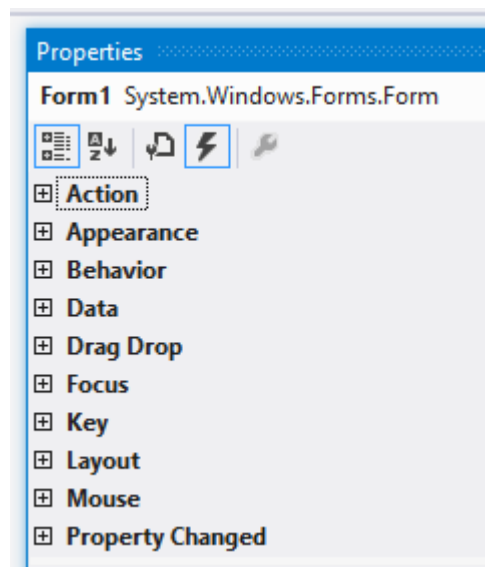
Vizual bajarish deganda formaning vizual loyihasi bilan ishlash kerak. Loyihayaratuvchisi uchun quyidagilari muhim hisoblanadi:

- Formani ustiga sichqonchanib o'sangiz u aktivlashadi.
- Kontekst meniyudan foydalanib, uning dastur fragmentiga o'tish mumkin (F7 tugmasi orqali)
- Dastur fragmentidan forma loyihasiga o'tish uchun kontekst meniyudan foydalanish mumkin (Shift + F7 tugmasi orqali)
- Kontekst meniyu oqali formani boshqarishni qulflab qo'yish va ochish mumkin (lock/unlock controls)
- Kontekst meniyu oqali formani xususiyatlarini vahodisalariga o'tish mumkin (Properties)

Formaning xususiyatlarini vahodisalar quyidagilarda bo'limlardan iborat (9.5 va 9.6-rasmlarga qarang)







9.5-rasm. Formaning xususiyatlari



9.6-rasm. Formaning hodisalari

Properties muloqot oynasida 4 ta uskunalarmavjud. Ularning vazifalari:

-  - forma parametrlarini guruhlab saralashni ta'minlaydi.
-  - forma parametrlarini alfavit bo'yicha saralashni ta'minlaydi.
-  - forma xususiyatlarini qo'ldan chiqaradi.
-  - forma hodisalarini qo'ldan chiqaradi.

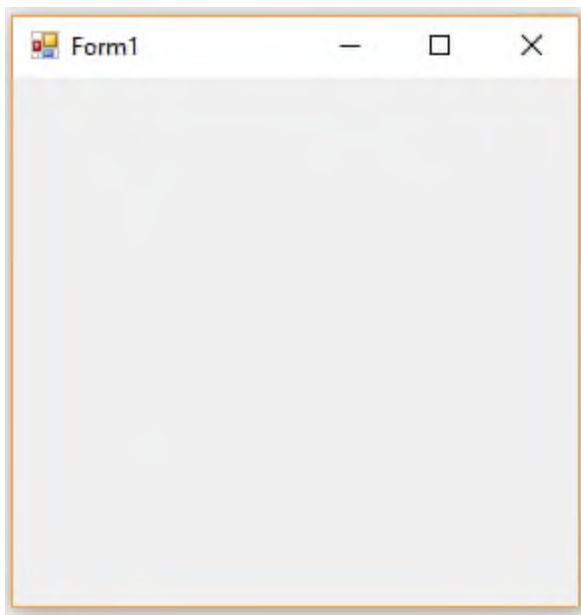
E'tibor bergan bo'lsangiz, tanlangan joriy turkumga o'tib burchakka olingan bo'ladi. Bueradabajarilgan har bir ish forma gamsini fanga zgartirib boriladi. Endi forma xususiyatlarining ba'zilarini bilgan tanib chiqamiz.

1 **Accessibility** Buning 3 ta xususiyati bor,

<b>guruhi</b>	buxususiyatlaribirformabilano‘zgarishini vhaniqlabbo‘lmaydishinguchunkeyinroqto‘xtalamiz.	
<b>2 Appearanceguruhi</b>	Formaningtashqiko‘rinishiuchunishlatiladiva	11
BackColor	Formagarango‘rnatish	
BackgroundImage	Rasmo‘rnatish	
BackgroundImageLayout	Rasmo‘rnatishtartibi	
Cursor	Keraklikursorbelgisinitanlash	
Font	Formadayozuvxususiyatlarinio‘rnatish	
ForeColor	Yozuvranginitanlash	
FormBorderStyle	Formaningstilinio‘rnatish	
RightToLeft	Formasarlavhasinio‘ngdanchapgao‘rnatish	
RightToLeftLayout	Formasarlavhasinio‘ngdanchapgao‘rnatishtartibi	
Text	Sarlavhanomi	
UseWaitCursor	Kutishkursoridanfoydalanishnio‘rnatish	
<b>3 Behaviorguruhi</b>	Formaningrejimlarinio‘rnatishva6xususiyatibor	
ContextMenuStrip	Kontekstmenyu o‘rnatish	
DoubleBuffered	Bufero‘rnatish	
Enabled	Formaniyoqishvao‘chirish	
<b>4 Dataguruhi</b>	Formaniichkivatashqima’lumotlarbilanbog‘lashuchunishlatiladiva3xususiyatibor. Buguruhikeyingielementlaribilanishlagandabatavsilqaraymiz	
<b>5 Designguruhi</b>	Formanilo yihalashuchunishlatiladiva3taxususiyatibor	
Language	Formagajoriytilnio‘rnatish	
Localizable	Formagalokalizatsiyanio‘rnatish	
locked	Formaniyopishyokiochish	
<b>6 Focusguruhi</b>	Formadafokuslarniboshqarishuchunishlatiladivabittaxususiyatibor	
CausesValidation	Fokuslarnitekshirishbosqoichlari	

<b>7 Layout guruhi</b>	Formanitari blash uchun ishlatiladiva taxususiyatibor
AutoScroll	Avtomatik formanikuzatishtugmalarinioʻrnatish
AutoSize	Avtomatikoʻlchamni joriy qilishnioʻrnatish
Location	Formaniosilboʻlish joyibelgilashnioʻrnatish
MaximumSize	Avtomatikkattaholdachiqish
MinimumSize	Avtomatikkichiklashtirilganxodachiqish
Size	Formagaoʻlchamoʻrnatish
StartPosition	Formaningchiqishjoyinioʻrnatish
WindowState	Formaningturibelgalash
<b>8 Miscguruhi</b>	Boshqaxususiyatlarnioʻrnatish, formanituriga qaraboʻzgarib turadiva 3 taxususiyatibor
<b>9 Windows Styleguruhi</b>	Oynastillarinioʻrnatish uchun moʻljallangan guruhboʻlib, 13 taxususiyatgaega.
ControlBox	Boshqarishtugmalarini joylashtirishni aniqlash
HelpButton	Yordamchaqiruvchitugmanioʻrnatish
Icon	Oynabelgisinioʻrnatish
MaximizeBox	Kattalashtirishtugmasinioʻrnatish
MinimizeBox	Kichiklashtirishtugmasinioʻrnatish
Opacity	Oynaningtinqliginisozlashnioʻrnatish

Mazkur xususiyatlardan foydalanib, formaningbaʼzi xususiyatlarinioʻzgartiramiz va uning sinfining ichigakiribkoʻrishimiz mumkin:



9.7-rasm.



9.8-rasm.



Formaningdastlabkiko‘rinishi      Formaningishlovberilganko‘rinishi  
 Formagaishlovberilganxususiyatlarianiqlashuchunsinfigamurojjetqil  
 ishmumkin. Quyidagidasturfragmentlariniolamiz.

Formaningdastlabkiko‘rnatilganxususiyatlari	Formaningishlovberilganxususiyatlari	+/-
<code>this-&gt;AutoScaleDimensions = System::Drawing::SizeF(6, 13);</code>	<code>this-&gt;AutoScaleDimensions = System::Drawing::SizeF(11, 22);</code>	+
<code>this-&gt;AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;</code>	<code>this-&gt;AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;</code>	-
	<code>this-&gt;BackColor = System::Drawing::SystemColors::ActiveCaption;</code>	+
<code>this-&gt;ClientSize = System::Drawing::Size(284, 261);</code>	<code>this-&gt;ClientSize = System::Drawing::Size(230, 257);</code>	+
	<code>this-&gt;Cursor = System::Windows::Forms::Cursors::Hand;</code>	+
	<code>this-&gt;Font = (gcnew System::Drawing::Font(L "Times New Roman", 14.25F, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point, static_cast&lt;System::Byte&gt;(204)));</code>	+
	<code>this-&gt;FormBorderStyle = System::Windows::Forms::FormBorderStyle::Fixed</code>	+

	3D;	
	this->HelpButton = true;	+
	this->Icon = (cli::safe_cast<System::Drawing::Icon^>(resources->GetObject(L"\$this.Icon"))));	+
	this->Margin = System::Windows::Forms::Padding(6, 6, 6, 6);	+
	this->MaximizeBox = false;	+
	this->MinimizeBox = false;	+
this->Name = L"Form1";	this->Name = L"Form1";	-
this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;	this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;	-
this->Text = L"Form1";	this->Text = L"Birinchi oynam";	+
this->ResumeLayout(false);	this->ResumeLayout(false);	-

Birinchi navbatdagi muhitning o'ziga xos xususiyatlarni o'rnatishni vatar birlashtirishni nazorat qiladi. Formada quyidagi o'zgarishlar amalga oshirilgan: formaning chiqish maydoni, rangi, joriyo'lchamlari, oynakursori, oynaning yozuv turi, ko'rinishi, yordamchitugma o'rnatilgan, ikonkasi o'zgartirilgan, holati o'zgartirilgan, kattalashtirish va kichiklashtirish tugmalarini olib tashlangan, oynaning sarlavhasi o'zgartirilgan.

Bularni amalga oshirish uchun  
munut vaqt ketadi va hammasi vizual bajariladi.  
Bunda oldin qilingan o'zgarishlar o'chirib,  
oxirgi o'zgarishlar qo'shib boriladi.

2-3

## Formaning hodisalar bilan tanishibchiqamiz. O'z hodisa nima?

Hodisa bu bir vazifa bajarilgan jarayon bo'lishi mumkin, masalan, formani yaratish uchun 3 ta hodisa bo'lishi mumkin, formani yaratilishidan oldin, formani yaratilayotgan vaqt, formani yaratilib bo'lingandan keyin.

Ularning ham 9.6- rasmda keltirilgan 10 ta guruhi bor. Ularning ba'zilariga izoh berib o'tamiz

- Action guruhi** Global hodisalar guruhi bo'lib, 8 ta hodisa dan iborat
  - Click Formaga tugma bosilganda
  - DoubleClick Formaga tugma 2 marta bosilganda
  - MouseCapture Sichqonchani olatib o'zgartirganda
  - Changed
  - MouseClicked Sichqoncha bosilganda
  - MouseDoubleClick Sichqoncha ikki marta bosilganda
  - ResizeBegin Forma o'lchamini o'zgartirish boshlanganda
  - ResizeEnd Forma o'lchamini o'zgartirish tugaganda
- Appearance guruhi** Formaning ichki hodisalarini uchun ishlatiladigan bitta xususiyatni ko'rsatib beradi, ularning keyingi mavzularni tanishamiz. t
- Behavior guruhi** Formaning holatlarini uchun hodisalarini o'z ichiga oladigan 15 ta hodisa dan iborat
  - ControlAdded Formaga biror boshqarish qurilganda
  - ControlRemoved Formadan biror boshqaruvchi o'chirilganda
  - FormClosed Forma yopilganda
  - FormClosing Forma yopilishidan oldi holatda
  - HelpButtonClicked Yordam tugmasi bosilganda
  - InputLanguageChanged Tilo o'zgartirilganda
  - InputLanguageChanging Tilo o'zgartirilayotganda
  - Load Forma yuklanayotganda
  - Shown Forma chiqarilganda
- Data guruhi** Ichki va tashqi ma'lumotlar bilan ishlash hodisalarini bo'lib, 2 ta hodisa dan iborat, bu xususiyatlarda ma'lumotlar bazasi bilan ishlashni ta'kidlash unarlib o'tiladi
- Drag** Formada harakatlanish hodisalarini bo'lib, 6

	<b>dropguruhi</b>	tahodisadaniborat	
	DragDrop	Birorbirharakatbo'lganda	
	DragEnter	Intertugmasibosilganda	
	DragLeave	To'xtashharakatibo'lganda	
	DragOver	Ketishharakatibo'lganda	
	GiveFeedback	Teskarialoqaqo'shish	
6	<b>Focusguruhi</b>	Formadafokuslarniboshqarishuchunbo'lib, tahodisani'o'zichigaoladi	6
	Activated	Fokusaktivbo'lganda	
	Enter	Fokusdama'lumotkiritilganda	
	Leave	Fokusdato'xtashsodirbo'lganda	
	Validated	Tekshirishtugaganidanso'ng	
	Validating	Tekshirishvaqtida	
7	<b>Keyguruhi</b>	Tugmalarbilashishlashhodisalaruichunbo'lib, tahodisadaniborat	6
	KeyDown	Tugmakelganda	
	KeyPress	Tugmabosilganda	
	KeyUp	Tugmaketganda	
	PreviewKeyDown	Tugmakelganda	
8	<b>Layoutguruhi</b>	Tartibguruhi formaniatrofidagilarbilanishlashtartibini nazoraqilashuchunrejalashtirilgan	
	DpiCahged	Joriyformatartibio'zgarganda	
	DpiCahgedAfterParent	Merosbergansinftartibio'zgargandankeyin	
	DpiCahgedBeforeParent	Merosbergansinftartibio'zgargandanoldin	
	Layout	Formatartibinio'rnatish	
	MdiChildActivate	Formasinfiningmerosxo'rifaolbo'lganda	
	Move	Formako'chirilganda	
	Resize	Formao'lchaminio'zgartirganda	
9	<b>Mouseguruhi</b>	Sichqonchabilanbo'ladiganhodisalarbo'lib, tahodisadaniborat	6
	MouseDown	Sichqonchakelganda	
	MouseEnter	Sichqonchajoriyholatida	
	MouseHover	Sichqonchaozginaharakatida	
	MouseLeave	Sichqonchagato'xtashbo'lganda	
	MouseMove	Sichqonchako'chganda	

	MouseUp	Sichqonchaketganda
1	<b>Property</b>	Formaningxususiyatlario‘zgargandabajariladigan 12
0	<b>Changedguruhi</b>	hodisasibor
	AutoSizeChanged	Avtomatiko‘lchamo‘zgarganda
	AutoValidateChanged	Avtomatikteksiribbo‘lganda
	BackColorChanged	Fonrangio‘zgarganda
	EnabledChanged	Holato‘zgarganda

Hodisalarnio‘rnatishjudaqulayhisoblanadi.

Keraklihodisaningo‘ngtomonigasichqonchaniikkimartabossangiz, hodichamosbo‘lgandasturfranmentiniyozishuchunfunksiyayaratibberadi. Birmisolbilanko‘ribchiqamiz.

Buninguchun MessageBox sinfining show funksiyasidanfoydalanamiz. Bufunksiyastatikfunksiyabo‘lib, System::Windows::Formsnomlarfazosigategishli, 21 variantdayozilishimumkin. Unatijasifatida DialogResult tipiniqaytaradi. Uningargumentlari:

**Text** - Stringtipiniqabulqiladivamatnixabarsifatidachi qaradi

**Caption** - Stringtipiniqabulqiladivaxabarsarlavhasinichiqaradi

**Buttons** -

MessageBoxButtonstipiniqabulqiladivaxabarniqabulqilishtugmalarinichiqaradi, AbortRetryIgnore, OK, OKCancel, RetryCancel, YesNo, YesNoCancel qiymatlarniqabulqiladi.

**Icon** -

MessageBoxIconstipiniqabulqiladivaxabarniikonkasinichiqaradi, Asterisk, Error, Exclamation, Hand, Information, None, Question qiymatlarniqabulqiladi

**defaultButton** -

MessageBoxDefaultButtonstipiniqabulqiladivaxabarnitugmalarinibelgilaydi, Button1, Button2, Button3 qiymatlarniqabulqiladi.

**Options** -

MessageBoxOptionstipiniqabulqiladivaxabarnichiqishvarintlarinianiqlaydi, DefaultDesktopOnly, RightAlign, RtlReading, ServiceNotification qiymatlarniqabulqiladi.

Bufunksiyaning boshqa argumentlarini hamkor, ularni keyinchalik o'rganish mumkin.

MessageBox sinfining show funksiyasidan foydalanish:

```
MessageBox::Show("Xabar", "Sar...",  
MessageBoxButtons::OK, MessageBoxIcon::Information  
);
```

Forma hodisalaridan foydalanish uchun 3 ta sinifni o'rganish kerak:

1. Click hodisasini ishlatish uchun, uning o'ng tomoniga sichqonchanib osamiz.

Forma sinifidatayyor Form1\_Click() nomli funksiyalar yaratiladi.

Bu funksiyani ichida quyidagicha fragmentni yozamiz:

```
MessageBox::Show("Click  
bo'ldi", "Xabar", MessageBoxButtons::OK,  
MessageBoxIcon::Information);
```

Shuningdek forma ning InitializeComponent()

funksiyasini ichida avtomatik tarzda this->Click += gnew

System::EventHandler(this,

&Form1::Form1\_Click); hodisa qo'shib qo'yilgan.

Agar bu hodisa noto'liq chirmoqchibo'lsangiz, Click

hodisasining o'ng tomoniga yozilgan Form1\_Click() ni o'chirib, enter

tugmasini bosish yetarli o'z avtomatik InitializeComponent()

funksiyasini ichida hodisa o'chiradi.

Dasturning qismidan funksiyani o'chirish mumkin.

Quyidagi ikki hodisa ishlatish uchun bir masala o'lamiz.

Yordam tugmasi bosilganda, yordam tugmasi boshida deb xabar chiqsin.

Foydalanuvchi hadeb javob bersa, oynaga qaytadi. Agaryo'q deb bossa,

yolg'onchimisiz deb chiqadivafoydalanuvchi, faqat javob nitanlaydi.

Javob tanlaganda oynaning sarlavhasi yolg'onchidebo'zgarsin.

Oynalathodisalariorqalib tekshirib,

yana oynasarlavhasi o'zgardidegan habarni chiqarsin.

2. HelpButtonClicked hodisasini ishlatish uchun,

uning o'ng tomoniga sichqonchanib osamiz.

Forma sinifidatayyor Form1\_HelpButtonClicked ()

nomli funksiyalar yaratiladi.

Bu funksiyani ichida quyidagicha fragmentni yozamiz:

```
System::Windows::Forms::DialogResult result;  
result =  
MessageBox::Show("HelpButtonClicked", "Xabar", MessageB  
oxButtons::YesNo, MessageBoxIcon::Hand,
```

```

MessageBoxDefaultButton::Button1);
        if(result ==
System::Windows::Forms::DialogResult::No) {
            MessageBox::Show("Yolg'nchimi siz
!", "Xabar", MessageBoxButtons::OK, MessageBoxIcon::Warn
ing);
                Form1::Text = "Yolg'nchi";
            }
        }
}

```

Shuningdek formaning InitializeComponent() funksiyasini ichida avtomatik tarzda this->HelpButtonClicked += gnew System::ComponentModel::CancelEventHandler(this, &Form1::Form1\_HelpButtonClicked); hodisa qo'yilgan.

3. TextChanged hodisasini ishlatish uchun, uning o'ng tomoniga sichqonchanib osamiz. Formasini fidatay yor Form1\_TextChanged() nomli funksiyayaratiladi. Bu funksiyani ichida quyidagicha fragmentni yozamiz:

```

MessageBox::Show("Forma sarlavhasi
o'z gardi", "Xabar", MessageBoxButtons::OK,
MessageBoxIcon::Information, MessageBoxDefaultButton::
Button3, MessageBoxOptions::RightAlign);

```

Shuningdek formaning InitializeComponent() funksiyasini ichida avtomatik tarzda this->TextChanged += gnew System::EventHandler(this, &Form1::Form1\_TextChanged); hodisa qo'yilgan.

Bu hodisalarni ishlatib ko'rsangiz, ancha narsalarni farq qaborishingiz kerak.

Matn dabr joyda forma, bir joyda oyna, bir joyda form deb ishlatilgan bo'lsa, ularning hammasi bir ma'noni beradi. Shubilan formaning xususiyatlarini vahodisalari bilan tanishib chiqdingiz va uning o'rganishni davometish kerak. Chunki, keyingi barcha ishlaringiz shukabi formalar bilan bog'liq. Keyinchalik formaning ma'lum bir xususiyatini vahodisalari bilan ishlaganda. Albatta unita'kidlabaytib o'tib ketamiz.

NAZARIY BILIMLARNI TEKSHIRISH UCHUN  
SAVOLLAR.

1. Integrallashgan ishlabchi qarish muhitlarining qayday turlarini bilasiz?
2. Qachondan boshlab integrallashgan muhitlar jadal rivojlangan?
3. Turbo muhitlarni sanabbering.
4. Ob'ektga yo'naltirilgan tilni birinchi bo'lib integrallashgan muhitlarga akim va qaysi firmajoriy qilgan.
5. Qanday muhitda zamonaviy matn muharrirlari kodni avtomatik bajarilishini ta'minlash (kodni tugatish), muharrir muhitda joriy yozilgan kodni sintaktik to'g'ri va uning davomini bo'lishi mumkin.
6. Taniqli NetBeans integrallashgan muhitni dastlab Java dasturlash uchun qaysi universitet talabasi loyihasi sifatida yaratgan.
7. Visual Studio 97 haqidagi ma'lumotlarni bilasiz?
8. Visual Studio 2012 haqidagi ma'lumotlarni bilasiz?
9. ISO xalqaro standartlari bilan standartlashtirilgan va bir necha implementations egabotilgan ob'ektga yo'naltirilgan kodni dasturlash platformasi qanday?
10. MS Visual Studio muhitini nasib etish uchun qanday qismlarni o'rnatish kerak?
11. Windows ilovalarni yaratish uchun qaysi tugmachalar majmuasini o'rnatish zarur.
12. Qaysi vosita bir necha qo'llab-quvvatlanadigan tillar uchun dasturlarni yaratish uchun ishlatiladi.
13. CLR Empty Project qanday loyiha yaratish uchun kerak.
14. Windows Forms Control Library qanday loyiha yaratish uchun kerak.
15. Loyiha uchun External Dependencis papkasida qanday fayllar saqlanadi.
16. [FILE] menyusining vazifalarini sanabbering.
17. [TOOLS] menyusining vazifalarini sanabbering.
18. Menyular va uskunalarning paneli foydalanuvchilarning qanday papkaga buyruqlarni kiritish uchun bir yo'l.
19. Bir buyruq yaratishda, buning uchun bir avvalni yaratish kerak.
20. Yangi menyular va uskunalarning majmuasini yaratish uchun, Visual Studio da qanday jadval fayllar qali ularni yaratish kerak.
21. Qatorlarga izoh qo'yish tugmasini ayting.
22. Satrga izoh qo'yish tugmasini ayting.
23. Tegishli funksiyalar variantlarini chaqirish tugmasini ayting.
24. [Ctrl+Shift +.] tugmachalar majmuasi qanday amal bajaradi.
25. [Ctrl+Shift+Tab] tugmachalar majmuasi qanday amal bajaradi.



26. Oynauchunyangi form1 sinfiyaratiladi.  
 Businfqaysisinfiningmerosxo‘rihisoblanadi
27. Kontekstmenyu oqaliformaningqandayfunksiyalarinibajarishmu  
 mkin.
28. Formaningrejimlarinio‘rnatishqaerdajoylashgan.
29. Ichkivatashqima’lumotlarbilanishlashhodisalarigaerdajoylashgan
30. MessageBox sinfiningnechtaargumentlaribor,  
 vaharbirigaizohbering.



**AMALIY KO‘NIKMA VA MALAKALARNI ANIQLASH  
 HAMDA RIVOJLANTIRISH UCHUN ASSISMENT  
 TOPSHIRIQLARI.**

<b>BIRINCHI ASSISMENT TOPSHIRIG‘I</b>	
	<p>Form xususiyatlarivahodisalarigaoidberilganquyidagid asturbo‘yichaberilgantopshiriqlarkeraklifragment lariasosidabajaring.</p> <p>☞ Bundadasturdagiba‘zio‘zgartirishlarnitopishorq alitopshiriqlarbosqichma – bosqichamalgaoshriladi.</p>
<b>№ - sizningtug‘ilgankuninggiz</b>	
<p>1. Formayaratingvaunio‘lchamlariga (№*50, (№%2)*100) qiymatlarnio‘rnating.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>	
<p>2. Formaning (№%6) - hodisasibajarinlanginitekshiring?          ( 0- Click, 1- DoubleClick, 2 - MouseClick, 3- MouseDoubleClick, 4- ResizeBegin, 5 – ResizeEnd )</p> <hr/> <hr/>	

<hr/> <hr/> <hr/>
<p>3. Formaning ((No/4)%6) - hodisasibajarinlanginitekshiring, bundaMessageBox::Showfunksiyasining (No%5)+1 taargumentidanfoydalanig ?</p> <p>( 0- Click, 1- DoubleClick, 2 - MouseClick, 3- MouseDoubleClick, 4- ResizeBegin, 5 – ResizeEnd )</p> <hr/> <hr/> <hr/> <hr/>
<p>4. Formaning ((No/4)%8) - hodisasibajarinlanginitekshiring, bundaMessageBox::Showfunksiyasining (No%5)+1 taargumentidanfoydalanig ?</p> <p>(0-FormClosed, 1- FormClosing, 2-HelpButtonClicked, 3- Load, 4-Shown, 5-Activated, 6-Move, 7-Resize)</p> <hr/> <hr/> <hr/> <hr/>
<p>5. Formaning ((No/4)%7) - hodisasibajarilanginitekshiring, bundaMessageBox::Showfunksiyasining (No%5)+1 taargumentidanfoydalanig ?</p> <p>(0-MouseDown, 1-MouseEnter, 2-MouseHover, 3-MouseLeave, 4-MouseMove, 5-MouseUp, 6-AutoSizeChanged)</p> <hr/> <hr/> <hr/> <hr/>

6. Formaning (№%8) - hodisasibajarilanga, tekshiringvanoto‘g‘rideganjavobda (M%7) - hodisanibajartiring, bundaMessageBox::Showfunksiyasining (№%3)+1 taargumentidanfoydalanig ?

M uchun – (0-MouseDown, 1-MouseEnter, 2-MouseHover, 3-MouseLeave, 4-MouseMove, 5-MouseUp, 6-AutoSizeChanged), № uchun - (0-FormClosed, 1- FormClosing, 2-HelpButtonClicked, 3-Load, 4-Shown, 5-Activated, 6-Move, 7-Resize)

---



---



---



---



---



---



---



---



---



---


7. Formaningxususiyatlarini ((№\*2)%12), ((№+6)%12), ((№/6)%12) bajaribtekshiring.

0- BackColor	_____
1- BackgroundImage	_____
2- Cursor	_____
3- Font	_____
4- FormBorderStyle	_____
5- RightToLeft	_____
6- Text	_____
7- UseWaitCursor	_____
8- Enabled	_____
9- Language	_____
10- locked	_____
11- AutoScroll	_____


<b>8. Formaning xususiyatlarini</b>	<b>((№*2)%13),</b>	<b>((№+6)%13),</b>
<b>((№/6)%13) bajarib tekshiring.</b>		
0- AutoSize 1- Location 2- MaximumSize 3- MinimumSize 4- Size 5- StartPosition 6- WindowsState 7- ControlBox 8- HelpButton 9- Icon 10- MaximizeBox 11- MinimizeBox 12- Opacity		
<b>9. Biror bir hodisa bilan yuqoridagimos xususiyatlarni</b>		<b>2</b>
<b>tasini o'zgartiring.</b>		
<b>10. Hodisalarni boshqarishgadoir masalating va uning bajarilishini.</b> Masala _____ _____ _____ _____ <b>Echimi</b> _____ _____ _____ _____ _____ _____		


- |   |
|---|
| <p>11. Barchavazifalarnio‘rganganingizbo‘yichaeslabko‘ring,<br/>nimalarnio‘rgandingizvabirtakrorlang.</p> <p>12. Barchabilimvako‘nikmalarasosidakichikloyihatayyorlang.</p> |
|---|


### 3.2. Komponentalar bilan ishlash.

 Komponenta tushunchasivaxususiyatlari, Standart komponentasiningxususiyatlarivahodisalari, Additional komponentasiningxususiyatlarivahodisalari, System komponentalarikomponentasiningxususiyatlarivahodisalari, Toolbox oynasi, Yangikomponentalarniqo‘shish, komponentalarnio‘zarobog‘lashbo‘yichanazariyma’lumotlarkeltirilganbo‘lib, nazariybilimlarniasoslashuchundasturlartuzibko‘rsatilgan. Bilimlarnimustahkamlashuchun 30 tanazariysavolvaamaliyko‘nikmavamalakalarnirivojlantrishuchun 8 assismenttopshirig‘iberilgan.

Butopshiriqlarnibajarishmavzunimustahkamlashuchunxizmatqiladi.

 *Kalitso‘zlar.* IDE, Visual C++, user interface, OYD, komponenta, xususiyatlari, hodisalari, Toolbox, Tab, Add Tab, Delete Tab, Mantlarnitahrirlash, Ma’lumotlarniko‘rsatish (faqato‘qishuchun), CheckBox, NumericUpDown, Label, Button, Additional komponentasi, Panel, TabControl, System komponentasi, MenuStrip, ContextMenuStrip.

 *Bilishshartbo‘lgantushunchalar.* sinfvasinfob’ekti, xususiyat, hodisa, forma, komponentadasturlashgaoiddastlabkitushunchalarhamdaC++tiliniqo‘llab quvvatlovchimuhitdaishlashnibilishlozim.

 *Bilibolasiz.* Visual C++ muhitidasturlash, integrallashganishlabchiqarishmuhati, Toolbox, Komponentatushunchasivaxususiyatlari, Toolbox oynasini, Toolbox oynasitahrirlash, Yangikomponentalarniqo‘shish, Komponentaningxususiyatlariguruhleri, loyihaboshqaruvoyinasiorqaliko‘rish, Komponentaningusullari (methods) vahodisalari (events), Standart komponentasiningxususiyatlarivahodisalari, Toolbox oynasidagi [Common Controls] Tab, Standartkomponentalarniguruhleri, komponentalarnioynagajoylashtirish, Additional komponentasiningxususiyatlarivahodisalari, Additional komponentalariningguruhleri, Panel komponentasi, groupBox komponentasi, tabControl komponentasi, splitContainer komponentasi,

tableLayoutPanel komponentasi, flowLayoutPanel  
 komponentasikabilarningxususiyatlariningvazifalari, System  
 komponentasiningxususiyatlarivahodisalari, System  
 komponentalariningguruhleri, Menyuvauskunalarpanellarini,  
 Components tab komponentalari, contextMenuStrip  
 ob'ektio'rnatishvakomponentalarbilanbog'lashusullari,  
 shuningdekba'zibirxususiyatlarivahodisalario'rganishingiz mumkin.

## REJA

1. Komponentatushunchasivaxususiyatlari.
2. Standartkomponentalariningxususiyatlarivahodisalari.
3. Additionalkomponentalariningxususiyatlarivahodisalari.
4. System komponentalariningxususiyatlarivahodisalari.

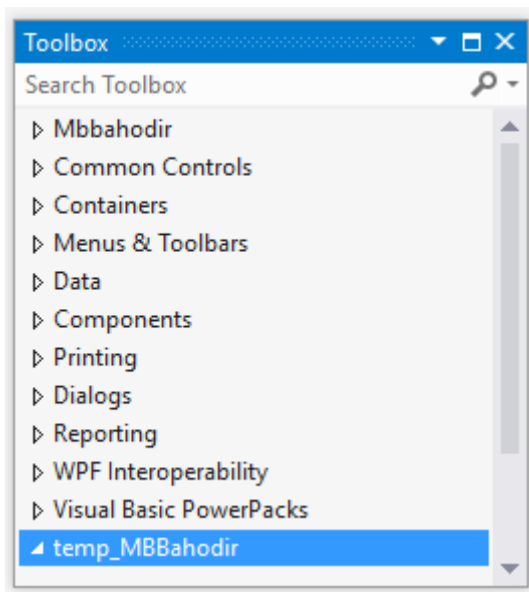
## KIRISH

MS Visual Studio  
 daishlashtamoyliningasosiyqulayliklaribukomponentalaridir.  
 Foydalnuvchining UI nitahlilqilibqaralsa,  
 undajudako'plabkomponentalarjoylashtirilgan. Bujudaosondir. MS Visual  
 Studio Visual C++  
 daturlivazifalarniamalgaoshirishkomponentalarimuvjud. Ularniasosan  
 Toolbox debhamyuritiladi. Toolbox  
 dagiharbirkomponentaningnomlarfazosivasinflarimavjud.  
 Ulardanfoydalangandashusinflardanmerosxo'rolibyaratiladivabiroynada  
 birxiltipdagikomponentadanbirnechtasidanfoydalanishmumkin.

**Komponentatushunchasivaxususiyatlari.** Komponentabu MS  
 Visual Studio ningengasosiyquolidir.  
 Dasturchibularorqalitez dasturlashva IDE  
 muhitningimkoniyatlaridanfoydalaniladi. Komponentalarto'plamini  
 Toolbox debyuritiladi.

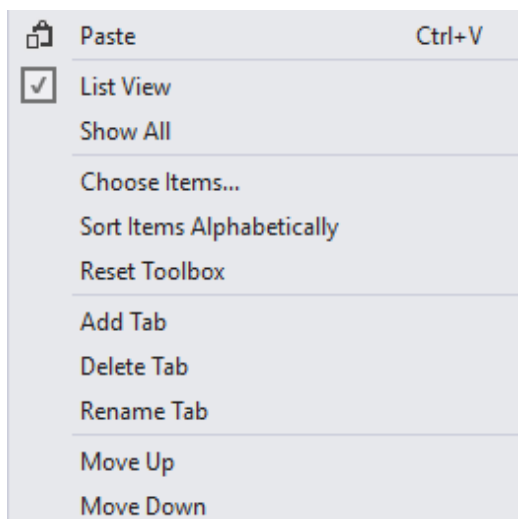
Toolbox  
 interaktivoy nasimuhitningixtiyoriyjoyidajoylashganbo'lishimumkin.  
 Loyihalovchiunio'zigamosqilibjoylashtirishmumkin.  
 Agarmuhitningoynasida Toolbox oynasiniko'rmayotganbo'lsangiz,  
 menyudanfoydalanib, [view → Toolbox] buyruqlaribajaring. [Ctrl + W,  
 X] tugmachalarmajmuasinihambosishorqali Toolbox  
 oynasigao'tish/chaqirishmumkin.  
 Muhitkomponentalarsoninechtadegansavolgajavobberishdanboshlaymiz.  
 Muhitintegrallashganmuhitliginibilamiz,

shuninguchunundagikomponentalarsonioldindanxechkimaytaolmaydi. Yangikutubxonalarinjoriyqilishyangikomponentalarniolibkiradi. Standartfoydalanuvchilaruchun Toolbox oynasi 10.1-rasmdatasvirlangan.



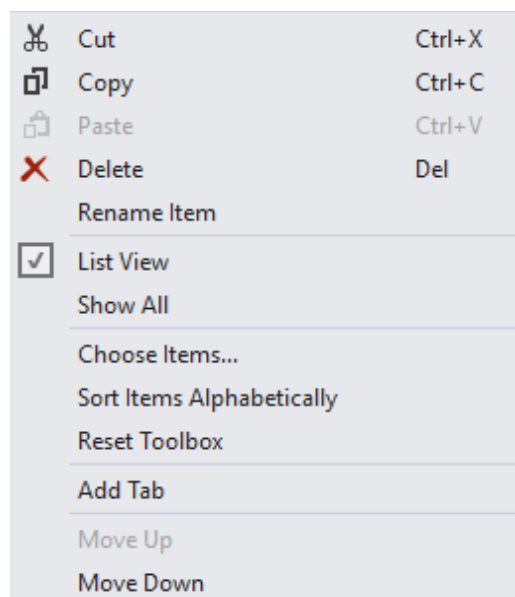
10.1-rasm. Toolbox oynasi.

Toolbox oynasitahrirlashmumkin. Buninguchunsichqonchanningo‘ngtugmasini Tab ga, ya‘nikomponentalarguruhiningnomigabosib, 10.2-rasmdagikontekstmenyuni, Tab ningichigakiribkomponentaniustigabossak 10.3-rasmdagikontekstmenyunchaqiramiz.



10.2-rasm.

komponentalarguruhiuchunkontekstmenyu.



10.3-rasm.

Komponentauchunkontekstmenyu.

Rasmlarga qarash, ulardabirxilnomlibuyruqlaruchraydi, bularbutun Toolbox oynasiuchunxizmatqiladiva 5 tabo'limdaniborat. Harbo'limgavauningbuyruqlarigato'xtalibo'tamiz:

1- Bo'lim.

Komponentalarustidaamal bajarish uchun mo'ljallangan bo'lib, [Cut] – komponentani buferga olish va joyidano'chirish uchun, [Copy] komponentaning nusxasini olish uchun, [Paste] – komponentaning olingan nusxasini joylashtirish, buni komponenta guruhiga qo'llash mumkin, [Delete] – komponenta ni guruhdano'chirib tashlash, [Rename item] – komponenta ni o'zgartirish.

2- Bo'lim. Toolbox oynasiga xizmatqiladi. [List View] – tanlagan guruh komponentalarini ro'yxatqilib chiqaradi, o'chirilgan bo'lsa, uskuna larto'plamisifatidako'rsatadi. [Show All] – muhitdagibarcha faollashtirilgan komponentalar chiqarib beradi. komponentalar judako'pligi uchun bu ishlatisho'tamush kulish deb hisoblayman. Tajribamda men hamba larning hammasini ishlatolmaganman.

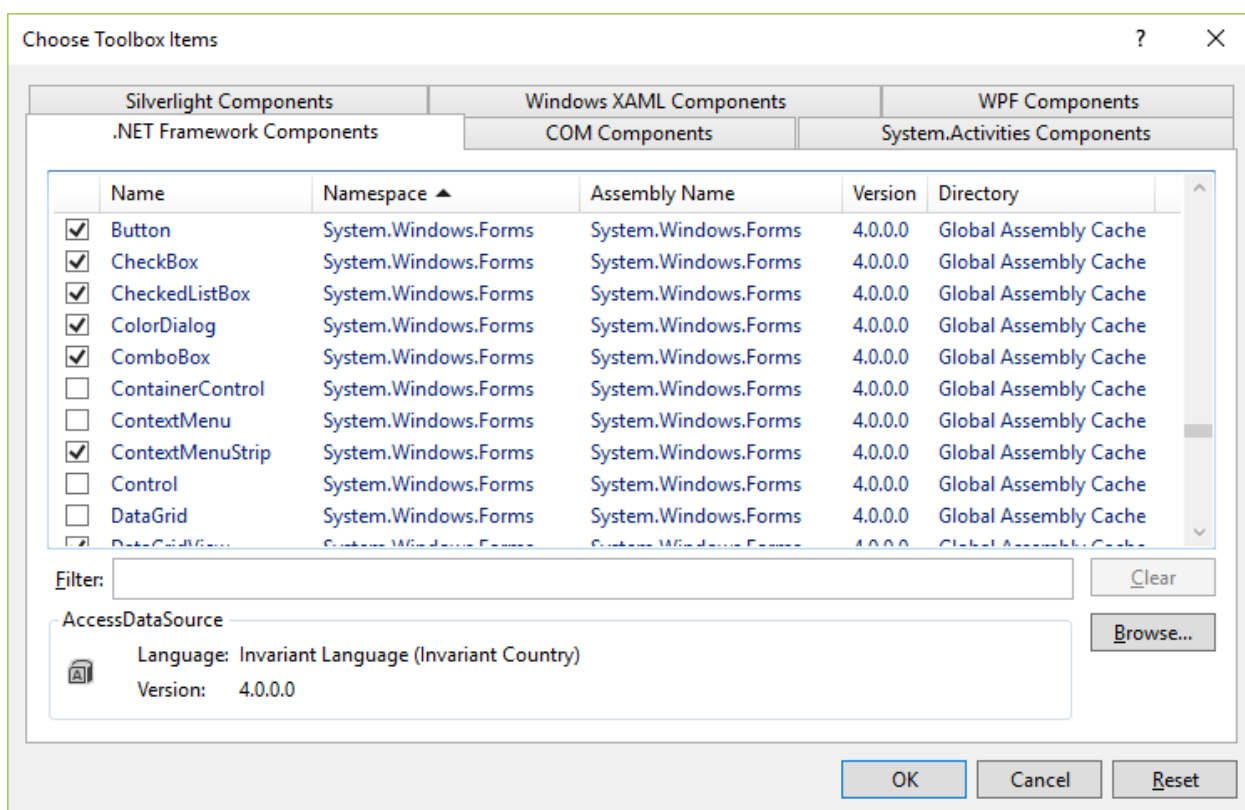
3- Bo'lim. Toolbox oynasiga xizmatqiladiva komponentalar hosil qilish uchun ishlatiladi. [Choose Items..] – yangi komponentalarni qo'shish uchun ishlatiladi, [Sort Items Alphabetically] – komponentalarni alfavit bo'yicha saralash, [Reset Toolbox] – Toolbox komponentalarni sozlamalarini standart shaklgakeltirish.

4- Bo'lim. Toolbox oynasiga Tab larni, yanikomponenta guruhlarini ustida ishlatishimkoniyatini beradi. [Add Tab] – yangikomponenta guruhini qo'shish, [Delete Tab] – komponenta guruhini o'chirish, [Rename Tab] – komponenta guruhini qaytanomlash, standart nomlaganlarini hamba malarbilano'zgartirish mumkin.

5- Bo'lim. komponenta guruhivakomponentalarning o'rinlarini almashtirish uchun ishlatiladi. [Move Up] – yuqorigako'chirish, [Move Down] – pastgako'chirish.

Bularning barchasi muhitning dasturchi gamoslashuvchanligini bildiradi. Yangikomponentalarni qo'shishko'rib chiqaylik, bubuyruq bosilish bilan 10.4- rasmda tasvirlangan muloqot oynasini chiqadi.





10.4-rasm. Yangikomponentalarniqo‘shish.

Burasmdankomponentalarningturlariko‘rishmumkin, harbirbo‘limi 1000 danortiqsinflargaega. Harbirsinfesa, bittakomponentahisoblanadi. Agartizimdama vjudbo‘lmagankomponentalarnihamshuoyanaorqaliqo‘shi shmumkin, masalan, MySql, oqimlarbilanishlash, OpenCV, OpenMP, Fuzzy ToolBox kabikomponentalarniqo‘shishmumkin.

**Komponentaningxususiyatlari.** Harbirkomponentaningshundayxu susiyatlaribor.

Eslabko‘ringOYDdasinfniyaratishvaqtidanimalaryaratilaredi.

Xuddishuningdek, harbirkomponentasinf, uningxususiyatlari (properties), usullari (methods) vahodisalari (events) mavjud.

Komponentaningxususiyatlariquyidagiguruhlargabo‘linadi (10.1-jadvalgaqarang)

10.1-jadval. Komponentaningxususiyatlari

1	<b>Accessibilit y guruhi</b>	Buguruhdagixususiyatlarbirkomponentadao‘zgarishlarn ianiqlashvao‘rnatishuchunishlatiladi.
2	<b>Appearanc eguruhi</b>	Komponentaningtashqiko‘rinishiuchunxususiyatlarnio‘r natishgamo‘ljallangan.
3	<b>Behaviorgu ruhi</b>	Komponentaningrejimxususiyatlarinio‘rnatishgamo‘ljall angan.
4	<b>Dataguruhi</b>	Komponentaniichkivatashqima’lumotlarbilanbog‘lashx ususityayalariuchunishlatiladi.
5	<b>Designguru</b>	Komponentanilo yihalashxususiyatlariuchunishlatiladi

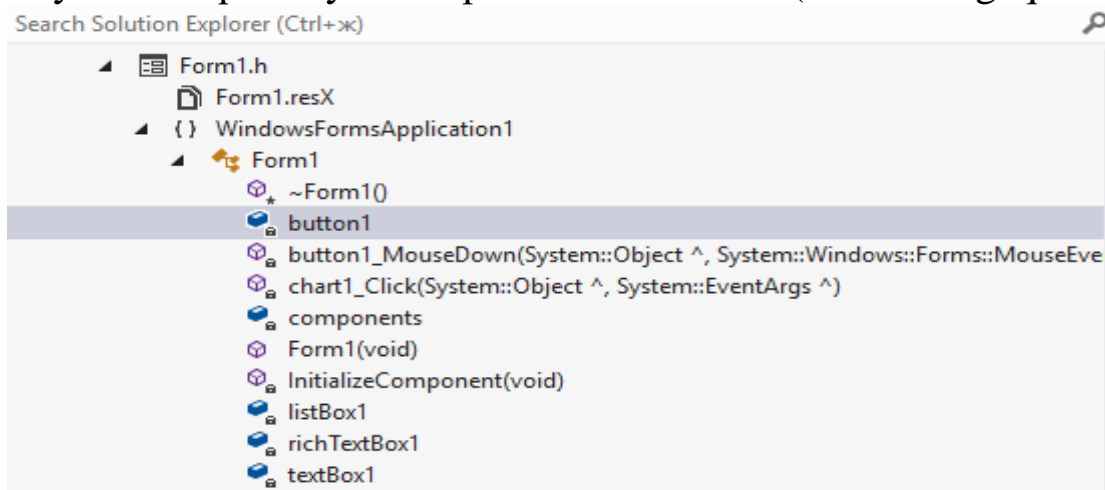
	<b>hi</b>	
6	<b>Focusguruhi</b>	Komponentadafokuslarniboshqarishxususiyatlariuchunishlatiladi
7	<b>Layoutguruhi</b>	Komponentanitartiblashxususiyatlariuchunishlatiladi
8	<b>Miscguruhi</b>	Boshqaxususiyatlarnio‘rnatish, komponentaniturigaqarabo‘zgaribturadi.

Buxususiyatguruhlarikomponentatipigamoslashganholdadoimiyo‘zgaribturadivatipigamosxususiyatlarnianiqlabberadi.

Komponentaxususiyatlargaqiymatlarmaxsus, to‘plamdantantlash, sonli, matnli, mantiqiykabima’lumtiplarqimatinioladi. Agarmurakkabtipbo‘lsa, albattamuloqotoynasiorqalitanlashmumkin.

Komponentaxususiyatlarigaqiymatlarniberishvizuataamalgaoshiriladi vaasosiyformagauningdasturfragmentlariyozibketiladi.

Buniloyihaboshqaruvoy nasiorqaliko‘rishmumkin (10.5-rasmgaqarang)



10.5-rasm. Komponentalarningqo‘shilganligi.

OynaningInitializeComponentfunksiyasidahamkomponentalarningxususiyatlariko‘shilganiniko‘rishmumkin.

```
void InitializeComponent(void)
{
    this->button1 = (gcnew
System::Windows::Forms::Button());
    this->listBox1 = (gcnew
System::Windows::Forms::ListBox());
    this->SuspendLayout();
    //
    // button1
    //
    this->button1->Location =
```

```

System::Drawing::Point(208, 73);
    this->button1->Name = L"button1";
    this->button1->Size =
System::Drawing::Size(75, 23);
    this->button1->TabIndex = 0;
    this->button1->Text = L"button1";
    this->button1->UseVisualStyleBackColor
= true;
    this->button1->MouseDown += gcnew
System::Windows::Forms::EventHandler(this,
&Form1::button1_MouseDown);
    //
    // listBox1
    //
    this->listBox1->FormattingEnabled =
true;
    this->listBox1->Location =
System::Drawing::Point(363, 62);
    this->listBox1->Name = L"listBox1";
    this->listBox1->Size =
System::Drawing::Size(120, 95);
    this->listBox1->TabIndex = 1;

    // Form1 ...
    //

}

```

Komponentaningusullari (methods) vahodisalari (events) quyidagiguruhlargabo‘linadi (10.2-jadvalgaqarang)

10.2-jadval. Komponentaningusullari (methods) vahodisalari (events)

1	<b>Actionguruhi</b>	Komponentauchunglobalhodisalarfaollashtirishvama’lumbiralgoritmyozish
2	<b>Appearanceguruhi</b>	Komponentaningichkihodisalariuchunishlatiladivama’lumbiralgoritmyozish
3	<b>Behavior</b>	Komponentaholatlariuchunhodisalarnifaollashtirish

	<b>guruhi</b>	
4	<b>Dataguruhi</b>	Ichki vatashqima'lumotlar bilan ishlash hodisalarini faollashtirish
5	<b>Drag drop guruhi</b>	Komponentaning harakatlanish hodisalarini faollashtirish
6	<b>Focus guruhi</b>	Komponentada fokuslarni boshqarish hodisalarini faollashtirish
7	<b>Key guruhi</b>	Komponentaning mostugmalar bilan ishlash hodisalarini faollashtirish
8	<b>Layout guruhi</b>	Komponentada atrofidagi elementlar bilan ishlash tartibini nazora qilish hodisalarini faollashtirish
9	<b>Mouse guruhi</b>	Komponentada sichqoncha bilan bo'ladigan hodisalarini faollashtirish
10	<b>Property Changed guruhi</b>	Komponentaning xususiyatlarini o'zgartirish hodisalarini faollashtirish

Bu usullar va hodisalarning odatdagi hammasi ham vizuallash imkonini beradi. Shuning uchun agar komponentaning biror funksiyasi bilan boshlab kiritilgan usul, metod, sifat bilan tugagan funksiyalar bo'lsa, bular ko'proq hodisalar hisoblanadi.

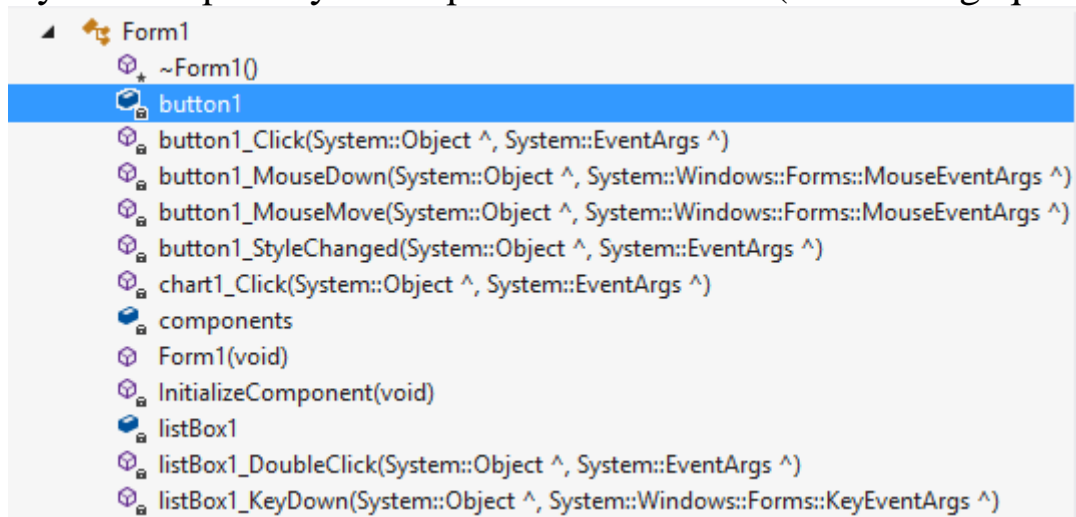
Guruhlar komponenta tipiga moslashgan holda doimiy o'zgarib turadigan tipga mos hodisalarini aniqlab beradi.

Komponenta hodisalarini funksiyalar sifatida yaratiladigan dasturchi uchun kerakli o'zining algoritmini yozadi.

Komponenta hodisalarini o'z shish vizual amalga oshiriladigan asosiy formaga uning dastur fragmentlarini yozib ketiladi.

Amma dasturchi o'z funksiyalarining algoritmini yozish kerak.

Buni loyihani boshqaruv yo'nalishi orqali ko'rish mumkin (10.6-rasmga qarang)



## 10.6-rasm. Komponentalarning hodisalari.

Oynaning InitializeComponent funksiyasida ham komponentalarning hodisalariko 'shilganiniko' rish mumkin.

```
void InitializeComponent(void){
    this->button1 = (gcnew
System::Windows::Forms::Button());
    this->listBox1 = (gcnew
System::Windows::Forms::ListBox());
    this->SuspendLayout();
    //
    // button1
    //
    // xususiyatlar
    this->button1->Click += gcnew
System::EventHandler(this, &Form1::button1_Click);
    this->button1->MouseDown += gcnew
System::Windows::Forms::MouseEventHandler(this,
&Form1::button1_MouseDown);
    this->button1->MouseMove += gcnew
System::Windows::Forms::MouseEventHandler(this,
&Form1::button1_MouseMove);
    this->button1->StyleChanged += gcnew
System::EventHandler(this,
&Form1::button1_StyleChanged);
    //
    // listBox1
    //
    // xususiyatlar
    this->listBox1->DoubleClick += gcnew
System::EventHandler(this,
&Form1::listBox1_DoubleClick);
    this->listBox1->KeyDown += gcnew
System::Windows::Forms::KeyEventHandler(this,
&Form1::listBox1_KeyDown);
    //
    // boshqa komponentalar
    // Form1
    // ... xususiyatlar
```

```

// hodisalar
this->Controls->Add(this->listBox1);
this->Controls->Add(this->button1);
// ... xususiyatlar
}
private: System::Void
button1_MouseDown(System::Object^ sender,
System::Windows::Forms::MouseEventArgs^ e) { }
private: System::Void
listBox1_DoubleClick(System::Object^ sender,
System::EventArgs^ e) {}
private: System::Void
listBox1_KeyDown(System::Object^ sender,
System::Windows::Forms::KeyEventArgs^ e) { }
private: System::Void
button1_MouseMove(System::Object^ sender,
System::Windows::Forms::MouseEventArgs^ e) { }
private: System::Void button1_Click(System::Object^
sender, System::EventArgs^ e) { }
private: System::Void
button1_StyleChanged(System::Object^ sender,
System::EventArgs^ e) { }

```

Shuningdek,

baʼzi komponentalar faol va faol boʻlmagan holatlarda boʻlishi mumkin.

Bu huddi windows

oynalarida gibuyruqlarning foydalanishiga oʻxshash ishlatiladi.

Komponentalarni oʻz joyida ishlatishni ham bilish kerak va ularni qachon ishlatish kerakligini bilish ham muhim.

### Standart

**komponentasining xususiyatlarivahodisalari.** Standart komponentalarga Toolbox oynasidagi [Common Controls] Tab dagi komponentalarni kiradi.

Ularni tarkibini oʻzgartirish mumkin,

yaʼni yangilarini qoʻshish va keraksizlarini oʻchirib tashlash mumkin.

Shuningdek,

agar maxsus ishlab chiqilgan yangi komponentalarni guruh boʻlsalardan ham foydalanish mumkin.

Standart komponentalarni quyidagi guruhlariga boʻlinadivashuasosidavularni ularning xususiyatlarivahodisalari 95% bir xil boʻladi.

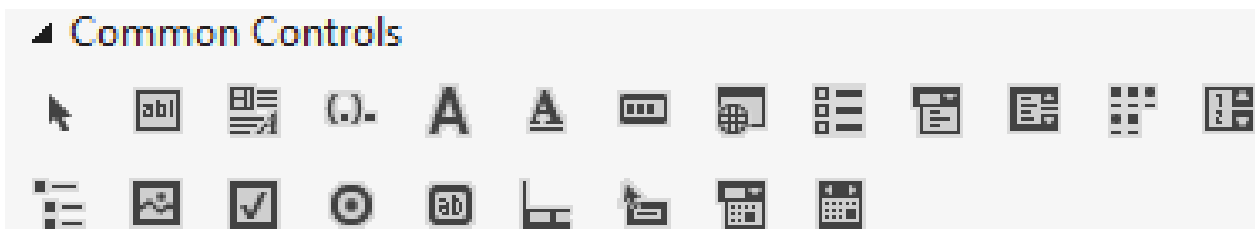
10.3-jadval. Standart komponentalarni guruhlarivavazifalari.

Komponentaning guruhi	Komponentaning nomi	Vazifasi va izoh
Mantlarni tahrirlash	TextBox	Matn tahrirlash imkonini beradi, dastur ishlayotgan vaqtda foydalanuvchi yoki algoritm yordamida matn tahrirlash.
	Rich TextBox	Oddiy va RTF formatida matn tahrirlash imkonini beradi, dastur ishlayotgan vaqtda foydalanuvchi yoki algoritm yordamida matn tahrirlash.
	Masked TextBox	Foydalanuvchi tomonidan kiritilayotgan matnlarni maxsus belgilar bilan himoyalaydi
Ma'lumotlarni ko'rsatish (faqat o'qish uchun)	Label	Foydalanuvchi tomonidan to'g'ri dan to'g'ri tahrirlanmaydigan matn ko'rsatadi.
	Link Label	Veb-link sifatida matn ko'rsatadigan foydalanuvchi joriy matnning boshqa oyna-gayok yoki veb-saytga havola faollashtiradi.
	Progress Bar	Foydalanuvchi uchun joriy amal bajarilish jarayonini ko'rsatadi.
Veb-sahifa ko'rsatish	Web Browser	Foydalanuvchi oynasida veb-sahifalar orqali harakat qilish imkonini beradi.
Ro'yxatdantlanish	Checked ListBox	Har bir katak bilan birgabo'lgan elementlarning takrorlanadigan ro'yxatini ko'rsatadi.
	ComboBox	Qalqib chiquvchi elementlar ro'yxatini ko'rsatadi
	ListBox	Matnlarni va grafik elementlar ro'yxatini ko'rsatadi
	List View	Elementlarni to'rt xil ko'rinishdan birida ko'rsatadi. Ko'rinishlarga faqat matn, kichik piktogramma, kattalik piktogramma va batafsil ko'rinish kiradi.
	Numeric UpDown	Foydalanuvchilari yuqoriga va pastga tugmalari yordamida o'tirish mumkin bo'lgan raqamlar ro'yxatini ko'rsatadi.
	TreeView	Qo'shimcha kataklari yoki piktogrammalar bilan matn

Komponentaning guruhi	Komponentaning nomi	Vazifasi va izoh
	w	ani borat bo'lish imkon bo'lgan tugunob'ektlarining yerarxi ko'plamini ko'rsatadi.
Grafiكلarni tasvirlash	PictureBox	bitmaps va piktoqramma kabit asvir fayllarini ko'rsatadi.
parametrlil qiymat	CheckBox	Matn chunka takchavayorliqni ko'rsatadi. Odatda parametrlarni belgilash uchun ishlatiladi.
	CheckedListBox	Har bir katak bilan birga bo'lgan elementlarning kaytari ladigan ro'yxatini ko'rsatadi.
	RadioButton	Yoqilgan yoki o'chirilish imkon bo'lgan tugmanini ko'rsatadi.
	TrackBar	Foydalanuvchilarga shkala bo'ylab "Thumb" ni harakatlantirib shkala qiymatlarini o'rnatish imkonini beradi.
Buyruqlar	Button	Boshlanadi, to'xtaydi yoki jarayonni bekor qiladi.
	LinkLabel	Veb-linksifatidagi matnini ko'rsatadigan va foydalanuvchi joriy matnni bosganda boshqa oyna g'ayoki veb-sayt gahavolafaollashtiradi.
	NotifyIcon	Fonda ishlaydigan dasturni ifodalovchi vazifalarni paneli holatida barcha nomasidagi belgini ko'rsatadi.
Foydalanuvchi uchun yordam	HelpProvider	Elementlari uchun pop-up yordam oynasini yoki tez koryordam oynasini beradi.
	ToolTip	Foydalanuvchini nazorat bilan hovers nazorat maqsadida bo'yicha qisqacha tavsifini ko'rsatadigan va bir pop-up oyna beradi.
Vaqt	DateTimePicker	Foydalanuvchilarga sana yoki vaqtni tanlash imkonini beradigan grafik taqvimni ko'rsatadi.
	MonthCalendar	Foydalanuvchilarga sana qatorini tanlash imkonini beradigan grafik taqvimni ko'rsatadi.

Standart komponentalarni guruhlashning asosiy maqsadi, guruhlarga mos xususiyatlar va hodisalarini birlashtirish bo'ladi. Bu komponentalarning Toolbox oynasida quyidagicha ko'rinishga ega (10.7-rasmga qarang)





10.7-rasm. Standartkomponentalarko‘rinishi.

Bu komponentalarni oynaga joylashtirish uchun sichqonchaning chap bittasiga oynaga keltirib chaptomonidan birmartaba oynaga data nlangan komponentaning nusxasini yaratib beriladi.

Bu nusxani ob'ekt deb yaratamiz.

Chunki eslab ko'ring,

sinfning nusxasini ob'ekt deb yaratiladi.

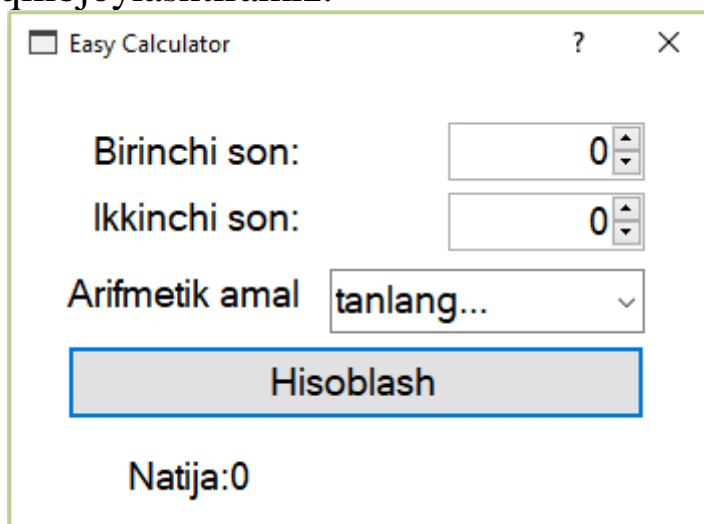
Ob'ektning ustiga sichqonchaning o'ng tugmasini bossangiz, u bilan bajarish mumkin bo'lgan amallarni ko'rasiz.

Agar ob'ekt dan nusxa olsangiz uning supersinfining, ya'ni nuqta ob'ektini yaratib beriladi.

Ikki tabir xil tipdagi ob'ektlarning xususiyatlarini vahodisalariga alohida alohida ishlov beriladi.

Standart komponentalarni xususiyat vahodisalarini ko'rib chiqish uchun narifmetik amallarni bajaruvchi dastury yaratish masalasini olamiz.

Mantiqiy jihatdan sonlarni kiritish uchun 2 ta NumericUpDown, matnli yozuvlar uchun 4 ta Label, hisoblash uchun bitta Button, amallar uchun bitta CheckBox komponentlarini olamiz va oynaga quyidagi 10.8-rasmda kildirib joylashtiramiz.



10.8-rasm.

Standart komponentalarni xususiyat vahodisalaridan foydalanish.

Bu masalani loyiha sifatida amalga oshirish uchun quyidagi qadamlar bajariladi:

**1-qadam.** Formaning xususiyatlarini o'rnatish:

BackColor

xususiyatiga oq rang; font xususiyatiga 14 o'lchamli yozuv; Icon

gamaxsusikinka; MaximizeBox xususiyatiga false qiymat; MinimizeBox xususiyatiga false qiymat; Size xususiyatigayangio‘lchamlar; StartPosition–CenterScreen holati; Text xususiyatiga “Easy Calculator” matnni yoziladi;

Buniqandayajratishshummumkin, ya‘nio‘zgarganxususiyatlarni, qiymatiningmatniyog‘onqorarangakirganlario‘zgartirilganhisoblanadi. Buo‘zgarishlarnidasturfragmentko‘rinidakeltiishhammumkin.

Budasturfragmentiniqo‘ldayozishkerakemas, ammotushunarlibo‘lishiuchunkeltiramiz.

BuniInitializeComponentfunksiyasiichidakerakliizohlarbilankeltirilganbo‘ladi.

```
this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
this->ClientSize = System::Drawing::Size(351, 225);
this->Font = (gcnew System::Drawing::Font(L"Microsoft
Sans Serif", 14.25F,
System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->HelpButton = true;
this->Icon = (cli::safe_cast<System::Drawing::Icon^
>(resources->GetObject(L"$this.Icon")));
this->MaximizeBox = false;
this->MinimizeBox = false;
this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterScre
en;
this->Text = L"Easy Calculator";
```

**2-qadam.**

Label1

ob’ektingxususiyatini o‘rnatishdasturfragmentisifatidakeltiramiz.

```
this->label1->Text = L"Birinchi son:";
```

**3-qadam.**

Label2

ob’ektingxususiyatini o‘rnatishdasturfragmentisifatidakeltiramiz.

```
this->label2->Text = L"Ikkinchi son:";
```

**4-qadam.**

Label3

ob’ektingxususiyatini o‘rnatishdasturfragmentisifatidakeltiramiz.

```
this->label3->Text = L"Natija:";
```

**5-qadam.**

numericUpDown1

ob’ektingxususiyatini o‘rnatishdasturfragmentisifatidakeltiramiz.

```
this->numericUpDown1->TextAlign =
System::Windows::Forms::HorizontalAlignment::Right;
```

**6-qadam.** numericUpDown2  
ob'ekting xususiyatini o'rnatish dastur fragmenti sifatida keltiramiz.

```
this->numericUpDown2->TextAlign =
System::Windows::Forms::HorizontalAlignment::Right;
```

7-

**qadam.** comboBox1 ob'ekting xususiyatlarini o'rnatish dastur fragmenti sifatida keltiramiz.

```
this->comboBox1->Items->AddRange(gcnew cli::array<
System::Object^ >(5) {L"[+] (qo'shish)", L"[-]
(ayirish)", L"[*] (ko'paytirish)",
L"[/] (butun bo'lish)", L"[%]
(qoldiq bo'lish)"});
this->comboBox1->Text = L"tanlang...";
```

8-

**qadam.** label4 ob'ekting xususiyatini o'rnatish dastur fragmenti sifatida keltiramiz.

```
this->label4->Text = L"Arifmetik amal";
```

9-

**qadam.** button1 ob'ekting xususiyatini o'rnatish dastur fragmenti sifatida keltiramiz.

```
this->button1->Text = L"Hisoblash";
this->button1->Click += gcnew
System::EventHandler(this, &Form1::button1_Click);
```

button1\_Click funksiyasiga quyidagi algoritmni yozamiz.

```
int one =
Convert::ToInt16(Math::Round(numericUpDown1->Value,
0));
int two = Convert::ToInt16(numericUpDown2->Value);
int result = 0;
label3->Text = "Natija:";

switch(comboBox1->SelectedIndex){
    case 0: result = one + two; break;
    case 1: result = one - two; break;
    case 2: result = one * two; break;
    case 3: result = one / two; break;
```

```

    case 4: result = one % two; break;
           default: result = 0;
                break;
}
label3->Text += result.ToString();

```

Algoritmgazohbrishshartemas, chunkiko‘pkamchiliklarimavjud, masalan, bo‘lisharifmetikamalida. Ammoba‘zixulosalarnichiqarishmumkin. [numericUpDown1->Value] ob’ektningtanlanganhaqiqiytipdagiqiymatiniqaytaruvchixususiyat, [Convert::ToInt16] – ixtiyoriytipnibutuntipgaalmashtiribberadi, [label3->Text] – ob’ektningmatnxususiyatiniio‘zgartirishimumkin, [comboBox1->SelectedIndex] – tanlagichningtanlanganindeksiniqaytaradi, [label3->Text += result.ToString();] – ob’ektningmatningamatnqo‘shishvatipnisatrlitipgao‘tkazishko‘rsatilgan.

Dasturfragmentida 3 tabelgi [::] – cinfningusullarigamurojaatqilish, [->] – ob’ektningxususiyatigamurojaatqilish, [.] ob’ektningqiymatiustidabajarilishikerakbo‘lganamal. Visual C++ dajudako‘pusul, vafunksiyalarniishlatishtamoyllaribor, ularnifaqatkompilyatorningimkoniyatigaqarabajratishmumkin.

Standartkomponentalarningba‘zixususiyatlarivahodisalari, ularuchunba‘ziusullarniishlatiyuko‘rdik.

Bularnichuqurroqo‘rganishuchunko‘proqamaliyishlarnibajarishkerak.

### **Additional**

**komponentasiningxususiyatlarivahodisalari.**Bustandartkomponentalabilashishlashuchunqo‘shimchakomponentlardir.

Bularasosanboshqaruvelementlarniguruhlashuchunishlatiladi.

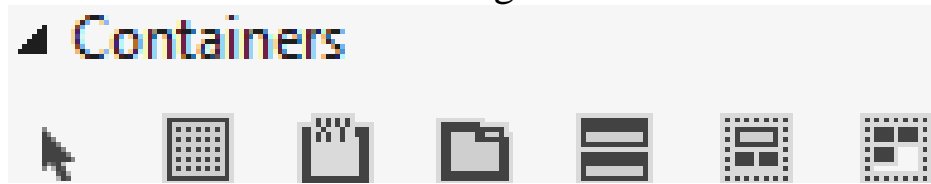
Quyidagijadvaldaqo‘shimchakomponentalarnivavazifalarinikelitiramiz.

10.4-jadval. Additional komponentalariningguruhlarivavazifalari

Komponenta-ningguruhi	Komponentaning nomi	Vazifasivaizoh
Boshqaruvla relementlarn iguruhlash	Panel	elementlarniboshqarishuchunto‘plaminiguruhlaydi
	GroupBox	Nomlanganelementlarniboshqarishto‘plaminiguruhlaydi
	TabControl	Guruhlanganob’ektlarnisamaralitashkileti shvakiritishuchunsohalisahifanitaqdimetadi.

	SplitContainer	Birbiridan ajratilgan ikki panel beradi. Splitcontainer – Splitter o‘rniga mo‘ljallangan.
	TableLayoutPanel	Kontent dinamik satr va ustunlar iborat panelni ifodalaydi.
	FlowLayoutPanel	Dinamik ravishda kontentni gorizontal yoki vertikal joylashtiradigan panelni ifodalaydi.

Bu komponentalar Containers nomli tabda joylashgan. Ularning ko‘rinishi 10.9-rasmda keltirilgan.



10.9-rasm. Containers komponentalari.

Bu komponentlarning xususiyat va hodiralari alohida to‘xtalib o‘tishimiz kerak. Vazifalarielementlariguruhiniboshqarishdan iborat, shu maqsadda ishlatiladigan xususiyat va hodisalarini keltiramiz.

**Panel** komponentasi. Panel – bu oddiy to‘rt tomonli chegaralangan hududga o‘xshaydi. Uni butun sohani egallashi uchun o‘ng tomonidagi [Panel Task] dagi [dock in parent container] ni bosish yetarli. [undock in parent container] ni tanlab, oldingi holatiga kelish mumkin.

Ob‘ektning harakatlantirish uchun chaptomondagi 4 tatomonga o‘naltirilgan tugmachadan foydalaniladi.

**AutoSize** - xususiyati true/false qiymatlarini qabul qiladi. Odatdagi qiymat ko‘proq ishlatiladi, chunki ob‘ektga joylashtirilgan boshqaruvelementlarining o‘lchamlariga mos holda o‘lchamni sozlaydi. Joriy holatda yo‘lga o‘n qiymatga ega bo‘ladi.

**Cursor** - xususiyati sichkonchani kursorko‘rinishini o‘rnatishimkonini beradi.

**BorderStyle** - ob‘ektning atrofdan chegaralarining ko‘rinishlarini o‘rnatish uchun ishlatiladi.

**AutoScroll** – ob‘ekt dabilashtirilgan elementlarni to‘liq ko‘rinishga, harakatlaguvchi tugmalarni o‘rnatish va olib tashlash uchun ishlatiladi. Bu xususiyat true/false qiymatlarini qabul qiladi.

**AutoScrollMargin** - bu xususiyat harakatlanuvchi tugmachalar chetmasofalarini o‘rnatida, Width va Height ichki xususiyatlarini beradi.

<b>AutoScrollMinSize</b>	-
buxususiyatharakatlanuvchitugmachalarengkichiko‘lchamlarinio‘rnatadi . Wight va Height ichkixususiyatlaribor.	
<b>Dock</b>	-
ob’ektnioynaningqaysiqismidajoylashtirishkerakliginio‘rnatadi.	
<b>Margin</b>	-
ob’ektchetqismlaridaqoldirilishikerakbo‘lganmasofalarnio‘rnatadi. Uning 5 taxususiyatibor, barchasi, chap, o‘ng, yuqorivapast.	
<b>groupBox</b> <b>komponentasi.</b> groupBox	-
nomlaganelementlarguruhinibirlashtirishuchunishlatiladi.	
Ob’ektniharakatlantirishuchunchaptomondagi	4
tatomongayo‘naltirilgantugmachadanfoydalaniladi.	
<b>Text</b> - buxususiyatelementlarguruhininominioynagachiqaradi	
<b>AutoSize</b> - xususiyati true/false qiymatlarniqabulqiladi. Elementlarningo‘lchamlarigamosob’ektgao‘zlchamajratadi.	
<b>BackColor</b> – ob’ektningfonxususiyatiranginio‘rnatadi.	
<b>BackgroundImage</b> - ob’ektningfonxususiyati                      Image	
ob’ektlarnio‘rnatadi.	
<b>Dock</b>	-
ob’ektnioynaningqaysiqismidajoylashtirishkerakliginio‘rnatadi.	
<b>FlatStyle</b> – ob’ektningstilxususiyatinio‘rnatadi.	
<b>Font</b>	-
buxususiyatob’ektningyozuvlariuchunaniqbirstandartyozuvnio‘rnatishuchunishlatiladi.	
<b>MaximunSize</b> – ob’ektningengkattaolchaminibelgilaydi.	
<b>MinimumSize</b> - ob’ektningengkichiko‘lchaminibelgilaydi.	
<b>Padding</b>	-
ob’ektgachabo‘lganmasofalarnio‘rnatishuchunishlatiladi.                      Uning                      5	
taxususiyatibor, barchasi, chap, o‘ng, yuqorivapast.	
<b>RightToLeft</b> – ob’ektningmatnino‘ngtomonidano‘rnatish.	
<b>TabIndex</b> –                      ob’ektningtabtugmasibosilganda,                      nechininavbatdabo‘lishinibelgilashnio‘rnatadi.	
<b>tabControl</b> <b>komponentasi.</b> tabControl	
buoddiytorrtomonidanchegaralanganhududgao‘xshaydi.	
Unigayangisahifaqo‘shishyokisaxifanio‘chirishuchuno‘ngtomonidagi [tabControl Task] dagi [Add Tab ] va[Remove Tab] nibosishyetarli.	
Ob’ektniharakatlantirishuchunchaptomondagi	4
tatomongayo‘naltirilgantugmachadanfoydalaniladi.	

## **Anchor**

ob'ektniaftoryokioynabo'yiyaatartiblabjoylashtirishxususiyatlarinitanlay di. Uningto'rttaqiymatio'ng, chap, yuqori, pastboryuBularningo'zaroalashuvidanfoydalaniladi.

## **Appearance**

buxususiyatob'ektningtugmachalariningko'rinishinio'rnatishuchunishlatiladi. Uningqiymatlari [Normal], [Buttons], [Flat Buttons] qiymatlarniqabulqiladi.

## **DrawMode**

ob'ektningxususiyatinichizilganformasinio'zgartrishuchunishlvtiladi.

## **TabPage**

buxususiyatto'plamqiymatlarniqabulqiladivauningnechtasahifabo'lishini ta'minlaydi.

Ubilanishlagandaharbirsahifaningalohidaxususiyatlarigaishlovberishmu mkin.

## **TabPage1**

buob'ektningsaxifaxususiyatlarnio'rnatishuilatiladigganob'ekthisoblana divapanelob'ektidekxususiyatlargaega.

## **tabStop**

ob'ektbo'yichatabtugmasiningharakatlanishio'rnatishvao'chirishqiymati nio'rnatadi.

## **splitContainer**

## **komponentasi.**

## **splitContainer**

ikkigaajratilganelementlarniboshqarishuchunto'plamnihosilqiladi.

Unibutunsohaniegalashiuchuno'ngtomonidagi [splitContainer Task] dagi [dock in parent container]nibosishyetarli. [undock in parent container]nitanlab, oldingiholatigakelishmumkin. Shuningdek, [Horizontal Splitter Orientation]

ob'ektramkalarinigorizontalo'rnatadiva [Vertical Splitter Orientation] esavertikalo'rnatadi. Ob'ektniharakatlantirishuchunchaptomondagi 4 tatomongayo'naltirilgantugmachadanfoydalaniladi.

## **Anchor**

ob'ektniaftoryokioynabo'yiyaatartiblabjoylashtirishxususiyatlarinitanlay di. Uningto'rttaqiymatio'ng, chap, yuqori, pastboryuBularningo'zaroalashuvidanfoydalaniladi.

**BackColor** - ob'ektningfonxususiyatiranginio'rnatadi.

## **BorderStyle**

ob'ektningatrofchegaralariningko'rinishlarinio'rnatishuchunishlatiladi.

## **FixedPanel**

ob'ektuchunasosiypanelnio'rnatishuchun, ya'niikkitanpanelndanbirasosiybo'lishikerak.

**isSplitterFixed** – ob'ekt uchun asosiy panelni o'rnatish amal qilish yoki qilmasligini o'rnatadi. xususiyati true/false qiymatlarini qabul qiladi.

**Orientation** – ob'ektning panellarini horizontal yoki vertikal o'rnatish uchun ishlatiladi. Ikkita Horizontal va Vertical qiymat qabul qiladi.

**splitContainer1.Panel1** – ob'ektning panel xususiyatlariga ishlov berish uchun panel xususiyatlarini o'chib beradivau uchun xususiyatlarni o'rnatish mumkin.

**Panel1.Collapsed** – ob'ekt dapanel 1 ni to'liq qilib o'rnatadi. Bunday xususiyat barcha panellari uchun o'rnatiladi.

**Panel1.MinSize** – ob'ekt dapanelning eng kichik o'lchamini o'rnatishdan iborat.

**RightToLeft** – ob'ekt panellarini o'ngdan o'rnatish xususiyati.

**tableLayoutPanel** **komponentasi.** `tableLayoutPanel` – bu jadval kabitartiblangan panellarni o'rnatish uchun elementlarni birlashtirish o'byektidir.

**AutoScroll** – ob'ekt dabilashtirilgan elementlarni to'liq ko'rinmasa, harakatlaguvchi tugmalarni o'rnatish va olib tashlash uchun ishlatiladi. Buxususiyat true/false qiymatlarini qabul qiladi.

**AutoScrollMargin** – bu xususiyat harakatlanuvchi tugma chetmasofalarini o'rnatida, Width va Height ichki xususiyatlaribor.

**AutoScrollMinSize** – bu xususiyat harakatlanuvchi tugma chetmasofalarini eng kichik o'lchamlarini o'rnatadi. Width va Height ichki xususiyatlaribor.

**CellStyle** – ob'ektning yacheykalarining chet chegaralarining stilini belgilaydi.

**ColumnCount** – ob'ekt dagi ustunlar sonini o'rnatish.

**Columns** – ob'ekt ustunlari uchun xususiyatlarni o'rnatish uchun alohida muloqot oynasi ochiladi.

**GrowStyle** – ob'ektga ustun yoki qator yoki ikkalasini qo'shish xususiyatini o'rnatish.

**RowCount** – qatorlar soni

**Rows** – ob'ekt qatorlari uchun xususiyatlarni o'rnatish uchun alohida muloqot oynasi ochiladi.



**FlowLayoutPanel**komponentasi. **FlowLayoutPanel** – ketmaketlikkabitartiblanganpanellarto‘plamiorqalielemlarnibirlashtiri shob’ektidir.

**Anchor** - ob’ektniaftoryokioynabo‘yiyaatartiblabjoylashtirishxususiyatlarinitanlay di. Uningto‘rttaqiymatio‘ng, chap, yuqori, pastboryuBularningo‘zaroalashuvidanfoydalaniladi.

**BackColor** - ob’ektningfonxususiyatiranginio‘rnatadi.

**AutoSize** - xususiyati true/false qiymatlarniqabulqiladi. Elementlarningo‘lchamlarigamosob’ektgao‘zlamajratadi.

**WrapContents**

ob’ektdasontentlarniketmaketliginiustuvorqilishyoki qilmasliknio‘rnatadi . Xususiyat true/false qiymatlarniqabulqiladi.

Buqo‘shimchakomponentalarningxususiyatlarohaqidama’lumotlar niberildi. Ularninghodisalaribirbirigao‘xshashbo‘lib, standartkomponentalarningdanfarqjudakam.

Shuninguchunhodisalarigato‘xtalaibo‘tirmaymiz. Shuningdek, qo‘shimchakomponentalarjudako‘p, buniko‘rishuchun Tollbox oynasigasichqonchaningog‘ngtugmasinibosingva [Shown All] buyrug‘initanlang.

**System**

**komponentasiningxususiyatlarivahodisalari.**Bukomponentalargaasosa noperatsiontizimbilanishlaydigankomponentalarkiradi.

Ularnixususiyatlarimurakabhisoblanadi.

Chunkibirginamenyuyaratishuchunstandartvaqo‘shimchakomponentalar hamfoydalaniladi.

Shuninguchunumumiyholdabukomponentalarningvazifalarinikeltiribo‘ta miz.

10.5-jadval. System komponentalariningguruhlarivavazifalari

Komponenta ningguruhi	Komponentaning nomi	Vazifasivaizoh
Bo‘yruqlar	ToolStrip	Microsoft Windows, Microsoft Office, Microsoft Internet Explorer bo‘lishimumkinuskunalmajmuasiyokifoydalanuvchiinterfeysiniyaratadi. ToolStrip control Asboblarpanteliboshqaruvinialmashtirishgamo‘lj allangan.

	ToolStrip Container	ToolStrip elementlarini ajralmasto'plamini beradi (birikib, sohasida gorizontal yoki vertikal almashish)
Ma'lumotni a'ksettirish (faqato'qishelementlari uchun)	StatusStrip	Windowsning holat satrin beradi. Statusstrip - Statusbar o'rnini bosadi. Statusstrip maxsus xususiyatlar tartibini o'z ichiga oladi,
Menyuni boshqarish elementlari	MenuStrip	Forma uchun menyutizimini beradi. Menustrip - Mainmenu o'rnini bosadigan yuqoridara jali konteyner hisoblanadi. Bundantashqari, asosiy ishlov berish va hujjat interfeysi imkoniyatlarini beradi. Funktsionaligi ToolStripitem olingan bo'lsa-da, ToolStripdropdownitem va ToolStripmenuItem bilan birga ishlatiladi.
	ContextMenuStrip	Kontekst menyuni ifodalaydi. Contextmenustrip bu Contextmenu o'rnini bosadi. Contextmenustrip bilan birga ishlatiladi va sichqonchaning o'ng tugmasini bosishda avtomatik ravishda kontekst menyuni amoyon bo'ladi.



10.10-rasm.

Menyular va skunalar panellarini yaratish uchun komponentlarning ko'rinishi

Menyular va skunalar panellarini yaratish uchun komponentlardan tashqari tizim bilan ishlaydigan komponentlarga quyidagi 10.6-jadvaldagi komponentlar ham qiradi, ular Components tab ga kiritilgan.

10.6-jadval. Components tab ning komponentlari.

Komponentning nomi	Vazifasi va izoh
<b>Backgr</b>	Alohida ajratilgan oqim uchun amal bajariladigan

<b>BackgroundWorker</b>	BackgroundWorker komponentaning nusxasini yaratadi
<b>DirectoryEntry</b>	Active Directory xizmat bilan o'zaro bog'lanib birlashilgan va uning ierarxiyasidan o'z ehtiyoqlarini qamrab olish uchun DirectoryEntry komponentaning nusxasini yaratadi
<b>DirectorySearcher</b>	Active Directory da so'rovlarini bajarish imkonini beradigan DirectorySearcher komponentaning nusxasini yaratadi
<b>ErrorProvider</b>	Formada elementni boshqarishda xato foydalanuvchi gako'rsatish uchun ErrorProvider komponentaning nusxasini yaratadi
<b>EventLog</b>	Tizimda foydalanuvchining hodisalar jurnaliga bog'lanish imkonini beradigan EventLog komponentaning nusxasini yaratadi
<b>FileSystemWatcher</b>	Ruxsat berilgan papkavafayllarga o'zgartirishlarni kiritish imkonini beradigan FileSystemWatcher komponentaning nusxasini yaratadi
<b>HelpProvider</b>	Tezkor foydalanuvchining yordam oynasini chaqirish imkonini beruvchi HelpProvider komponentaning nusxasini yaratadi
<b>ImageList</b>	Image ob'ektlarining to'plamini boshqarish uchun ImageList komponentaning nusxasini yaratadi
<b>MessageQueue</b>	Navbatdagi xabarlar ni o'qish imkoniyatini beruvchi MessageQueue komponentaning nusxasini yaratadi
<b>PerformanceCounter</b>	Windowsning hisoblagichlar bilan ishlash uchun foydalaniladi, yangi toifalar va misollarni yaratish, hisoblagichlardan qiymatlarni o'qish va hisoblagich ma'lumotlariga asoslangan hisob-kitoblarni bajarish uchun PerformanceCounter komponentaning nusxasini yaratadi
<b>Process</b>	Tizimda jarayonlar bilan bog'liq ma'lumotlarni o'xtatish, ishgatish va o'zgartirish uchun Process komponentaning nusxasini yaratadi
<b>SerialPort</b>	sinxron va voqea imkoniyatlar uchun SerialPort komponentaning nusxasini yaratadi
<b>ServiceController</b>	Mavjud xizmatlarni, shu jumladan, boshlang'ich va o'xtatish xizmatlarini boshqarish uchun buyruqlar berish uchun ServiceController komponentaning nusxasini yaratadi
<b>Timer</b>	Windows

ilovalar uchun vaqtga asoslangan vazifalarni kiritish uchun Tim er komponentaning nusxasini yaratadi
---



10.11-rasm. Components tab gakiritilgan komponentalarning ko‘rinishi.

Bu komponentalardan kontekst menu yaratish va boshqa ob’ektlarga o‘rnatishni ko‘rib chiqamiz.

Masala. Oynada ikki kontekst menu yaratish kerak. Birinchi oyna uchun, ikkinchisi RichTextBox ob’ekti uchun bo‘lsin.

Birinchi kontekstda oynayopish, kattalashtirish, kichiklashtirish, yig‘ishtirish, tugmachalarni yoqish va o‘chirish bo‘yruqlari bo‘lsin. Ikkinchi kontekstda matn tahrirlash uchun buferga nusxalash, buferga nusxalash va o‘chirish, nusxa qo‘shish, ob’ektni tozalash, gorizontaal va vertikal harakatlanish tugmachalarni qurish va o‘chirish ham alla ribo‘lsin.

Yangi loyihani yaratamiz va biroi nabo‘ladi. Oynaning bir chetiga RichTextBox ni o‘rnatamiz. Menu va uskunalartabidan contextMenuStrip ob’ektidan ikki ta o‘rnatamiz. Ular formada ko‘rinmasligi mumkin. Ular loyiha ning ishchi maydoni pastki qismida joylashadi (10.12-rasmga qarang).

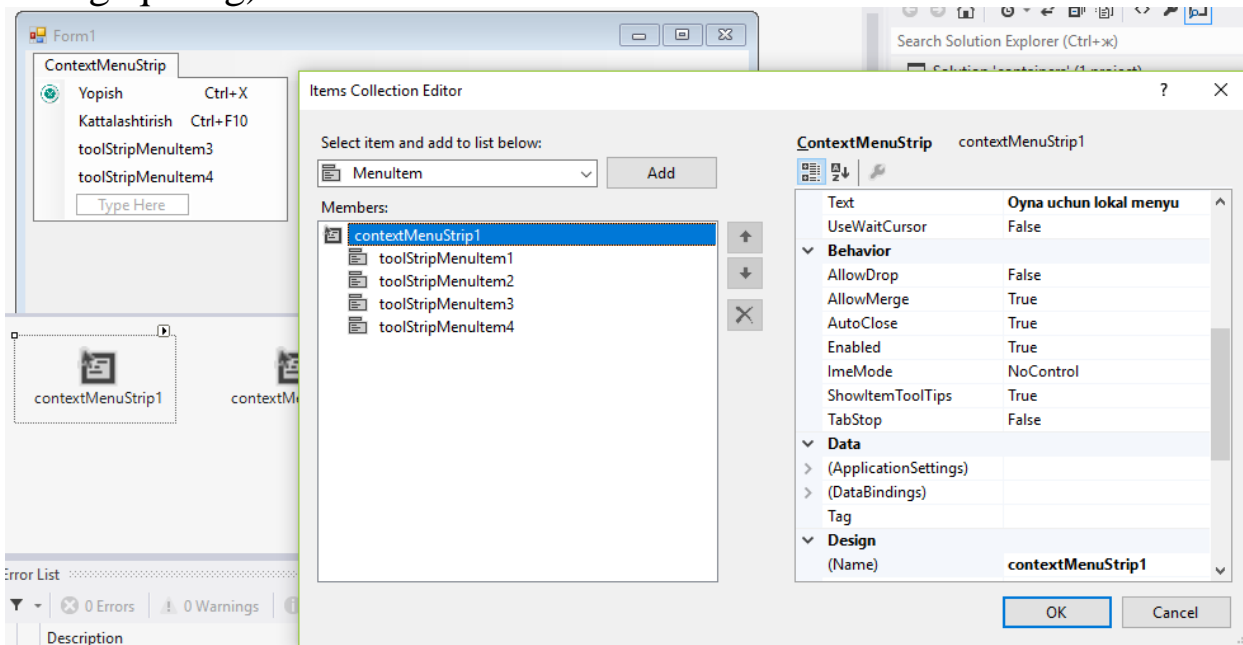


10.12-rasm. contextMenuStrip ob’ekti o‘rnatilgan ko‘rinishi.

Bularni kerakli ob’ektlar bilan bog‘laymiz.

Oynaga birinchi kontekst menu ni [contextMenuStrip1], [RichTextBox1] ob’ektiga ikkinchi kontekst menu ni [contextMenuStrip2] o‘rnatamiz. Buning uchun oynaning [ContextMenuStrip] xususiyatiga [contextMenuStrip1] niva RichTextBox1 ob’ektning [ContextMenuStrip] xususiyatiga [contextMenuStrip2] ni o‘rnatamiz. Agar bitta kontekstni ikki ta ob’ektga o‘rnatilsa, kontekstning buyruqlariga joriy qilingan algoritmlarni bajaraveradi.

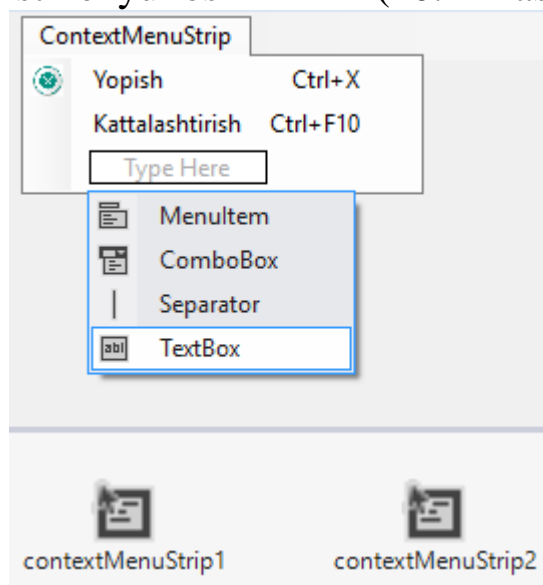
O‘rnatilgan kontekst menyularning xususiyatlarini, ya‘ni [ContextMenuStrip] xususiyatlarini o‘rnatish uchun ochib beradi. [contextMenuStrip] Buxususiyatlarini hammasiga o‘xtalib o‘tmaymiz. Faqat Items xususiyatiga o‘xtalamiz. Shu xususiyatdan langanda quyidagicha muloqot oynasini chiqadi (10.13-rasmga qarang).



10.13-rasm. Kontekst menyularni o‘rnatish.

E‘tibor qaratsangiz, muloqot oynasida qilingan o‘rnatishlar muloqot oynasida emas balki asosiy oynada ko‘rinadi. Agar bu muloqot oynasida ishlash noqulay bo‘lsa, asosiy ishchima yondaham ishlash mumkin.

Buning keraklik kontekst menyularni ochish uchun qancha bir martaba osilsa, oynada kontekst menyularni hosil bo‘ladi (10.14-rasmga qarang).



10.14-rasm. Oynadakontekstmenyugaishlovberish.

[Type Here] tugmasiorqalijoriy [MenuItem], [ComboBox], [Separator], [TextBox] - 4 taelementlarniqo'shishmumkin.

Qo'shilganharbirelementningo'ngtugmasinibosib,

ubilanma'lumbajarilishimumkinbo'lganamallaribajarishmumkin.

Shuningdek, [properties] deganxususiyatlargaolibkiradi. [Text] - xususiyatigakiribbuyruknominiberishmumkin.

Agarharfbilanmurojjaqtqilishnixoxlasangiz, kerakliharfoldiga [&] belgisiniqo'yishkerak. Inteaktivtugmao'rnatishuchunesa [ShortcutKeys] xususiyatigakeraklitugmalarkombinatsiyasinio'rnatishmumkin.

Ikonkao'rnatishuchun [Image]

xususiyatgakeraklirasnmio'rnatishmumkin.

Shundayqilibkeraklibarchabuyruqlarniamalgaoshirishmumkin.

Buyruqlarniguruhlashuchun [Separator] komponentasinitanlashkerak.

Buyruqlargaalgortimyozishshuchunularniustigasichqonchaniikkimartabo sishyetarliyokihodisalarigakiribhodisasigayozishmumkin.

Agaryuqoridakontekstmenyugaishlovberishtushunarlibo'lganbo'lsa , masalanio'zingiztugatibqo'ying.

Agarketmaketlikasosidabajarsangizalbattamasalanihechbo'lmasa, algoritmlarsizhalqilishingizlozim.



## NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1. Toolbox interaktivoyinasinimauchunkerak?
2. Qaysitugmachalarmajmuasinihambosishorqali Toolbox oynasigao'tish/chaqirishmumkin
3. Toolbox oynasitahrirlashmumkinmi? Mumkinbo'lsaqandayvanimauchun.
4. Nima Toolbox oynasiuchunxizmatqiladiva 5 tabo'limdaniborat.
5. [List View] va [Show All] buyruqlarinimauchunkerak.
6. Toolbox interaktivoyinasinechtakomponentamuvjud?
7. Toolbox interaktivoynasikomponentalariningo'rnialmashtirishvaqaytanomlash mumkinmi?
8. Komponentaxususiyatlariningasosiynechtaguruhbor.
9. KomponentalarningFocusguruhinimauchunishlatiladi.
10. Asosiyoynagaqo'shishlgankomponentalarro'yxatiniqaerdanko'ri shmumkin.

11. Oynaning InitializeComponent funksiyasida ham komponentalarni ngxususiyatlariko 'shilganiniko' rish mumkinmi?
12. Komponentaning nechta usullari (methods) va hodisalari (events) guruhibor?
13. Komponentaning Layout guruhi nima uchun ishlatiladi?
14. Standart komponentalarga Toolbox oynasida gifaysi Tab dagi komponentalarkiradi.
15. Mantlarni tahrirlash komponentalarinisanabbering?
16. Ro'yxatdantantlash komponentalarinisanabbering?
17. Komponentalarni oynaga joylashtirish uchun qanday amal bajarish kerak?
18. Label1 ob'ektining text xususiyatini o'rnatish qanday amalga oshiriladi?
19. comboBox1 ob'ektining Items xususiyatlarini o'rnatish qanday amalga oshiriladi?
20. [numericUpDown1->Value] ob'ektning tanlangan qanday tipdagi qiymatini qaytaruvchi xususiyat.
21. [comboBox1->SelectedIndex] tanlagichning tanlangan qanday qiymatini qaytaradi. —
22. Additional komponentasining xususiyatlarini va hodisalarini nima uchun ishlatiladi.
23. TabControl komponentasining vazifasini ayting.
24. Panel komponentasining xususiyatlariga misollar keltiring.
25. Panel komponentasini oynaning qaysi qismida joylashtirish kerakligini o'rnatadigan xususiyatni ayting.
26. Buxususiyat o'plam qiymatlarni qabul qiladigan va uning nechta sahifa bo'lishini ta'minlaydi. U bilan ishlaganda har bir sahifaning alohida xususiyatlariga ishlov berish mumkin. Gap qaysi komponentaning qaysi xususiyati haqida ketmoqda.
27. System komponentasining xususiyatlarini va hodisalarini sanabbering.
28. Menyuni bosh qarish elementlari uchun ishlatiladigan komponentalarni sanabbering.
29. Menyuvau skunalar panellarini yaratish uchun komponentalardan tashqari tizim bilan ishlaydigan komponentalar qaysi tab gaki kiritilgan.
30. Kontekst menyuni ishlatishni tushuntiribbering.



**AMALIYKO'NIKMAVAMALAKALARNIANIQLASHHAMDARI  
VOJLANTIRISHUCHUNASSISEMENTTOPSHIRIG'I.**

<b>ASSISEMENTTOPSHIRIG'I</b>	
	<p>Komponentalar bilan ishlashga oid berilgan quyidagi dastur bo'yicha berilgan topshiriqlar kerakli fragmentlari asosida bajarilgan.</p> <p> Bundasturda giba'zi o'zgartirishlarni topish orqali topshiriqlar bosqichma bosqich amalga oshiriladi.</p>
№ - sizning tug'ilgan kuningiz	
<p><b>1. Oddiy kalkulyator yaratilgan.</b> Loyihabo'yicha qisqacha asosiy fragmentlarni yozing.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>	
<p><b>2. Sanoq sistemalar uchunarifmetika mallarni bajaruvchi kalkulyator yaratilgan.</b> Loyihabo'yicha qisqacha asosiy fragmentlarni yozing.</p> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>	



---

---

---

---

---

---

---

---

3. Menyugaegaloyihayaring.  
Umenyuoinaxususiyatlarivahodisalariniboshqarsin.  
Loyihabo'yichaqsqachaasosiyfragmentlarniyozing.

---

---

---

---

---

---

---

---

---

---

---

4. Kontekstmenyugaegaloyihayaring.  
Umenyuoinaxususiyatlarivahodisalariniboshqarsin.  
Interaktivtugmalarvaklaviaturatugmalaribilanishlasin.  
Loyihabo'yichaqsqachaasosiyfragmentlarniyozing.

---

---

---

---

---

---

---

\_\_\_\_\_

\_\_\_\_\_

**5. Oddiyo‘yindasturiniyarating.  
Loyihabo‘yichaqisqachaasosiyfragmentlarniyozing.**

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**6. Birdasturiniyarating.  
Undaholatsatrivauskunalarpanelito‘liqishlasin.  
Loyihabo‘yichaqisqachaasosiyfragmentlarniyozing.**

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_


\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_


- |   |
|---|
| <p>7. Barchavazifalarnio‘rganganingizbo‘yichaeslabko‘ring,<br/>nimalarnio‘rgandingizvabirtakrorlang.</p> <p>8. Barchabilimvako‘nikmalarasosidakichikloyihatayyorlang.</p> |
|---|


### 3.3. Muloqotoynalaribilanishlash.


 Muloqot oynasitushunchasivaxususiyatlari, diaologtabkomponentalariiningxususiyatlarivahodisalari, messageBox vauning Show usulidanfoydlanish, uningparametrlarivaqiymatlari, shaxsiymuloqotoynalariniyaratishbo‘yichanazariyvaamaliybilimlarniha mdaasoslashuchundasturlartuzibko‘rsatilgan.

Bilimlarnimustahkamlashuchun 40  
tanazariysavolvaamaliyko‘nikmavamalakalarnirivojlantrishuchun 7  
assismenttopshirig‘iberilgan.

Butopshiriqlarnibajarishmavzunimustahkamlashuchunxizmatqiladi.

 *Kalitso‘zlar.* IDE, Visual C++, user interface, OYD, komponenta, xususiyatlari, hodisalari, Toolbox, Tab, muloqotoynasi, ColorDialog, FontDialog, OpenFileDialog, PrintDialog, PrintPreviewDialog, FolderBrowserDialog, SaveFileDialog, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, HelpNavigator, Object, DialogResult, InvalidEnumArgumentException, IWin32Window tiplari.

 *Bilishshartbo‘lgantushunchalar.* sinfvasinfob’ekti, xususiyat, hodisa, forma, komponentadasturlashgaoiddastlabkitushunchalarhamdaC++tiliniqo‘llab quvvatlovchimuhitdaishlashnibilishlozim.

 *Bilibolasiz.* Visual C++ muhitidasturlash, integrallashganishlabchiqarishmuhati, Toolbox, muloqotoynasitushunchasivaxususiyatlari, ColorDialog, FontDialog, OpenFileDialog, PrintDialog, PrintPreviewDialog, FolderBrowserDialog, SaveFileDialog muloqotoynalaribilanishlash, MessageBox sinfivauning Show funksiyasiqabulqilishimumkinbo‘lgan String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, HelpNavigator, Object, DialogResult, InvalidEnumArgumentException, IWin32Window tiplarivaqiymatlari, funksiyasiniyozilishvariantlari, muloqotoynasiniloyihaboshqaruvoynaorqaliyaratishvamuloqotoynalarig aqo‘yiladigantalablar, o‘rnatishvakomponentalaribilanbog‘lashusullari, shuningdekba‘zibirxususiyatlarivahodisalario‘rganishingiz mumkin.

## REJA

1. Visual C++ muhitidamuloqotoynalari.
2. Muloqotoynalarinisoqlash.
3. Muloqotoynalariboshqarish elementlari.
4. Muloqotoynalarini yaratish.

## KIRISH

Foydalanuvchilar bilantizimning muloqotini interaktiv amalga oshirish uchun muloqotoynalar kerak. Muloqotoynalari 3 takatlaguruhlarga bo'linadi. Tizimli, ya'ni OT bilan ishlashga o'ljallangan, interaktiv xabarlar ni berish va aniqlavoblarni olish uchun mo'ljallangan, dasturchining yoki foydalanuvchining tashabbusi bilan yaratiladigan muloqotoynalar ibor.

**Visual C++ muhitidamuloqotoynalari.** Yuqorida aytib o'tilgandek, muloqotoynalar yaratish ularni maqsadlaridankelib chiqqan holda amalga oshiriladi. Bu oynalarning o'zining talablari mavjudki, shu talablar bajarilsa, umuloqotoyna sibo'laoladi. Butalablarga quyidagilarkiradi:

1. Muloqotoynasining sarlavhasi bo'lishi va unda faqat oynani yopishtug masining bo'lishi lozim,

tizimli menyu va boshqatugmalar bo'lishi mumkin emas.

Istesnotari qasidaba'zihollarda, yordam tugmasini joylashtirish mumkin.

2. Muloqotoyna siteskari aloqaga o'ljallanganligi uchun, unishartini bajar masdantizimning boshqa oyna sigao'tish mumkin emas.

3. Teskari aloqaning bir netaurlarini amalga oshiruvchi tugmalar bo'lishi kerak.

4. Muloqotoynalar asosiy oynadan har doim kichik bo'lishi shart.

5. Muloqotoynaning asosiy maqsad aniqlanishi kerak, yoki turi, nima munosabat uchun muloqotoyna sichiqligini.

6. Muloqotoynasidan boshqa muloqotoyna sigao'tish mumkin emas, asosiy oynaga o'tish lozim.

7. Muloqotoyna murojaat qilingan dastur tilishi vateskari aloqa qabul qililgan dastur so'ng xotiradan o'chirib tashlanishi lozim.

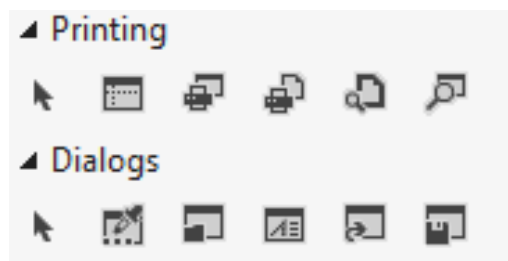
8. Muloqotoyna gadinamik xotiralar bo'lishi mumkin emas.

Ushbu talablarni bajarish har qanday oyna muloqotoyna sihisoblanadi.

Visual

C++ da OT bilan muloqot qilishga o'ljallangan muloqotoynalariga

[Dialogs] tab dagiva [Printing] tab dagibarcha komponentalarkiradi (11.1-rasmga qarang).



11.1-rasm. Tizimlimuloqotoynalariyaratishkomponentalari.

Visual

C++ninghujjatlarigaqarasangiztizimlimuloqotoynalariuchunyagonamuloqotoynalariro'yxatituzilgan.

Buro'xatgaquyidagijadvalgakeltirilgankomponentalarkiradi.

11.1-jadval.Visual

C++daOTbilanmuloqotqilishgamo'ljallanganmuloqotoynalariyaratishgamo'ljallangankomponentalarningvazifalari

№	Komponent nomi	Vazifasi
1	ColorDialog	Foydalanuvchilarinterfeyselementiranginio'rnatishimkoniniberuvchiranglarpalitrasiuchunmuloqotoynasiniko'rsatadi.
2	FontDialog	Foydalanuvchilargakeraklikkomponentauchunshriftvauningxususiyatlarinio'rnatishimkoniniberuvchimuloqotoynasiniko'rsatadi.
3	OpenFileDialog	Foydalanuvchilaruchunfaylnitanlashimkoniniberadiganmuloqotoynasiniko'rsatadi.
4	PrintDialog	Foydalanuvchilargaprinternitanlashvauningxususiyatlarinio'rnatishimkoniniberuvchimuloqotoynasiniko'rsatadi.
5	PrintPreviewDialog	FoydalanuvchilaruchunchopqilishdaPrintDocumentboshqaruv elementiningko'rinishiningko'rsatishimkoniniberadiganmuloqotoynasiniko'rsatadi.
6	FolderBrowserDialog	Foydalanuvchilaruchunpapkalarko'rish, yaratishvatanlashimkoniniberadiganmuloqotoynasiniko'rsatadi.
7	SaveFileDialog	Foydalanuvchilaruchunfaylnisaqlashimkoniniberadiganmuloqotoynasiniko'rsatadi.

Bukomponentalarmaxsusxususiyatgaasoslagankomponentalarbilashishlatiladi.

Interaktivxabarlarniberishvaaniqjavoblarniolish Visual C++ da MessageBoxsinfimavjud.

Businfbilanbarchaixtiyoriyturdagimuloqotoynalariyaratishmumkin.  
 Sinfningnomlarfazosi System.Windows.Forms  
 bo‘libhisoblanadivakutubxonasi System.Windows.Forms.dll  
 hisoblanadi. Bumuloqotoynasiformasinfningmerosxo‘rihisoblanadi.  
 Unda 21 taturlikombinatsiyalishow funksiyachibor.  
 Uningquyidagiparametrlaribor.

11.2-jadval. MessageBox sinfnining show funksiyasiparamertlari

Paramertno mi	Tipi	vazifasi
text	String	Muloqotoynasiningxaba ri
caption	String	Muloqotoynasining sarla vhasi
buttons	MessageBoxButtons	Teskari aloqanita‘minala shtugmalariturlarinianiqlash
icon	MessageBoxIcon	Muloqotoynalariningiko nkalariturlarinianiqlash
defaultButton	MessageBoxDefaultButton	Teskari aloqanita‘minala shuchun joriy tugmalariturlarinianiqlash
options	MessageBoxOptions	Muloqotoynalariningam allariturlarinianiqlash
helpFilePath	HelpNavigator	HelpNavigator ob’ekting qiymatlari uchun foydalaniladi
param	Object	Yordam tugmasibosilgan da ID fiymatnianiqlashimkoni niberuvchi parametr
Returns	DialogResult	Muloqotoynalarining qiy matlari qaytarish uchun foydalaniladigan DialogResult tipidagi turlarinianiqlash
Exceptions	InvalidEnumArgumentExcep tion	MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton

		ob'ektlarini biriniqabul qiluvchi kengaytirilgan ob'ekt tipi
owner	IWin32Window	muloqot oynasini egasi Iwin32window amalga oshirish.

Bu parametrlarning o'ziga mos qiymatlari oldindan aniqlab berilgan bo'lib, oddiy soddako'rinish dastur muloqot oynasini yaratishimkon beradi. Keyinroq busin parametrlari qiymatlarivaularga ishlov berish niko'ribchiqamiz.

Foydalanuvchi tomonidan yaratiladigan muloqot oynasi forma kabi yaratiladi va ularni loyihalash dastur chining yoki foydalanuvchining xoxishiga qarab amalga oshiriladi.

Interaktiv muloqot oynalarikabi interaktiv tugmalarni yaratish va ularni boqarish, kerakli ma'lumotlarni olish uchun ishlatiladi. Shuni ham inbo'atga olish kerak yaratiladigan muloqot oynasitalablarga mos kelish kerak.

Bu talablarni amalga oshirish uchun forma ning xususiyatlariga ishlov berish, lohim bo'lsa, asosiy oyndakerakli xususiyatlarni o'rnatishimkon. Bunda ham forma ning show usuli mavjud bo'lib. Shu orqali forma chaqiriladi.

Sinfning nomlar fazosi System.Windows bo'lib hisoblanadi va kutubxonasi PresentationFramework.dll hisoblanadi. Bu muloqot oynasi forma sinfining meros xo'ri hisoblanadi.

.NET 5 Preview 1, .NET Core 3.1, 3.0 va .NET Framework 4.8 4.7.2 4.7.1 4.7 4.6.2 4.6.1 4.6 4.5.2 4.5.1 4.5 4.0 3.5 3.0 variantlarida qo'llab quvvatlanadi.

Foydalanuvchi muloqot oynalarini yaratishni keyinroq batavsil ko'ribchiqamiz.

**Muloqot oynalarini sozlash.** Bu muloqot oynalarini sozlash uchun tizim muloqot oynalaridan foydalanish va ularga ishlov berish nazarda tutilgan. Yuqoridagi keltirilgan 7 ta muloqot oynalaridan foydalanishlar, xususiyatlarini va hodisalarini boshqarish to'g'risida to'xtalamiz.

**1. ColorDialog muloqot oynasi.** Bu oyna foydalanuvchilar interfeys elementirangini o'rnatishimkonini beruvchi ranglar palitrasini uchun muloqot oynasini ko'rsatadi.

Bu komponentdan forma gao'rnatilganda hech qanday ko'rinish hosil bo'lmaydi, ammo forma ning ichki tuzilmasiga qo'yshiladi. Forma ning ishchi holatidagi forma sinining pastki qismida uning ob'ekti yaratil

adi. `solorDialog1` ob'ektini yaratish orqali boshqariladi. Uning xususiyatlarini va hodisalarini ham mavjud va loyiha oynasida foydalanuvchi xususiyatlarini o'z ichiga oladi.

Uyerdan kerakli xitoyriyani nashu ob'ektga mos xususiyat va hodisalarini o'rnatish mumkin.

`ColorDialog` muloqot oynasidan foydalanish uchun `ColorDialog()` konstruktorini ishlatish kerak.

Uning xususiyatlarini, usullarini va hodisalarini mavjud.

11.3-jadval. `ColorDialog` muloqot oynasining xususiyatlari

<code>AllowFullOpen</code>	Maxsus ranglarni qilish uchun muloqot oynasini foydalanish mumkin yoki yo'qligini o'rnatish
<code>AnyColor</code>	Muloqot oynasida asosiy ranglarni majmuining barcha mavjud ranglarni o'rnatish yoki yo'qligini o'rnatish
<code>CanRaiseEvents</code>	Komponentning hodisalarini o'rnatish mumkinligini aniqlash.
<code>Color</code>	foydalanuvchi tomonidan tanlangan rangni o'rnatish.
<code>Container</code>	Komponentni o'z ichiga olgan <code>IContainer</code> ni o'rnatish
<code>CustomColors</code>	Muloqot oynasida ko'rsatilgan maxsus ranglarni o'z ichiga oladi yoki yo'qligini o'rnatadi.
<code>DesignMode</code>	Komponentning dizayn rejimida ekanligini ko'rsatadigan qiymatni o'ladi.
<code>Events</code>	komponentning o'z ichiga olinadigan hodisalarini o'z ichiga oladi.
<code>FullOpen</code>	Muloqot oynasini ochilganda maxsus ranglarni yaratish uchun ishlatiladigan boshqaruv elementlarini ko'rsatish yoki yo'qligini o'rnatadi.
<code>Options</code>	<code>ColorDialog</code> boshlash uchun xususiyatlarini o'ladi.
<code>ShowHelp</code>	Yordam tugmasini rang muloqot oynasida ko'rsatish yoki yo'qligini o'rnatish
<code>Site</code>	Komponentning <code>ISite</code> ni o'ladi yoki yo'qligini o'rnatadi.
<code>SolidColorOnly</code>	Muloqot oynasini qattiq ranglarni tanlash uchun foydalanuvchilarni cheklash yoki yo'qligini o'rnatish



ly	
Tag	nazorathaqidama'lumotlarnio'zichigaolganob'ektsozlash.

#### 11.4-jadval. ColorDialog muloqotoynasiningusulari

CreateObjRef(Type)	Ob'ekt bilan muloqot qilish uchun ishlatiladigan proksi ishlab chiqarish uchun zarur bo'lgan barcha tegishli ma'lumotlarnio'zichigaolganob'ektyaratadi.
Dispose()	Komponent tomonidan ishlatiladigan barcha resurslarni chiqaradi.
Dispose(Boolea)	Tarkibiy qisim tomonidan ishlatiladigan boshqarilmaydigan resurslarni chiqaradi va ixtiyoriy ravishda boshqariladigan resurslarni chiqaradi.
Equals(Object)	Belgilangan ob'ekt joriy ob'ektga teng yoki yo'qligini aniqlaydi.
GetHashCode()	Standart hash funksiyasi sifatida xizmat qiladi.
GetService(Type)	Komponent yoki uning konteyneri tomonidan taqdim etilgan xizmatni ifodalovchi ob'ekt ni qaytaradi.
GetType()	Joriy ob'ekt turi ni oladi.
HookProc(IntPtr, Int32, IntPtr, IntPtr)	Umumiy muloqotoynasiga xos funksiyalarni kiritish uchun belgilangan umumiy muloqotoynasiga protsedura ni belgilaydi.
MemberwiseClone()	Joriy ob'ektning oddiy nusxasini yaratadi.
MemberwiseClone(Boolea)	Joriy Marshal byref object ob'ektning oddiy nusxasini yaratadi.
OwnerWndProc(IntPtr, Int32, IntPtr, IntPtr)	Umumiy muloqotoynasiga maxsus funksiyalarni qo'shish uchun belgilanadigan oynatartibini belgilaydi.
Reset()	Ularning standart xususiyatlariga barcha imkoniyatlarini ishga soladi, oxirgi tanlangan qorarang vaularning standart xususiyatlariga maxsus ranglar moslashtiriladi

RunDialog(I ntPtr)	Umumiymuloqotoynasini belgilaydi.
ShowDialog ( )	Umumiymuloqotoynasini ishlatadi.
ShowDialog (IWin32Wi ndow)	Umumiymuloqotoynasini ishlatadi.
ToString()	Muloqotoyna qiymatini satrga o'tkazish.

11.5-jadval. ColorDialog muloqotoynasining hodisalari

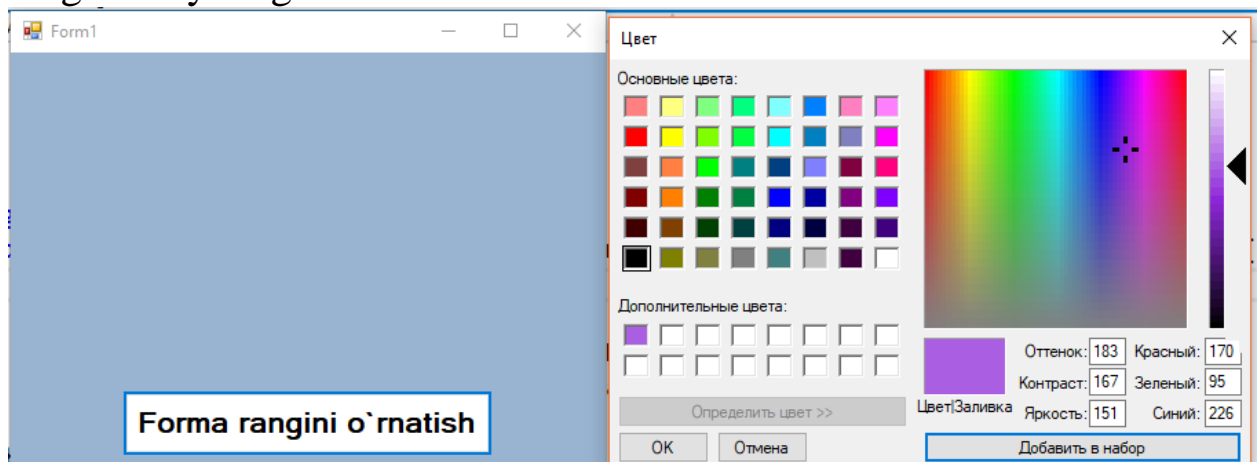
Disposed	Komponentni Dispose() metodigacha qirib qo'yilgan va yuzaga keladi
HelpReq uest	Foydalanuvchi umumiy muloqotoyna sida tugma sinib o'tganligini bildiradi.

ColorDialog muloqotoynasidan foydalanish uchun forma gabitugma o'rnatilgan va uning Click hodisasi yordamida chaqiramiz. Tanlangan rang safora maning va tugma ning fonini o'zgartirsin.

Tugmaning Click hodisasi quyidagicha dastur fragmentini o'rnatamiz.

```
if(colorDialog1->ShowDialog() ==
::System::Windows::Forms::DialogResult::OK)
    Form1::BackColor = colorDialog1->Color;
    button1->BackColor = colorDialog1->Color;
```

Dastur fragmentida muloqotoynachicha qirilgan davataskari aloqasi ob'ektini qaytarsa forma va tugma ning mos xususiyatlarini o'zgartirish algoritmi yozilgan.



11.2-rasm. ColorDialog muloqotoynasidan foydalanish.

ColorDialog muloqotoynasining xususiyatlari, usullari va hodisalari nimasalarini masalaning ahamiyatiga qarab ishlatish mumkin.

## 2.FontDialog

## muloqotoinasi.Buoyna

foydalanuvchilargakeraklikkomponentauchunshriftvauningxususiyatlarini o'rnatishimkoniniberuvchimuloqotoinasiniko'rsatadi.

Bukomponentdaniformagao'rnatilgandahechqandayko'rinishhosilbo'lmaydi, ammoformaningichkituzilmasigaqo'yshiladi. Fomraningishchiholatidagiformasiningpastkiqismidauningob'ektiyaratiladi.

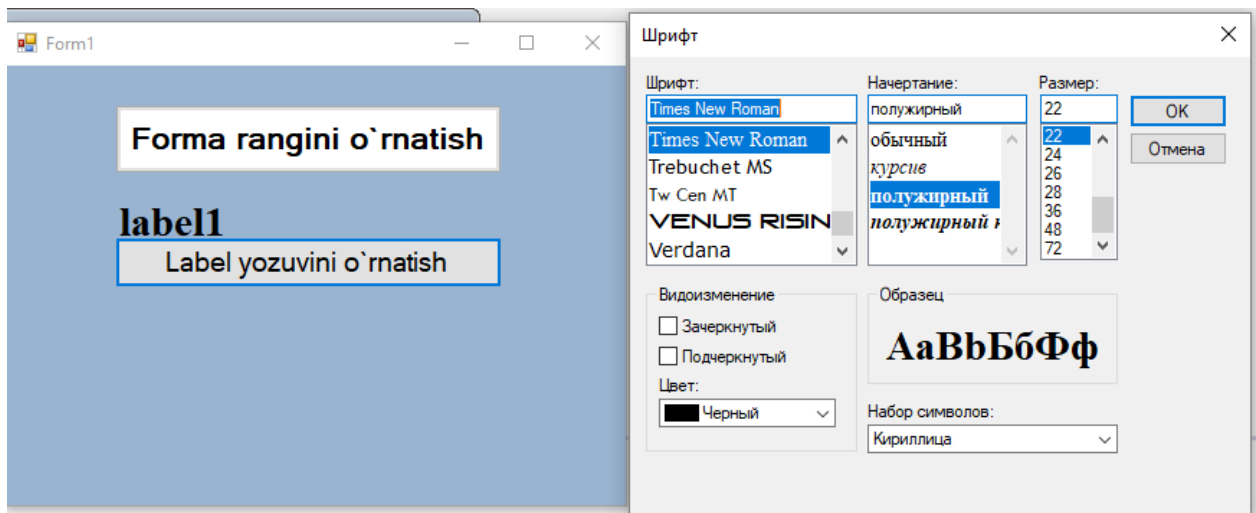
Bumuloqotoinasiniishlatishuchunformagabir Label va button ob'ektlarinio'rnatamiz. Tugmabosilganda Label ob'ektningmatniniyozuvxususiyatlarinio'rnatishniko'ribchiqamiz.

Buninguchuntunmaningbosilgandaxususiyatigaquyidagidasturfragmentni iyozi shiyetarli.

```
fontDialog1->ShowColor = true;
fontDialog1->Font = label1->Font;
fontDialog1->Color = label1 ->ForeColor;
Color color = label1->ForeColor;
System::Drawing::Font^ font = label1->Font;
System::Windows::Forms::DialogResult result =
fontDialog1->ShowDialog();
if(result ==
::System::Windows::Forms::DialogResult::OK) {
    label1->Font = fontDialog1->Font;
    label1->ForeColor = fontDialog1->Color;
}
```

Dastudadastlab Label ob'ektningrangvayozuvlarinisqaqlabolinadi, chunkiyozuvni formatlashmuloqotoinasichaqirilgandajoriyholatniolishuchun.

Dasturdarangvayozuvqiymatlarinisqaqlashuchuno'zgaruvchilarnianiqlash olibhamkeltirilgan. Muloqotoinasining [OK] hodisasibajarilgandayozuvvarangnio'zgartirishko'rsatilgan.



11.3-rasm. FontDialog muloqotoynasidanfoydalanish.

Businfningxususiyailari, usullarivahodisalarimavjud. Ularamaliyvazifalarnibajarishdafoydalanishmumkinvamustaqilo`rganish lozim.

Chunkibirvazifaniamalgaoshirishuchunturlixilalgoritmlardanfoydalanish mumkin.

### 3.OpenFileDialog

### muloqotoynasi.Buoyna

foydalanuvchilaruchunfaylnitanlashimkoniniberadiganmuloqotoynasinik o`rsatadi.

Bumuloqotoynasihamyuqoridagidekfoydalaniladi.

Shuningdek, sinfningshosxususiyatlari, usullarivahodisalarimavjud. Farqlixususiyatlarnikuo`rsatishuchunbirmisololamiz.

Undatugmabosilganda

label

matnigafaylningto`liqyo`liniolishvao`rnatishuchunquyidagidasturfragme ntiniyoziladi.

```

openFileDialog1->InitialDirectory = "c:\\";
openFileDialog1->Filter = "txt files (*.txt)|*.txt|doc files (*.doc)|*.doc,*.docx|All files (*.*)|*.*";
openFileDialog1->FilterIndex = 2;
openFileDialog1->RestoreDirectory = true;
openFileDialog1->Title = "Fayl nomini olish";
if ( openFileDialog1->ShowDialog() == System::Windows::Forms::DialogResult::OK ){
    if ( openFileDialog1->OpenFile() != nullptr ){
        label1->Text = openFileDialog1->FileName;
    }
}

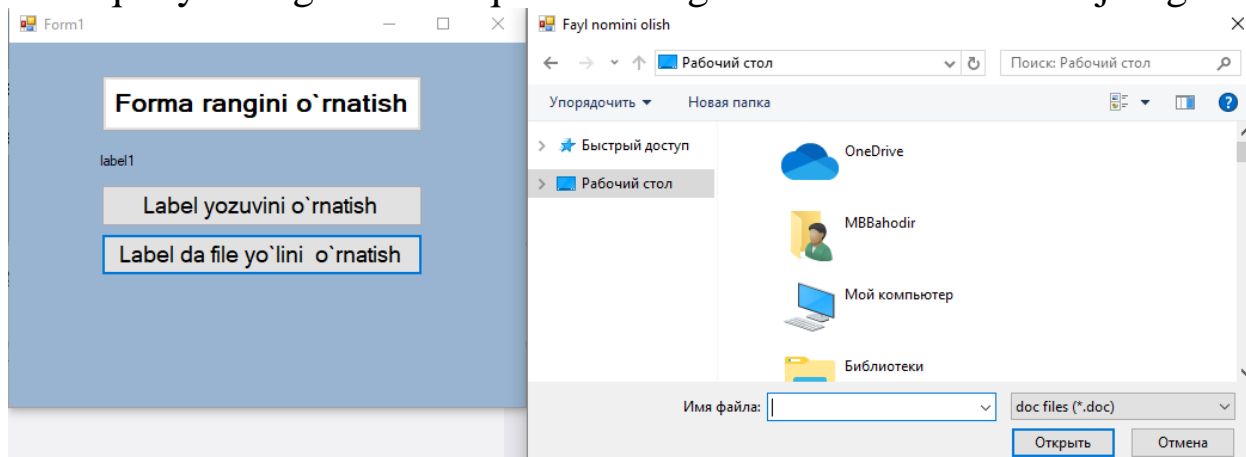
```

Dasturning birinchi satri joriy katalog o'ralmas. Shuningdek, joriy foydalanuvchining kerakli kataloglarini ham o'ralmas qilish zarur. So'ng, fayllarni tip bo'yicha filtrlash o'ralmas. Hozirda 3 ta format o'ralmas.

Keyingi satri da fayl kengaytmalaridan qaysi biri joriy bo'lib chiqishini belgilash zarur.

Ko'rsatilgan katalogni faollashtirish bajarilgan va muloqot oynasining sarlavhasi dastur kerakli matn joylashtirilgan.

Muloqot oynasining teskari aloqasi tekshirilgan va shu asosida amal bajarilgan.



11.4-rasm. FontDialog muloqot oynasidan foydalanish.

Odatda bu muloqot oynasiga lumtizi matn shafayllarni matnini joylashtirish uchun ishlatiladi.

Buni qanday amalga oshirish mumkin.

Buning uchun kichik bo'lsah matn muharriri yaratish lozim va unga OT dan oddiy manli fayllarni yuklab olish mumkin.

Fayl ichida gima lumotlarni olish uchun quyidagicha dastur fragmenti yozish mumkin:

```

openFileDialog1->Filter = "text files
(*.txt)|*.txt|cpp files (*.cpp)|*.cpp|All files
(*.*)|*.*";
openFileDialog1->FilterIndex = 2;
openFileDialog1->RestoreDirectory = true;
openFileDialog1->Title = "Fayl nomini olish";
openFileDialog1->FileName = "";
if ( openFileDialog1->ShowDialog() ==
System::Windows::Forms::DialogResult::OK){
    if ( (myStream = openFileDialog1->OpenFile())
!= nullptr ){
        System::IO::StreamReader ^ sr = gcnew

```

```

System::IO::StreamReader(openFileDialog1-
>FileName);
    richTextBox1->Text = sr->ReadToEnd();
sr->Close();
    }
}

```

Bu amaliy dastur fragmentini amaliyot gasinabko'rish orqali tahlil qilindi. 4. PrintDialog mulqotoinasi. Bu oyna foydalanuvchilarga printerni tanlash va uning xususiyatlarini o'rnatishimko'rinib beruvchi mulqotoinasini ko'rsatadi. Oynaning judako'pxususiyatlari, usullari va hodisalaribor.

Bularni mustaqil ishlarini bajarish vaqtida amalga oshirish mumkin. Shuningdek, bu mulqotoinani ishlatish usullari ham judako'p. Odatda dastur chining loyihaholatidankelib chiqqan holda foydalaniladi.

```

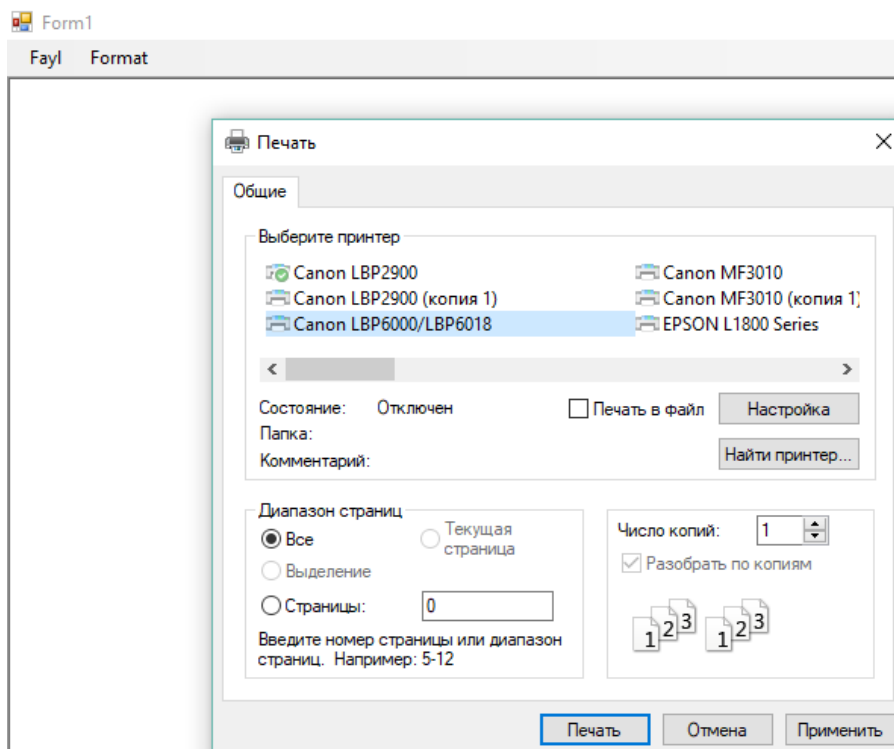
printDialog1->AllowSomePages = true;
    printDialog1->ShowHelp = true;
    if ( printDialog1 == nullptr )
        System::Windows::Forms::MessageBox::Show("pnull"
);
        System::Windows::Forms::DialogResult result =
printDialog1->ShowDialog();

System::Windows::Forms::MessageBox::Show(result.ToString());
    if ( result ==
System::Windows::Forms::DialogResult::OK )
    {
        // docToPrint->Print();
        System::Windows::Forms::MessageBox::Show("Chop
qilish boshlanadi");
    }
}

```

Dastur fragmentidama'lumotlarni chop qilish uchun avval unima'lumbirchi quvchi oqimga yozish va oqimni esa, hujjat formatiga joylashtirish kerak. Hujjatni esa, sahifalarni sozlab docToPrint ob'ekti yaratilish kerak. Ammo chop qilishning turliparametrlarini ishlatish uchun mulqotoinasini chiqarib beradi.

Kerakli xususiyatlarni o'rnatganda so'ngamalni bajarish mumkin.



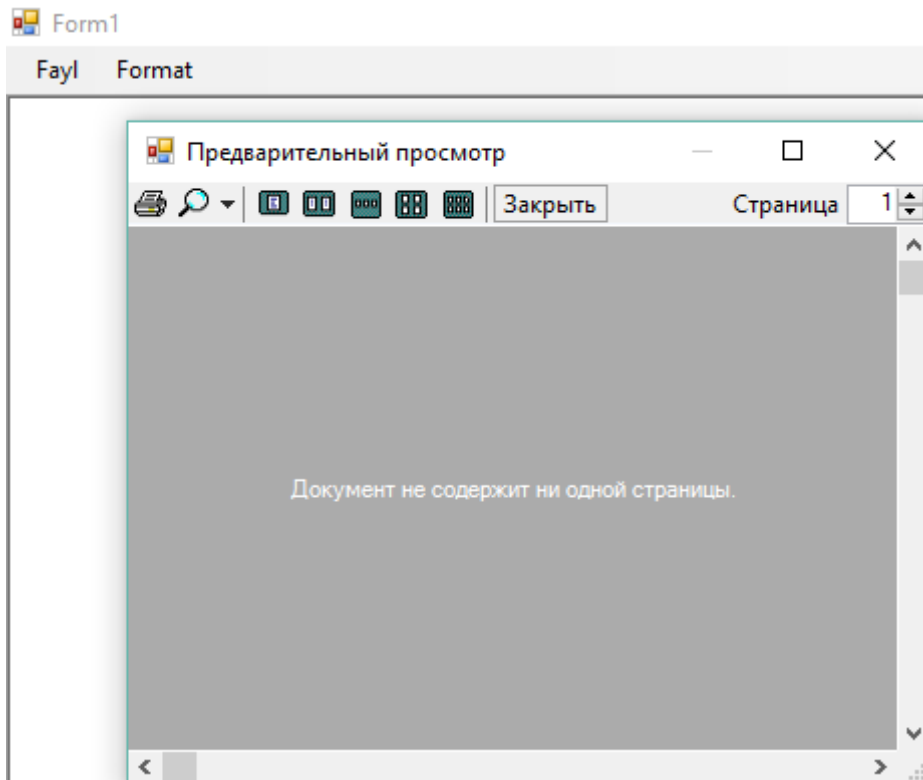
11.5-rasm. PrintDialog muloqotoynasidan foydalanish.

Dasturfragmentini bir loyihajoylashtirib, ishlatsangiz rasmda gide kishlash kerak.

**5. PrintPreviewDialog muloqotoynasi.** Bu oyna foydalanuvchilar uchun chop qilishda PrintDocument boshqaruvelementining ko'rishining ko'rsatishimkonini beradigan muloqotoynasi ko'rsatadi.

```
printPreviewDialog1->MinimumSize =
System::Drawing::Size( 375, 250 );
    printPreviewDialog1->UseAntiAlias =
true;
    printPreviewDialog1->Document =
document;
        printPreviewDialog1-
>ShowDialog();
```

Bu muloqotoynasi ham yuqoridagi muloqotoynasi kabi sozlashlarni bajarishgandansa foydalanish mumkin. Dasturfragmentida qarasangiz document ob'ektini yaratish lozim. Chop qilishni g'ob'ektidan farqqilgan xolda oqimdagima'lumotni formatla shvaunichiqishini A4 shakl gake ltrish orqali amalga oshiriladi.



11.6-rasm. PrintPreviewDialog muloqotoynasidanfoydalanish.

**6.FolderBrowserDialog muloqotoynasi.**Buoyna foydalanuvchilaruchunpapkalarko‘rish, yaratishvatanlashimkoniniberadiganmuloqotoynasiniko‘rsatadi. Bundanma‘lumotlarniko‘chirishda, papkalaranitaqqsolashdaishlatishmumkin. Muloqotoynasiningxususiyatlari, usullarivahodisalarmavjud. Ularningba‘zilariniquyidagidasturfragmentigakeltiribo‘tamiz.

```
System::IO::Stream ^ myStream;
System::Windows::Forms::DialogResult result =
folderBrowserDialog1->ShowDialog();
if ( result ==
System::Windows::Forms::DialogResult::OK ) {
    System::String^ folderName =
folderBrowserDialog1->SelectedPath;
    openFileDialog1->InitialDirectory = folderName;
    openFileDialog1->FileName =
String::Concat(folderName, "\\1.cpp");
    if ( (myStream = openFileDialog1->OpenFile())
!= nullptr ){
        System::IO::StreamReader ^ sr = gcnew
System::IO::StreamReader(openFileDialog1-
>FileName);
```



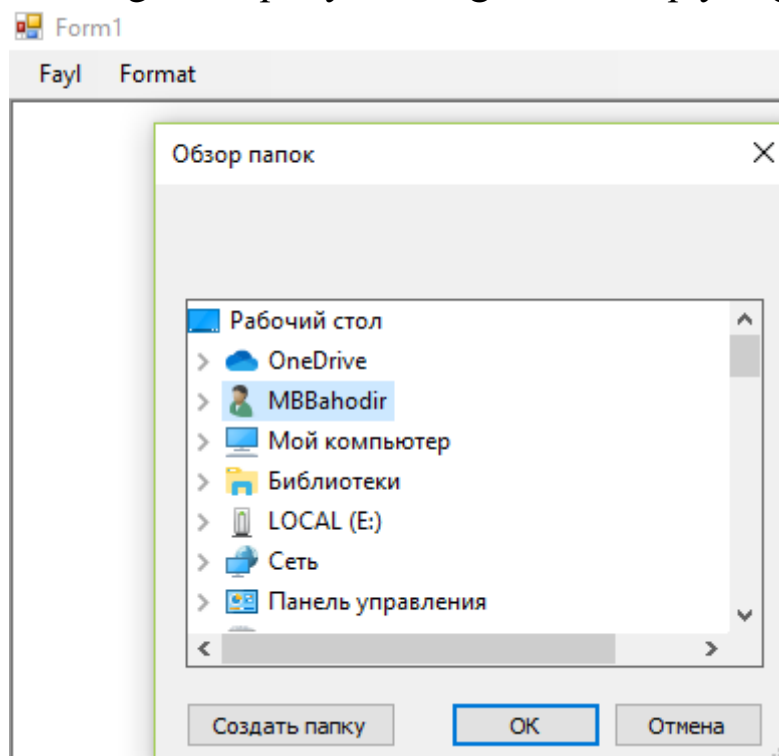
```

    richTextBox1->Text = sr->ReadToEnd();
    sr->Close();
}
}

```

Dasturfragmentidapapkauchunmuloqotoynasichaqirilganvako‘rsatilganpapkadan 1.cpp fayliniyuklabkelgan. Bundayholatfaylmenejerlaruchunpapkadagifayllarningro‘yhatinihamolis hmumkin.

Dasturfragmentidabir StreamReader oqimyaratilgan. Oqimfaylningma’lumotlarinio‘qishuchunyaratilgan. Oqimningkonstruktoriaosida sr oqimob’ektiyaratiladi. Buoqimning ReadToEnd funksiyasiorqali richTextBox gama’lumotlarjoylashtiriladi. FolderBrowserDialog muloqotoynasiningko‘rinishiquyidagicha:



11.7-rasm. FolderBrowserDialog muloqotoynasidanfoydalanish

### 7. SaveFileDialog muloqotoynasi. Buoyna

foydalanuvchilaruchunfaylnisaqlashimkoniniberadiganmuloqotoynasini ko‘rsatadi.

Buhamasosanmatnlivamaxsustuzilmalarima’lumotlarinisaqlashuchunish latiladi. Ma’lumotqanaqatuzilmadayozilsa, shundaytuzilmadao‘qiladi.

```

System::IO::Stream^ myStream;
    MemoryStream^ userInput = gcnew
MemoryStream();

```

```

        richTextBox1->SaveFile( userInput,
RichTextBoxStreamType::PlainText );
        userInput->WriteByte( 32 );

        // saveFileDialog1->CreatePrompt = true;
        // saveFileDialog1->OverwritePrompt = true;
        // saveFileDialog1->FileName = "myText";
        // saveFileDialog1->DefaultExt = "txt";

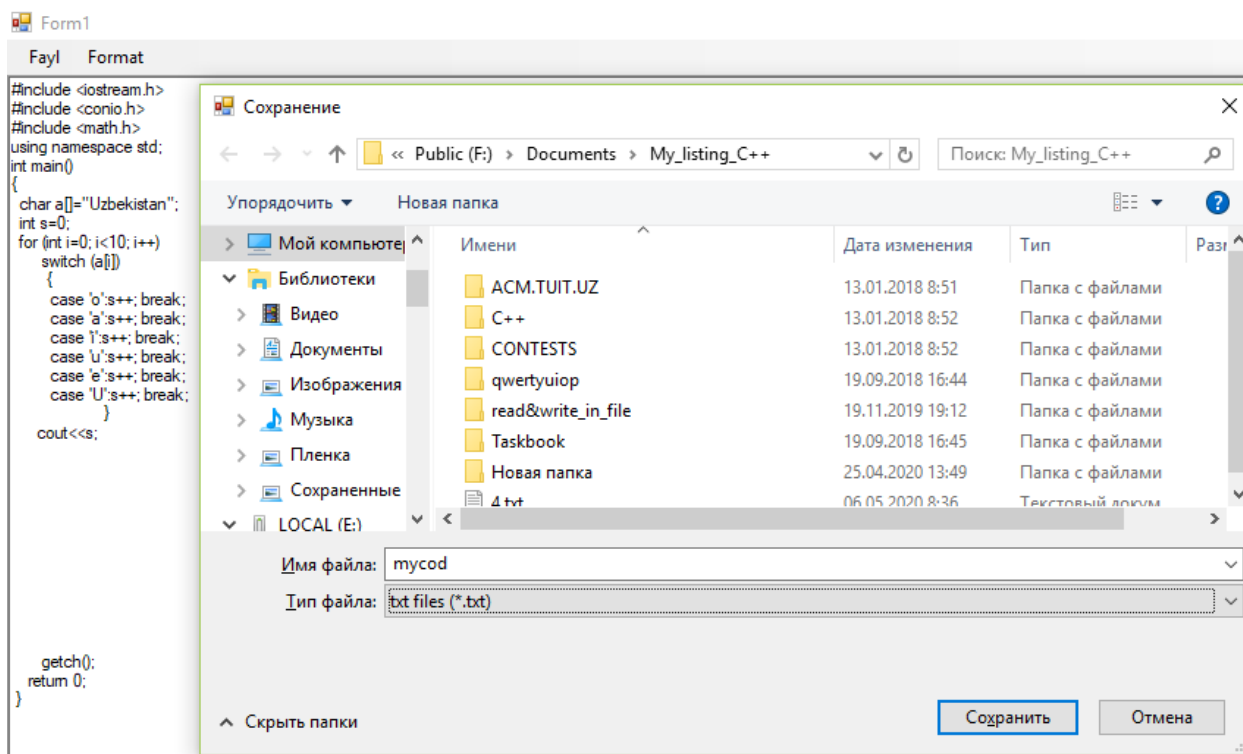
        saveFileDialog1->Filter = "txt files
(*.txt)|*.txt";
        saveFileDialog1->FilterIndex = 2;
        saveFileDialog1->RestoreDirectory = true;
if ( saveFileDialog1->ShowDialog() ==
System::Windows::Forms::DialogResult::OK ){
        myStream = saveFileDialog1->OpenFile();
        userInput->Position = 0;
        userInput->WriteTo( myStream );
        myStream->Close();
}

```

Dasturfragmentida ikki xil oqim yaratilgan. myStream - birinchi sifatda anaviy oqim va MemoryStream - xotira oqim yaratilgan. Birinchi oqimni olish va ikkinchi sifatda oqimni xotiraga yozish, berilgan oqim bilan oqimlashgan qaratilgan.

Position buni pozitsiya anib oshlash, ammo olib tashlash ham ishlaydi.

Bu mulotoynasina faqat matnli ma'lumotlarni balki, foydalanuvchining xoxlagan ma'lumotni matn yoki binarkodlash orqali saqlash ham mumkin. Mulotoynasining ko'rinishi quyidagidek g'irasmda keltirilgan.



11.8-rasm. **SaveFileDialog** muloqotoy nasidan foydalanish

Ko'rib chiqilgan muloqotoy nalaribarchasi OT bilan ishlashgamo'ljalla nganligiko'rinib, turibdi. OT qanday sozlangan bo'lsa, bu muloqotoy nalar ham shu rejimda ishlaydi.

Oynalarga deyarli o'zgartirish kiritish shart emas, xuddi kimbarchasi kutilganideki yaratilgan gao'xshaydi. Faqat eng kattamu ammosi lokalizatsiya qilish.

### **Muloqotoy nalariboshqarish elementlari.**

Bunda interaktiv muloqot qilish oynalaritushiniladi.

Ularni yaratish va boshqarish dasturchining xoxishiga qarab amalga oshiriladi. Yuqorida aytib o'tganimizdek, Visual C++ da MessageBox sinfi haqida yuqoridagideki aytib o'tgan edik. Unda 21 turlik kombinatsiyali show funksiyachibordigini ham.

Shuningbuni funksiyalar va ularning parametrlari hamda parametrlarining qiy matlarit o'g'risima'lumotlarni va muloqotoy nalarini yaratish gavaishlov berishga e'tiborni qaratamiz.

**1.Show(String) fknutsiyasi.** Bir argumentli funksiyabo'lib, belgilangan matn bilan xabarko'rsatadigan muloqotoy nasini yaratish uchun ishlatiladi. Uning kiruchi parametri System::String tipidabo'lib, System::Windows::Forms::DialogResult tipidagi qiymat qaytaradi.

```
if( MessageBox::Show("Bu oddiy sohow") ==
System::Windows::Forms::DialogResult::OK)
    this->Close();
```

2.Show(IWin32Window, String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, String, String) funksiyasi. Buko 'pparamerlibo'lib, belgilangan paoametlarasosidainteraktivmuloqotoynasiniyaratishuchunis hlatiladi. Parametlari quyidagi qiymatlari qabalqiladi.

IWin32Window Interface - System.Windows.Forms nomlar fazosiva System.Windows.Forms.dll kutubxonasi dan foydalanib, Win32 HWND nijoriy qilish interfeysi beradi. IWin32Window sinfi interfeysidanmerosxo'roladi. System.Windows.Forms.Control va System.Windows.Forms.NativeWindow umumiy ruxsat sinflari interfeyslarini ishlatadi, qiymatlar sifatida ComVisibleAttribute, GuidAttribute, InterfaceTypeAttribute tiplarini ishlatadi.

MessageBoxButtons - Enum tipidagi parametrdir. System.Windows.Forms nomlar fazosiva System.Windows.Forms.dll kutubxonasi dan foydalanadi.

Muloqotoynalaridako'rsatilish kerak bo'lgan tugmalarni aniqlaydi.

Qiymatlari public enum class MessageBoxButtons ta'luqlidir.

11.6-jadval. MessageBoxButtons qabul qiluvchi qiymatlar

No	Qiymat nomi	vazifasi
1	AbortRetryIgnore	Muloqotoynasida Abort, Retry Ignore tugmalarini o'rnatish
2	OK	Muloqotoynasida OK tugmasini o'rnatish
3	OKCancel	Muloqotoynasida OK va Cancel tugmalarini o'rnatish
4	RetryCancel	Muloqotoynasida Retry va Cancel tugmalarini o'rnatish
5	YesNo	Muloqotoynasida Yes va No tugmalarini o'rnatish
6	YesNoCancel	Muloqotoynasida Yes, Nova Cancel tugmalarini o'rnatish

Muloqotoynasiga MessageBoxButtons tugmalarini o'rnatish dastur fragmenti:









```

if ((MessageBox::Show("Joriy oynani yopishni xoxlaysizmi?", "Xabar", MessageBoxButtons::YesNo) == System::Windows::Forms::DialogResult::Yes)) {
    this->Close();
}

```

MessageBoxIcon – Enum tipidagi ob'ekt bo'lib, u ham yuqoridagidek, nomlar fazosi vakutubxonabilan ishlaydi. Muloqot oynasidako'rsatish uchun muloqot oynalarining turlarini belgilovchi konstantalarni aniqlaydi.

11.7-jadval. MessageBoxIcon qabul qiluvchi qiymatlar

N	Qiymat nomi	belgisi	vazifasi
1	Asterisk		Qandaydir hodisa haqida faqat xabar beruvchi muloqot oynasi uchun ikonka
2	Error		Qandaydir hodisa haqida faqat xato amal bajarilganligi haqida ma'lumot beruvchi muloqot oynasi uchun ikonka
3	Exclamation		Qandaydir hodisa haqida faqat ogohlantirish haqida ma'lumot beruvchi muloqot oynasi uchun ikonka.
4	Hand		Qandaydir hodisa haqida faqat xato amal bajarilganligi haqida ma'lumot beruvchi muloqot oynasi uchun ikonka
5	Information		Qandaydir hodisa haqida faqat xabar beruvchi muloqot oynasi uchun ikonka
6	None		Ikonkasiz muloqot oynasi uchun
7	Question		Qandaydir hodisa haqida faqat savolga javob beruvchi muloqot oynasi uchun ikonka
8	Stop		Qandaydir hodisa haqida faqat xato amal bajarilganligi haqida ma'lumot beruvchi muloqot oynasi uchun ikonka
9	Warning		Qandaydir hodisa haqida faqat ogohlantirish haqida ma'lumot beruvchi muloqot oynasi uchun ikonka.

MessageBoxIcon ni ishlatish uchun joriy formani yopish uchun muloqot oynani ishlatish uchun dastur fragmentini keltiramiz.

Buning uchun formaning FormClosing hodisasiga quyidagi dastur fragmenti ni yozamiz:

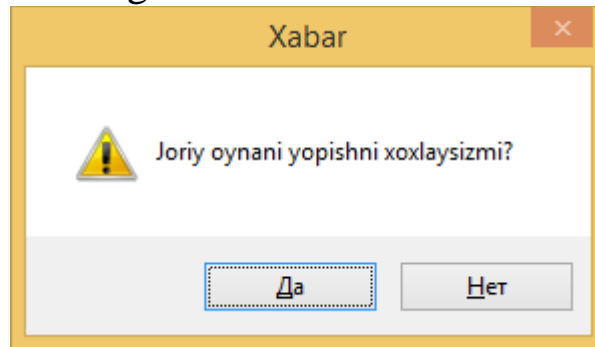
```
System::Void Form1_FormClosing(System::Object^ sender, System::Windows::Forms::FormClosingEventArgs^ e) {
    if ((MessageBox::Show(
        "Joriy oynani yopishni xoxlaysizmi?",
        "Xabar",
        MessageBoxButtons::YesNo,
        MessageBoxIcon::Exclamation) ==
```

```

System::Windows::Forms::DialogResult::No)) {
    e->Cancel = true;
}
}

```

Dasturfragmentidahodisaningto‘liqyozilishikeltirilgaganliginingsaba biundahodisagaishlrvberilgan.



11.9-rasm. MessageBoxIcon bilanmuloqotoynasiniyaratish.

Mazkuroynadayo‘qtugmasibosilsa, formayopilmaydivaaksinchatugmasibosilsa, formayopiladi.

MessageBoxDefaultButton - Enum tipidagiob‘ektbo‘lib, uhamyuqoridagidek, nomlarfazosivakutubxonabilanishlaydi. Muloqotoynasidastandarttugmalarnio‘rnatuvchio‘zgarnaslarnibelgilaydi

11.7-jadval. MessageBoxDefaultButton qabulqiluvchiqiyimatlar

No	Qiymat nomi	vazifasi
1	Button1	Muloqotoynasidastarndartbirinchitugmanijoriytugmasifatidao‘rnatish
2	Button2	Muloqotoynasidastarndartikkinchitugmanijoriytugmasifati dao‘rnatish
3	Button3	Muloqotoynasidastarndartuchinchitugmanijoriytugmasifati dao‘rnatish

Tugmabosilgandaformaniyopishniso‘rashmuloqotoynasiniyaratish dasturfragmentinikeltiramiz.

```

String^ message = "Joriy oynani yopishni
xoxlaysizmi?";
String^ caption = "Xabar";
MessageBoxButtons buttons = MessageBoxButtons::YesNo;
System::Windows::Forms::DialogResult result;
System::Windows::Forms::DialogResult mayli =
System::Windows::Forms::DialogResult::Yes;

```

```

result = MessageBox::Show( this, message, caption,
buttons, MessageBoxIcon::Question,
MessageBoxDefaultButton::Button2);
        if ( result == mayli )
        {
            this->Close();
        }

```

Odatdaturgmalarning farqini aniqlash ancha murakkab, agar OT maskabo'lsa, farqlash mumkin.

MessageBoxOptions - Enum tipidagi ob'ekt bo'lib, u ham yuqoridagidek, nomlar fazosida kutubxon bilan ishlaydi. Muloqot oynasidagi standart xususiyatlarni o'rnatish uchun foydalaniladi.

11.8-Jadval . MessageBoxOptions

N	Qiymat nomi	vazifasi
1	DefaultDialogOptions	Oddiy muloqot oynalariga o'xshaydigan faollikni o'z ichiga olmaydigan holatda o'zlashtiriladi.
2	RightAlign	Muloqot oynasining ma'lumotini o'ngga joylashtirish
3	RtlReading	Muloqot oynasidagi ma'lumotni o'ngdan chapga qarab o'qish uchun moslab joylashtirish.
4	ServiceNotification	Oddiy muloqot oynalariga o'xshaydigan faollikni o'z ichiga olmaydigan holatda o'zlashtiriladi va javob bermasdan o'tib ketish mumkin xususiyati mavjud

Tugma bosilganda forma ni yopishni so'rash muloqot oynasini yaratish dastur fragmentini keltiramiz.

```

String^ message = "Joriy oynani yopishni xoxlaysizmi?";
        String^ caption = "Xabar";
        MessageBoxButtons buttons =
MessageBoxButtons::YesNo;
        System::Windows::Forms::DialogResult
result;

System::Windows::Forms::DialogResult mayli =
System::Windows::Forms::DialogResult::Yes;

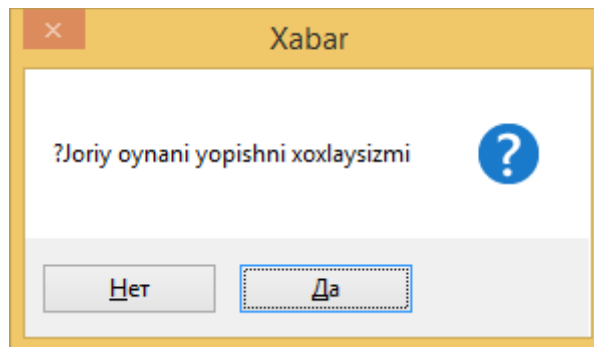
```

```

        result = MessageBox::Show( this,
message, caption, buttons,
MessageBoxIcon::Question,MessageBoxDefaultButton::But
ton1,MessageBoxOptions::RtlReading);
        if ( result == mayli )
        {
            this->Close();
        }

```

Dasturfragmentini ishlatib ko'rib, natijasini tahlil qilish va nimalarni o'ng danchapga qarabo'qisho'z garga nini ko'rish mumkin.



11.10-rasm. MessageBoxOptions bilan muloqot oynasini yaratish.

Barcha muloqot oynalari kabi yuqoridagi keltirilgan dastur fragmentlarid anko'rinib turibdiki, DialogResult tipini qaytaradi.

3. Show(IWin32Window, String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, String, String) funksiyasi. Bu funksiyaga yordam fayli va ichki indeks fayllarga ko'rsatkich yaratadi.

```

String^ message = "Joriy oynani yopishni
xoxlaysizmi?";
String^ caption = "Xabar";
MessageBoxButtons buttons = MessageBoxButtons::YesNo;
System::Windows::Forms::DialogResult result;
System::Windows::Forms::DialogResult mayli =
System::Windows::Forms::DialogResult::Yes;

result = MessageBox::Show( this, message, caption,
buttons,
MessageBoxIcon::Question, MessageBoxDefaultButton::But
ton1, MessageBoxOptions::RightAlign,
"d:\tut.chm", HelpNavigator::KeywordIndex, "iv");
        if ( result == mayli )

```

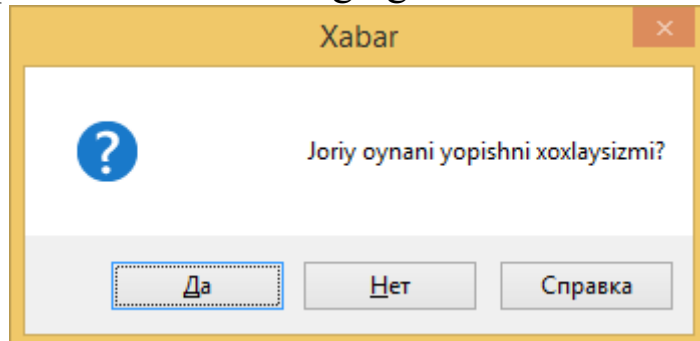


```

    {
        this->Close();
    }

```

Dasturfragmentini ishlatib ko'rib, natijasini tahlil qilish va nimalarni o'zgartirish mumkin.



11.11-rasm. Yordam tugmalini muloqot oynasini yaratish.

Dasturfragmentida `YesNo` tashlatilgan, ammo, muloqot oynasida 3 ta tugma chiqqan, demak yordam tugmasining faol holatga kelganining ko'rish mumkin.

`Show` usulining boshqa yangi parametrlarni qabul qilmaganligi uchun ularni yozilis hiva dasturfragmentlaridankeltiramiz.

`4.Show(IWin32Window, String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, String)`

```

result = MessageBox::Show( this, message, caption,
    buttons,
    MessageBoxIcon::Question, MessageBoxDefaultButton::But
    ton1, (MessageBoxOptions)0,
    "d:\tut.chm", HelpNavigator::KeywordIndex, "iv");

```

`5.Show(String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, String, HelpNavigator)`

```

System::Windows::Forms::DialogResult r3 =
    MessageBox::Show( message, caption,
    MessageBoxButtons::OK, MessageBoxIcon::Question,
    MessageBoxDefaultButton::Button1,
    (MessageBoxOptions)0, "mspaint.chm",
    HelpNavigator::Index );

```

`6.Show(String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, String, String)`

```
System::Windows::Forms::DialogResult r7 =
MessageBox::Show(message, caption,
MessageBoxButtons::OK, MessageBoxIcon::Question,
MessageBoxDefaultButton::Button1,
(MessageBoxOptions)0, "mspaint.chm",
"mspaint.chm:/paint_brush.htm" );
```

7.Show(IWin32Window, String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions)

```
result = MessageBox::Show( this, message, caption,
buttons, MessageBoxIcon::Question,
MessageBoxDefaultButton::Button1,
MessageBoxOptions::RightAlign );
```

8.Show(String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, String)

```
System::Windows::Forms::DialogResult r1 =
MessageBox::Show( " message, caption,
MessageBoxButtons::OK, MessageBoxIcon::Question,
MessageBoxDefaultButton::Button1,
(MessageBoxOptions)0, "mspaint.chm" );
```

9.Show(String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, Boolean)

```
System::Windows::Forms::DialogResult r =
MessageBox::Show(message, caption,
MessageBoxButtons::OK, MessageBoxIcon::Question,
MessageBoxDefaultButton::Button1,
(MessageBoxOptions)0, true );
```

10.Show(IWin32Window, String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, String, HelpNavigator)

```
System::Windows::Forms::DialogResult r4 =
MessageBox::Show( this, message, caption,
MessageBoxButtons::OK, MessageBoxIcon::Question,
MessageBoxDefaultButton::Button1,
(MessageBoxOptions)0, "mspaint.chm",
HelpNavigator::Index );
```

11.Show(IWin32Window, String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton)

```
result = MessageBox::Show( this, message, caption,
```

```
buttons, MessageBoxIcon::Question,  
MessageBoxDefaultButton::Button1,  
MessageBoxOptions::RightAlign );
```

```
12.Show(IWin32Window, String, String, MessageBoxButtons,  
MessageBoxIcon)
```

```
result = MessageBox::Show( this, message, caption,  
buttons, MessageBoxIcon::Question );
```

```
13.Show(String, String, MessageBoxButtons, MessageBoxIcon,  
MessageBoxDefaultButton)
```

```
result = MessageBox::Show( this, message, caption,  
buttons, MessageBoxIcon::Question,  
MessageBoxDefaultButton::Button1 );
```

```
14.Show(IWin32Window, String, String, MessageBoxButtons)
```

```
result = MessageBox::Show( this, message, caption,  
buttons);
```

```
15.Show(String, String, MessageBoxButtons, MessageBoxIcon)
```

```
result = MessageBox::Show(message, caption, buttons,  
MessageBoxIcon::Question);
```

```
16.Show(IWin32Window, String, String)
```

```
result = MessageBox::Show( this, message, caption);
```

```
17.Show(IWin32Window, String)
```

```
result = MessageBox::Show( this, message);
```

```
18.Show(String, String, MessageBoxButtons, MessageBoxIcon,  
MessageBoxDefaultButton, MessageBoxOptions)
```

```
result = MessageBox::Show( this, message, caption,  
buttons, MessageBoxIcon::Question,  
MessageBoxDefaultButton::Button1,  
MessageBoxOptions::RightAlign );
```

```
19.Show(IWin32Window, String, String, MessageBoxButtons,  
MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions,  
String, HelpNavigator, Object)
```

```
System::Windows::Forms::DialogResult r6 =  
MessageBox::Show(this, message, caption,  
MessageBoxButtons::OK, MessageBoxIcon::Question,  
MessageBoxDefaultButton::Button1,  
(MessageBoxOptions)0, "mspaint.chm",
```

```
HelpNavigator::KeywordIndex, "ovals" );
```

Buusullarni.NET5 Preview 1, .NET Core3.1 3.0va.NET Framework

4.8, 4.7.2, 4.7.1, 4.7, 4.6.2, 4.6.1, 4.6, 4.5.2, 4.5.1, 4.5, 4.0, 3.5, 3.0, 2.0, 1.1kabivarintlarda qoʻllabquvatlanadi.

**Muloqotoy nalarini yaratish.** Foydalanuvchi tomonidan muloqotoy nalarini yaratish oyna formasiga ishlov berish asosida amalga oshiriladi. Forma oyna sigamuloqotoy nasini oʻrnatish uchun quyidagi qadamlaribajaris hlozim.

1-qadam. Menyudan foydalanib, [menu] → [project] → [addClass] → [CLR] → [Windows Form] buyruqlar ketma ketligi asosida yangi forma qoʻshiladi. Yoki, [menu] → [project] → [add New item] → [UI] → [Windows Form] ham bajarsa boʻladi.

2-qadam. Yaratilgan yangi forma oyna sigi oʻtib, forma ga quyidagi xususiyatlar beriladi. [AutoSize] xususiyatning qiymatini [true] ga, [StartPosition] xususiyatining qiymatini [CenterParent] ga, [FormBorderStyle] xususiyatining qiymatiga [none], [FixedDialog], [FixedToolWindows], [SizableToolWindows] larning birini oʻrnatish mumkin.

3-qadam. Forma oyna nasini oʻzining xolagandek loyihalashingiz mumkin. Masalan, bir label, inputBox vabir button joylashtiramiz, ularni ham kerakli xususiyatlarini oʻrnatamiz.

4-qadam. Asosiy forma ga oʻtib, unga #include "MyForm.h" sarlavha faylni qoʻshamiz.

Bumuloqotoy nasi uchun yaratilgan forma oyna sibilanishlash uchun kerak.

5- Qadam. Asosiy forma ga bir label vabir button joylashtiramiz. Tugmaning klik hodisasida muloqotoy nasini chaqirish dastur fragmentini yozamiz.

```
// MyForm myForm;  
// myForm.Show();  
// myForm.ShowDialog();  
  
MyForm^ myFormWith = gcnew  
MyForm();  
// myFormWith->Show();  
myFormWith->ShowDialog();
```

Dasturfragmentida izohga olib qo'yilgan yordam ham muloqot oynalarni yaratish va chaqirish mumkin. Ammo ularni farqlar mavjud. Shuning uchun ularni hammasini shu dasturfragmentida keltirdik. Birinchi qatorda formani `MyForm myForm` oddiy ob'ekt sifatida yaratilgan. Uning birinchi `Show()` usuli hisoblanib, bu usul bilan muloqot oynasini chaqirish mantiqan xato, chunki tizim oynani yaratadivastunlikni asosiy forma ga beradi. Bu holda yaratilgan oyna birlahzagako'rinadixolos. Ikkinchi usulibu `ShowDialog()` usulyaxshiyondashuvlardan bo'lib, muloqot formasiga o'rnatilgan barcha xususiyatlargari oyaqiladi hamda foydalanish mumkin. Ikkinchi `MyForm()` konstruktordan foydalanib, `new` operatoriasosida yaratilgan, uning birinchi funksiya `Show()` hamoldingisiga o'xshash bo'lib, ammu ustunlikni foydalanuvchining o'ziga qo'yib beradi, ya'ni foydalanuvchi muloqot oyna ga javob bermas danturib, asosiy forma oynasiga o'tishim mumkin. Ikkinchisiesabu `ShowDialog()` usulyaxshiyondashuvlardan bo'lib, muloqot formasiga o'rnatilgan barcha xususiyatlargari oyaqiladi hamda foydalanish mumkin.

6-qadam.

Muloqot oynasini tugmasida quyidagicha algoritmni yozamiz. Muloqot oynasiga kiritilgan ma'lumotni asosiy forma ga olib o'tish uchun avval `public: System::String^ email;` kabibiro'zgaruvchi yaratib olamiz.

```
email = textBox1->Text;
        this->Close();
```

7-qadam.

Asosiy formaning tugmasining hodisasiga yozilgan dasturfragmentida vomi dan quyidagini qo'shib qshyamiz.

```
label1->Text = myFormWith->email;
```

8-qadam.

Loyihani ishlatib yaratilgan, yaratilgan muloqot oynani ishlashini ko'rish mumkin.

Muloqot oynani yuqoridagi darajada yaratish uchun uning ayni konstruktor yozish ham mumkin. Uni quyidagicha amalga oshiriladi.

```
public: MyForm(System::String^ title)
{
    InitializeComponent();
    _title = title;
}
public: System::String^ _title;
```

```
//...
private: System::Void MyForm_Load(System::Object^
sender, System::EventArgs^ e) {

        this->Text = _title;
    }
// Aspsiy formada esa
        MyForm^ myFormWith = gcnew
MyForm("Xabar");
        myFormWith->ShowDialog();
        label1->Text = myFormWith->email;
```

Bundayimkoniyatbilanfoydalanuvchiuchunixtiyoriymuloqotoynasi niyaratishmumkin.

Muloqotoynalaridanfoydalanishdasturfoydalanuvchilarigako‘plabq o‘layliklaryaratibberadi. Harbirdasturdabirasosiyoyonagakamidafunksionalimkoniyatigaqarab 4-5 tamuloqooyinalaribo‘ladi.

### **NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.**

1. Axborottizimlaridamuloqotoynalariningqandayguruhlaribor.
  2. Muloqotoynalarigaqo‘yiladigantalablarbormivabo‘lsa, nimauchun?
  3. Muloqotoynasidateskarialoqadegandanimanitushunasiz?
  4. Muloqotoynadanimalaraniqko‘rsatilishikerak?
  5. [Dialogs] tab
  - oynasidajoylashganmuloqotkomponentalarinisanabbering?
  6. ColorDialogkomponentasiningvazifasiniaying?
  7. Qaysikomponentafoydalanuvchilaruchunpapkalarko‘rish, yaratishvatanlashimkoniniberadiganmuloqotoynasiniko‘rsatadi.
  8. Interaktivxabarlarniberishvaaniqjavoblarniolish Visual C++
- daqandaysinfimavjud.

9. MessageBox sinfnining show funksiyasiparamertlarinisana bbering?
10. MessageBoxOptions qanday ob'ektipi va nimavazifa bajaradi.
11. MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton ob'ektlarini birini qabul qiluvchi kengaytirilgan ob'ektipini nomini ayting.
12. Foydalanuvchi tomonidan yaratiladigan muloqot oynasi qanday yaratiladi.
13. Interaktiv muloqot oynalarini kabinimalarni yaratish va ularni boqarish, kerakli ma'lumotlarni olish uchun ishlatiladi.
14. Qaysi sinfning nomlar fazosi System.Windows vakutubxonasi PresentationFramework.dll hisoblanadi.
- 15. ColorDialog muloqot oynasining asosiy vazifasi va ob'ekti qanday yaratiladi.**
16. ColorDialog muloqot oynasidan foydalanish uchun qanday konstruktorni ishlatish lozim.
17. CanRaiseEvents xususiyatining vazifasi ni ayting?
18. Komponentaga ilova qilinadigan hodisalar ro'yxatini o'ladixususiyat nomini ayting?
19. Muloqot oynasi ochilganda maxsus ranglarni yaratish uchun ishlatiladigan boshqaruvelementlariko'rinish turini ko'rsatuvchi qiymatni o'ladiga nyoki'rnatadigan xususiyat nomini ayting?
20. Dispose() bu qanday usul?
21. Komponenta yoki uning konteyneri tomonidan taqdim etilgan xizmatni ifodalovchi ob'ektni qaytaradigan hodisani ayting?
22. Disposed hodisaning vazifasi ni ayting?
23. Foydalanuvchi umumiy muloqot oynasidagi yordam tugmasini bosgan dasodir bo'ladigan usul nomini ayting?
24. Foydalanuvchilarga kerakli komponenta uchun shrift va uning xususiyatlarini o'rnatish imkonini beruvchi muloqot oynasini nomini ayting?
25. PrintPreviewDialog muloqot oynasi qaysitab dajoylashgan va vazifasi ni qaytib bering.
26. Dastur fragmentidabir StreamReader oqimini mauchun yaratiladi.
27. Win32Window Interface nima uchun ishlatiladi.
28. Muloqot oynasida OK va Cancel tugmalarini o'rnatish qayday amalga oshiriladi.
29. MessageBoxIcon muloqot oynalaridan imauchun ishlatiladi.

30. Qandaydir hodisa haqida faqat xato amal bajarilganligi haqidama'lu motberuvchi muloqot oynasi uchun konkanomivayozlashni tushuntirib bering.

31. Muloqot oynasida standart tugmalarni o'rnatuvchi o'zgarishlarni belgilaydigan parametrni ayting?

32. RtlReading qaysi parametr uchun ishlatiladi va muloqot oynasidan mahodisa sodir bo'ladi.

33. System::Windows::Forms::DialogResult r6 = MessageBox::Show(this, message, caption, MessageBoxButtons::OK, MessageBoxIcon::Question, MessageBoxDefaultButton::Button1, (MessageBoxOptions)0, "mspaint.chm", HelpNavigator::KeywordIndex, "ovals" ) – dastur fragmentini tushuntirib bering

34. Asosiy forma ganyang forma qanday qo'shiladi

35. Nima uchun [StartPosition] xususiyatining qiymatini [CenterParent] ga tenglashtiriladi.

36. ShowDialog() va Show() usullarining farqini tushuntirib bering.

37. Oddiy ob'ektni yaratish va konstruktorasosida ob'ektni yaratish farqini tushuntirib bering.

38. Muloqot oynalarini yaratishda foydalanuvchi yangi konstruktor yaratish mumkinmi?

39. [menu] → [project] → [add New item] → [UI] → [Windows Form] ketma ketlik qanday amal bajaradi.

40. Muloqot oynasida gima'lumotlarni qanday qilib asosiy forma ga o'tkazish mumkin.



## AMALIYKO'NIKMAVAMALAKALARNI ANIQLASH HAMDARI VOJLANTIRISH UCHUN ASSISMENT TOPSHIRIG'I.

ASSISMENT TOPSHIRIG'I	
<input type="checkbox"/>	Muloqot oynalarini bilanishlashga oid berilgan quyida gidastur bo'yicha berilgan topshiriqlar kerakli fragmentlari asosida bajarilgani. ☞ <input type="checkbox"/> Bu vazifa larni kichikloyiha sifatida bajariladi, kichikloyiha deganda kamida 100 tavazifa sibo'lganda sturtushuniladi. Jamoabilanishlashga maqsad gamuvofiq.





---

---

---

3. Intellektual turli ma'lumotlarni kiritishni ta'lab qiluvchi IQ testlardan iborat dastur tuzing.  
Loyihabo'yicha qisqacha asosiy fragmentlarni yozing.

---

---

---

---

---

---

---

---

---

---

---

---

4. C++ dasturlash muhitini yarating.  
C++ kompilyatorini vakutub'onalarini tayyor tizimdan oling.  
Loyihabo'yicha qisqacha asosiy fragmentlarni yozing.

---

---

---

---

---

---

---

---

---

---

---

5. Barcha turdagi mulqot oynalarini o'z ichiga olgan, interaktiv savol-javob tizimini yarating va savollarni, javoblarni fayl gasaqalaydigandastur tuzing.  
Loyihabo'yicha qisqacha asosiy fragmentlarni yozing.



DrawLines, DrawPolygon, DrawRectangle, DrawString, FillPie, FillEllipse, Chart, ChartAreas, Series, ChartType, Visual Basic Power Packs, PrintForm, LineShape, OvalShape, RectangleShape, DataRepeater.

☑ **Bilishshartbo'lgantushunchalar.** Grafika, geometriya, xarakterning paydobo'lishi, sinf va sinf ob'ekti, xususiyat, hodisa, forma, komponenta dasturlashga oid dastlabki tushunchalar hamda C++ tilini qo'llab quvvatlovchi muhitda ishlab chiqarishni bilish lozim.

☛ **Bilibolasiz.** Visual C++ muhitida dasturlash, Visual C++ ning grafik koniyatlari uchun Graphics sinfi xususiyatlarini va usullari, Graphics sinfi usul va sinfi dasturlashni turlitirish, matn joylashtirish, rasmlarni yaratish va saqlash, tahrirlash amallari, Chart komponenta xususiyat va hodisalari, turli tipli diagrammalarni hosil qilish usullari, Funktsiyalarni grafiklarini qurish va Visual Basic Power Packs komponentalarini va ularni ishlatish, tasvirlarni harakatlantirish, oqimlarasosida animatsiyalarni yaratish gatalablari, o'rnatish va komponentalar bilan bog'lash usullari, shuningdek, Clip, ClipBounds, CompositingMode, CompositingQuality, DpiX, DpiY, Display, Document, Inch, Millimeter, Pixel, Point, World, Pen, DrawArc, DrawBezier, DrawCurve, DrawEllipse, DrawImage, DrawLines, DrawPolygon, DrawRectangle, DrawString, FillPie, FillEllipse, Chart, ChartAreas, Series, ChartType, Visual Basic Power Packs, PrintForm, LineShape, OvalShape, RectangleShape, DataRepeater usullar va boshqalar xususiyatlari, hodisalari o'rganishingiz mumkin.

## REJA

1. Visual C++ ning grafik koniyatlari
2. Graphics sinfi usul va sinfi dasturlashni qurish
3. Chart komponentalari.
4. Funktsiyalarning grafiklarini qurish.
5. Visual Basic Power Packs komponentalarini va ularni ishlatish.

## KIRISH

Visual C++ o'z grafik tuzilma malumotini taqdim etish uchun ishlatiladi. Grafik deganda ixtiyoriy narsani kompyuterda piksellarda hosil qilish tushuniladi. Pikselni taqdim etish uchun argumentlar berib, ya'ni koordinatalari  $A(x,u)$  ko'rinishida. Har qanday IDE muhitlarida grafik bilan ishlash imkoniyatini beradi. Ularni turlicha nomlanishi mumkin. Ammo, ularning ruzukasi (qalami) vamo'y qalam ibo'ladi. GUI

ko‘rinishdaishlashto‘liqgrafikabilanbog‘liqbo‘lib, texnikaninggrafikrejimdaihlashinita‘minlaydi.

Hozirdadeyarlibarchafoydalanuvchilargrafikimkoniyatlaridanfoydalanibishlaydilar.

### Visual

**C++ninggrafikimkoniyatlari.** Grafikimkoniyatgaegabo‘lgantizimlardaasosan, nuqta, chiziq, to‘rtburchak, aylana, ko‘pburchakkabishakllarniqamraboladi. GUI asosidagibarchaelementlargadiqqatbilanqarasangizshugrafiktuzilmalarda niboratbo‘ladi.

**Visual C++ningimkoniyatlariniko‘rishuchun** Graphics sinfigamurojaatqilamiz.

Businfningnomlarfazosi System.Drawing vakutubxonasi System.Drawing.Common.dll bo‘libhisoblanadi.

Chizishuchun GDI+ modulniinkapsulyatsiyaqiladivabusinfndanmerosxo‘rolishmumkinemas.

Businf MarshalByRefObject, IDisposable, System::Drawing::IdeviceContext gaasoslanganbo‘lib, Object → MarshalByRefObject → Graphics sinfiningmerosxo‘rihisoblanadi.

Grafikasinfiekrangatasvirlarnichizishusullarinio‘ziichigaqamarabolangan.

Grafikamuayyanqurilmanikontekstbilanbog‘lashuchunhamxizmatqiladi.

Grafikob‘ektyordamidako‘pturlishaklvachizizqlarchizishmumkin.

Chizizqlarvashakllarchizishuchunmaxsus DrawGraphicalElement usullarnio‘rganishlozim. Buusullar DrawLine, DrawArc, DrawClosedCurve, DrawPolygon va DrawRectangle o‘zichigaoladi. Chizizqlarvashakllarchizishuchunqalamyordamidavashakllarnito‘ldirishuchunmo‘yqalamyordamidaamalgaoshiriladi.

Grafikasinfimkoniyatlariniuningxususiyatlarivausulariorqaliko‘rsa tibo‘tamiz. Shuningyangiparaemetkelsa, unghamto‘xtalibo‘tiladi.

12.1-jadval.Grafikasinfixususiyatlari.

Xususiyaitnomi	vazifasi
Clip	Grafikaningchizilganchegarasinicheklaydiganchegaranioladiyokio‘rnatadi.
ClipBounds	Grafikaningkesishchegarasinichegaralovchi <u>RectangleF</u> tuzilishinioladi.
CompositingM	Kompozittasvirlarchizishholatinioladiyokio‘rnatadi.

ode	
CompositingQuality	Grafikagachizilgankompozitsiontasvirlarningko'rsatishsifatini o'rnatadi.
DpiX	Ushbugrafikaninggorizontalo'lchaminioladi.
DpiY	Ushbugrafikaningvertikalo'lchaminioladi.
InterpolationMode	Ushbugrafikalarbilanbog'liqinterpolyatsiyarejiminioladi yoki o'rnatadi.
IsClipEmpty	Bugrafikakesishsohasinibo'shyokiyo'qliginiko'rsatib, birqiymatoladi.
IsVisibleClipEmpty	Bugrafikaaniqkesishsohasinibo'shyokiyo'qliginiko'rsatib, birqiymatoladi.
PageScale	Bugrafikauchunsaahifamodulivabirlikmodulio'rtasidache garasinisozlash.
PageUnit	Bugrafikasahifakoordinatalariniuchunishlatiladigan o'lchov birligisozlash.
PixelOffsetMode	Bugrafikako'rsatishpaytidaPikseloffsetqandayko'rsatilgan qiymatsozlash.
RenderingOrigin	Bugrafikako'rsatishrejimini o'rnatadi.
SmoothingMode	Grafikalaruchunko'rsatishsifatinioladiyoki o'rnatadi.
TextContrast	Matnko'rsatishuchun gamma qiymatinibelgilash.
TextRenderingHint	Bugrafikabilanbog'liqmatnuchunko'rsatishrejimini o'rnatadi.
Transform	Grafikalaruchungeometriko'zgarishiningnusxasinioladiyoki o'rnatadi.
VisibleClipBounds	Grafikninganiqkesishsohasinitekslashuchunto'rtburchakoladi.

Grafikasinfusulariquyidagilardaniborat:

1. AddMetafileComment(Byte[]) - Rasmmetafayligaizohqo'shadi. BundaByte[] belgilimassivbo'lib, <System::Byte> ^ data tipidaaniqlanadivamasalan, array<Byte>^metaCom = {(Byte)'T',(Byte)'e',(Byte)'s',(Byte)'t'}; kabianiqlanishimumkin.

2. BeginContainer() - Grafikaninghozirgiholatibilangrafikkonteynernisaqlaydivayangigrafikkonteynerniochadivaishlatadi.

BeginContainer(Rectangle, Rectangle, GraphicsUnit) - Grafikjoriyholatibilanbirgrafikkonteynersaqalaydi, belgilanganparametrlilo'zgartirishbilanyangigrafikkonteynerdanfoydalanadiva aochadi. Bunda Rectangle to'rtburchakbo'lib, Rectangle(0,0,200,200) kabianiqlanadi. Birinchito'rtburchakkonteyneruchunshkalanivaikkinchisikonteyneruchunsohao'zgarishirianiqlaydi. GraphicsUnit - konteyneruchuno'lchovbirliginianiqlashuchunishlatiladi. Uning enum bo'lib, quyidagi 12.2-jadvalgakeltirilganqiymatlarnioladiva GraphicsUnit::Pixel ko'rinishdafoydalaniladi.

12.2-jadval. GraphicsUnit qiymatlari.

Qiymatnomi	vazifasi
Display	Ko'rsatishqurilmasiningo'lchovbiriginibelgilaydi. Odatda video displeylaruchunpiksellarvaprinterlaruchun 1/100 dyuym.
Document	O'lchovbirligisifatidahujjatbirligini (1/300 dyuym) belgilaydi.
Inch	O'lchovbirligisifatidadyuymnibildiradi.
Millimeter	Millimetrnio'lchovbirligisifatidabelgilaydi.
Pixel	O'lchovbirligisifatidaqurilmapikselinibelgilaydi.
Point	O'lchovbirligisifatidaprinterningnuqtasini (1/72 dyuym) belgilaydi.
World	Koordinatatizimibiriginio'lchovbirligisifatidabelgilaydi.

BeginContainer(RectangleF, RectangleF, GraphicsUnit) - grafikjoriyholatibilanbirgrafikkonteynersaqalaydi, belgilangansohalio'zgartirishbilanyangigrafikkonteynerdanfoydalanadiva aochadi. Bininchiikkitaparametlartuzilmabo'lib,

ikkixilaniqlanishimumkin:

RectangleF(PointF(30.0F,40.0F),SizeF(50.0F,100.0F)) va RectangleF(float x, float y, float width, float height); GraphicsUnit - esakonteyneruchuno'lchovbirliginianiqlashuchunishlatiladi.

3. Clear(Color) - butunchizmayuzasinitozalaydivabelgilanganfonrangibilantoidiradi. Color tuzilmabo'libuningqiymatlarigakengaytirishamali [::] bilanmurojaatqiladilar.

4.CopyFromScreen(Int32, Int32, Int32, Int32, Size) - Piksellito'rtburchakkamosranglima'lumotlarniekrandangrafikaningchiz mayuzasigabit-blokliuzatishniamalgaoshiradi. CopyFromScreen(int

sourceX, int sourceY, int destinationX, int destinationY, System::Drawing::Size blockRegionSize) - bundax, ulartu rrt burchakning mos koordinatalariva Size sohaning o'lchamidir. Hamtuzilmabo'lib, Size(int width, int height) ko'rinishdayaratiladi.

CopyFromScreen(Int32, Int32, Int32, Int32, Size, CopyPixelOperation) -

Piksellito'rt burchak kamos rang lima'lumotlarniekrandangrafikaningchiz mayuzasigabit-blokliuzatishniamalgaoshiradi.

Buningparemetrlariyuqoridagidekbo'lib,

CopyPixelOperationdagimanbarangiyakuniyranggaolibkeladiganmaqsad rangibilanbirlashtirilganliginianiqlaydiva Blackness, CaptureBlt, DestinationInvert, MergeCopy, MergePaint, NoMirrorBitmap, NotSourceCopy, NotSourceErase, PatCopy, PatInvert, PatPaint, SourceAnd, SourceCopy, SourceErase, SourceInvert, SourcePaint, Whiteness kabiqiy matlarniqabulqiladi.

CopyFromScreen(Point, Point, Size) -

Ekrandangrafikaningchizmayuzasigapiksellito'rt burchak kamos rang lima'lumotlarnibit-blokliuzatishniamalgaoshiradi. Point -

bunuqtayaratu vchituzilmabo'lib, o'zining 3ta Point(Int32), Point(Int32, Int32), Point(Size) konstruktorlar mavjud.

CopyFromScreen(Point, Point, Size, CopyPixelOperation) - ekrandangrafikaningchizmayuzasigapiksellito'rt burchak kamos rang lima'lumotlarnibit-blokliuzatishniamalgaoshiradi.

5. CreateObjRef(Type) -

ob'ekt bilan muloqot qilish uchun ishlatiladigan proksi ishlab chiqarish uchun zarur bo'lgan barcha tegishli lima'lumotlarni o'z ichiga olgan ob'ektyaratadi.

Type - requestedType tipidagi abstrakt tip, array<Type^> { int::typeid, int::typeid }); kabianiqlanadi.

6. Dispose() - Bu grafikatomonidan ishlatiladigan barcha resurslarni null qiladi.

7. DrawArc. Bu funksiya cho'lchovlisohani ifodalovchi yoynichizadi. Uning quyidagi variantlaribor:

DrawArc(Pen, Int32, Int32, Int32, Int32, Int32, Int32) - void  
DrawArc(System::Drawing::Pen ^ pen, int x, int y, int width, int height, int startAngle, int sweepAngle);

DrawArc(Pen, Rectangle, Single, Single) - void  
DrawArc(System::Drawing::Pen ^ pen,



```
System::Drawing::Rectangle rect, float startAngle, float sweepAngle);
```

```
DrawArc(Pen, RectangleF, Single, Single) - void
```

```
DrawArc(System::Drawing::Pen ^ pen,
```

```
System::Drawing::RectangleF rect, float startAngle, float sweepAngle);
```

```
DrawArc(Pen, Single, Single, Single, Single, Single, Single) - void
```

```
DrawArc(System::Drawing::Pen ^ pen, float x, float y, float width, float height, float startAngle, float sweepAngle);
```

DrawArc funksiyasida Pen buqalambo‘lib, uninguchun Pen sinfi ishlatiladi. Pen ning 4 takonstruktor mavjud bo‘lib, Pen(Brush), Pen(Brush, Single), Pen(Color), Pen(Color, Single) konstruktorlardan foydalaniladi,

xususiyat va usullarini o‘zgartirish orqali o‘rnatish mumkin. Brush bumo‘yqalambo‘lib, Brush sinfi ishlatiladi. Brush ning Brush() konstruktor mavjud bo‘lib,

xususiyat va usullarini o‘zgartirish orqali o‘rnatish mumkin.

Funksiyaning qolgan parametrlari barchasi yuqoridagi keltirilgan parametrlar kabi amalga oshiriladi.

8. DrawBezier. Bu funksiya berilgan 4 tanuqtaliso hauchun sohasini chizadi. Uning 3 taturibor:

```
DrawBezier(Pen, Point, Point, Point, Point) - void
```

```
DrawBezier(System::Drawing::Pen ^ pen, System::Drawing::Point pt1, System::Drawing::Point pt2, System::Drawing::Point pt3, System::Drawing::Point pt4);
```

```
DrawBezier(Pen, PointF, PointF, PointF, PointF) - void
```

```
DrawBezier(System::Drawing::Pen ^ pen, System::Drawing::PointF pt1, System::Drawing::PointF pt2, System::Drawing::PointF pt3, System::Drawing::PointF pt4);
```

```
DrawBezier(Pen, Single, Single, Single, Single, Single, Single, Single, Single, Single) - void DrawBezier(System::Drawing::Pen ^ pen, float x1, float y1, float x2, float y2, float x3, float y3, float x4, float y4);
```

9. DrawBeziers. Bu funksiya berilgan nuqtalar sohasini chizadi. Uning 2 taturibor:

```

    DrawBeziers(Pen,          Point[])          -          void
DrawBeziers(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::Point> ^ points);

```

```

    DrawBeziers(Pen,          PointF[])         -          void
DrawBeziers(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::PointF> ^ points);

```

Funksiyada Point[] - ikki o'ldirish tekislikdani nuqtalar belgilaydigan x va y koordinatalar uchun tartibli juftlikni ifodalaydimassivdir.

10. DrawClosedCurve.

Bufunksiyayopiq egrichiziq chizish gamo'ljallangan. Uning 4 taturibor:

```

    DrawClosedCurve(Pen,          Point[])          -          void
DrawClosedCurve(System::Drawing::Pen ^ pen,
cli::array <System::Drawing::Point> ^ points);

```

```

    DrawClosedCurve(Pen, Point[], Single, FillMode) - void
DrawClosedCurve(System::Drawing::Pen ^ pen,
cli::array <System::Drawing::Point> ^ points, float
tension,
System::Drawing::Drawing2D::FillMode
fillmode);

```

```

    DrawClosedCurve(Pen,          PointF[])         -          void
DrawClosedCurve(System::Drawing::Pen ^ pen,
cli::array <System::Drawing::PointF> ^ points);

```

```

    DrawClosedCurve(Pen, PointF[], Single, FillMode) - void
DrawClosedCurve(System::Drawing::Pen ^ pen,
cli::array <System::Drawing::PointF> ^ points, float
tension,
System::Drawing::Drawing2D::FillMode
fillmode);

```

Bufunksiyada FillMode tipibor, busohanibo'yaashturini belgilaydi. FillMode ning 2 xilusulibor, Alternate - Muqobil to'ldirish va Winding o'rab to'ldirish rejimlarini belgilaydi.

11. DrawCurve. Bufunksiyaberilgan parametrlar asosida spaly (egrichiziq) chizadi. Uning 7 taturimavjud.

```

    DrawCurve(Pen, Point[])          -          void
DrawCurve(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::Point> ^ points);

```

```

    DrawCurve(Pen, Point[], Int32, Int32, Single) - void
DrawCurve(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::Point> ^ points, int offset, int
numberOfSegments, float tension);

```

```

    DrawCurve(Pen, Point[], Single) - void
DrawCurve(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::Point> ^ points, float tension);
    DrawCurve(Pen, PointF[]) - void
DrawCurve(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::PointF> ^ points);
    DrawCurve(Pen, PointF[], Int32, Int32) - void
DrawCurve(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::PointF> ^ points, int offset, int
numberOfSegments);
    DrawCurve(Pen, PointF[], Int32, Int32, Single) - void
DrawCurve(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::PointF> ^ points, int offset, int
numberOfSegments, float tension);
    DrawCurve(Pen, PointF[], Single) - void
DrawCurve(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::PointF> ^ points, float tension);

```

Bufunksiyaning barcha parametrlaritasniflangan.

12. DrawEllipse.

Bufunksiyaberilgan parametrlar asosida ellips chizadi. Uning 4 taturim mavjud.

```

    DrawEllipse(Pen, Int32, Int32, Int32, Int32) - void
DrawEllipse(System::Drawing::Pen ^ pen, int x, int y,
int width, int height);
    DrawEllipse(Pen, Rectangle) - void
DrawEllipse(System::Drawing::Pen ^ pen,
System::Drawing::Rectangle rect);
    DrawEllipse(Pen, RectangleF) - void
DrawEllipse(System::Drawing::Pen ^ pen,
System::Drawing::RectangleF rect);

```

```

    DrawEllipse(Pen, Single, Single, Single, Single) - void
DrawEllipse(System::Drawing::Pen ^ pen, float x,
float y, float width, float height);

```

Bufunksiyaning barcha parametrlaritasniflangan.

13. DrawIcon.

Bufunksiyaberilgan parametrlar asosida belgilangan belgibo'y chizadi. Uning 3 taturim mavjud.

```

    DrawIcon(Icon, Int32, Int32) - void
DrawIcon(System::Drawing::Icon ^ icon, int x, int y);
    DrawIcon(Icon, Rectangle) - void
DrawIcon(System::Drawing::Icon ^ icon,
System::Drawing::Rectangle targetRect);
    DrawIconUnstretched(Icon, Rectangle) - void
DrawIconUnstretched(System::Drawing::Icon ^ icon,
System::Drawing::Rectangle targetRect);
    Bufunksiyaning barcha parametrlaritasniflangan.

```

#### 14. DrawImage.

Bufunksiyaberilgan parametrlar asosidabelgilangantasvirnichizadi. Uning 30 taturimavjud.

```

    DrawImage(Image, Int32, Int32) - void
DrawImage(System::Drawing::Image ^ image, int x, int
y);
    DrawImage(Image, Int32, Int32, Int32, Int32) - void
DrawImage(System::Drawing::Image ^ image, int x, int
y, int width, int height);
    DrawImage(Image, Int32, Int32, Rectangle, GraphicsUnit) - void
DrawImage(System::Drawing::Image ^ image, int x, int
y, System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit srcUnit);
    DrawImage(Image, Point) - void
DrawImage(System::Drawing::Image ^ image,
System::Drawing::Point point);
    DrawImage(Image, Point[]) - void
DrawImage(System::Drawing::Image ^ image, cli::array
<System::Drawing::Point> ^ destPoints);
    DrawImage(Image, Point[], Rectangle, GraphicsUnit) - void
DrawImage(System::Drawing::Image ^ image, cli::array
<System::Drawing::Point> ^ destPoints,
System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit srcUnit);
    DrawImage(Image, Point[], Rectangle, GraphicsUnit,
ImageAttributes) - void DrawImage(System::Drawing::Image ^
image, cli::array <System::Drawing::Point> ^
destPoints, System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit srcUnit,

```

```

System::Drawing::Imaging::ImageAttributes ^
imageAttr);
    DrawImage(Image, Point[], Rectangle, GraphicsUnit,
ImageAttributes, Graphics+DrawImageAbort) - void
DrawImage(System::Drawing::Image ^ image, cli::array
<System::Drawing::Point> ^ destPoints,
System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Imaging::ImageAttributes ^
imageAttr, System::Drawing::Graphics::DrawImageAbort
^ callback);
    DrawImage(Image, Point[], Rectangle, GraphicsUnit,
ImageAttributes, Graphics+DrawImageAbort, Int32) - void
DrawImage(System::Drawing::Image ^ image, cli::array
<System::Drawing::Point> ^ destPoints,
System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Imaging::ImageAttributes ^
imageAttr, System::Drawing::Graphics::DrawImageAbort
^ callback, int callbackData);
    DrawImage(Image, PointF) - void
DrawImage(System::Drawing::Image ^ image,
System::Drawing::PointF point);
    DrawImage(Image, PointF[]) - void
DrawImage(System::Drawing::Image ^ image, cli::array
<System::Drawing::PointF> ^ destPoints);
    DrawImage(Image, PointF[], RectangleF, GraphicsUnit) -
void DrawImage(System::Drawing::Image ^ image,
cli::array <System::Drawing::PointF> ^ destPoints,
System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit);
    DrawImage(Image, PointF[], RectangleF, GraphicsUnit,
ImageAttributes) - void
DrawImage(System::Drawing::Image ^ image, cli::array
<System::Drawing::PointF> ^ destPoints,
System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit,

```

```

System::Drawing::Imaging::ImageAttributes ^
imageAttr);
    DrawImage(Image, PointF[], RectangleF, GraphicsUnit,
ImageAttributes, Graphics+DrawImageAbort) - void
DrawImage(System::Drawing::Image ^ image, cli::array
<System::Drawing::PointF> ^ destPoints,
System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Imaging::ImageAttributes ^
imageAttr, System::Drawing::Graphics::DrawImageAbort
^ callback);
    DrawImage(Image, PointF[], RectangleF, GraphicsUnit,
ImageAttributes, Graphics+DrawImageAbort, Int32) -
<System::Drawing::PointF> ^ destPoints,
System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Imaging::ImageAttributes ^
imageAttr, System::Drawing::Graphics::DrawImageAbort
^ callback, int callbackData);
    DrawImage(Image, Rectangle) - void
DrawImage(System::Drawing::Image ^ image,
System::Drawing::Rectangle rect);
    DrawImage(Image, Rectangle, Int32, Int32, Int32, Int32,
GraphicsUnit) - void DrawImage(System::Drawing::Image ^
image, System::Drawing::Rectangle destRect, int srcX,
int srcY, int srcWidth, int srcHeight,
System::Drawing::GraphicsUnit srcUnit);
    DrawImage(Image, Rectangle, Int32, Int32, Int32, Int32,
GraphicsUnit, ImageAttributes) - void
DrawImage(System::Drawing::Image ^ image,
System::Drawing::Rectangle destRect, int srcX, int
srcY, int srcWidth, int srcHeight,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Imaging::ImageAttributes ^
imageAttr);
    DrawImage(Image, Rectangle, Int32, Int32, Int32, Int32,
GraphicsUnit, ImageAttributes, Graphics+DrawImageAbort) - void
DrawImage(System::Drawing::Image ^ image,

```

```

System::Drawing::Rectangle destRect, int srcX, int
srcY, int srcWidth, int srcHeight,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Imaging::ImageAttributes ^
imageAttr, System::Drawing::Graphics::DrawImageAbort
^ callback);

```

```

DrawImage(Image, Rectangle, Int32, Int32, Int32, Int32,
GraphicsUnit, ImageAttributes, Graphics+DrawImageAbort, IntPtr) -

```

```

void DrawImage(System::Drawing::Image ^ image,
System::Drawing::Rectangle destRect, int srcX, int
srcY, int srcWidth, int srcHeight,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Imaging::ImageAttributes ^
imageAttrs, System::Drawing::Graphics::DrawImageAbort
^ callback, IntPtr callbackData);

```

```

DrawImage(Image, Rectangle, Rectangle, GraphicsUnit) - void
DrawImage(System::Drawing::Image ^ image,
System::Drawing::Rectangle destRect,
System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit srcUnit);

```

```

DrawImage(Image, Rectangle, Single, Single, Single, Single,
GraphicsUnit) - void DrawImage(System::Drawing::Image ^
image, System::Drawing::Rectangle destRect, float
srcX, float srcY, float srcWidth, float srcHeight,
System::Drawing::GraphicsUnit srcUnit);

```

```

DrawImage(Image, Rectangle, Single, Single, Single, Single,
GraphicsUnit, ImageAttributes) - void
DrawImage(System::Drawing::Image ^ image,
System::Drawing::Rectangle destRect, float srcX,
float srcY, float srcWidth, float srcHeight,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Imaging::ImageAttributes ^
imageAttrs);

```

```

DrawImage(Image, Rectangle, Single, Single, Single, Single,
GraphicsUnit, ImageAttributes, Graphics+DrawImageAbort) - void
DrawImage(System::Drawing::Image ^ image,
System::Drawing::Rectangle destRect, float srcX,
float srcY, float srcWidth, float srcHeight,

```

```

System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Imaging::ImageAttributes ^
imageAttrs, System::Drawing::Graphics::DrawImageAbort
^ callback);

```

```

DrawImage(Image, Rectangle, Single, Single, Single, Single,
GraphicsUnit, ImageAttributes, Graphics+DrawImageAbort, IntPtr) -
void DrawImage(System::Drawing::Image ^ image,
System::Drawing::Rectangle destRect, float srcX,
float srcY, float srcWidth, float srcHeight,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Imaging::ImageAttributes ^
imageAttrs, System::Drawing::Graphics::DrawImageAbort
^ callback, IntPtr callbackData);

```

```

DrawImage(Image, RectangleF) - void
DrawImage(System::Drawing::Image ^ image,
System::Drawing::RectangleF rect);

```

```

DrawImage(Image, RectangleF, RectangleF, GraphicsUnit) - void
DrawImage(System::Drawing::Image ^ image,
System::Drawing::RectangleF destRect,
System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit);

```

```

DrawImage(Image, Single, Single) - void
DrawImage(System::Drawing::Image ^ image, float x,
float y);

```

```

DrawImage(Image, Single, Single, RectangleF, GraphicsUnit)
- void DrawImage(System::Drawing::Image ^ image, float
x, float y, System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit);

```

```

DrawImage(Image, Single, Single, Single, Single) - void
DrawImage(System::Drawing::Image ^ image, float x,
float y, float width, float height);

```

Bufunksiyaningparametrlariko‘pchiligitasvirlargaishlovberishbilanb  
 og‘liq. Shuninguchuno‘maishvaqtidazarurlarparametrlarikeltirilgan.  
 Shungdek, birxulosashundaniboratki,  
 meno‘zimbularninghammasiniishlatibko‘rmaganman.

15. DrawImageUnscaled.

Bufunksiyaberilganparametrlarasosidatasvirnibelgilanganjoydaaslkattali  
 gidanfoydalanibchizadi. Uning 5 taturimavjud.



```
DrawImageUnscaled(Image, Int32, Int32) - void
DrawImageUnscaled(System::Drawing::Image ^ image, int
x, int y);
```

```
DrawImageUnscaled(Image, Int32, Int32, Int32, Int32) - void
DrawImageUnscaled(System::Drawing::Image ^ image, int
x, int y, int width, int height);
```

```
DrawImageUnscaled(Image, Point) - void
DrawImageUnscaled(System::Drawing::Image ^ image,
System::Drawing::Point point);
```

```
DrawImageUnscaled(Image, Rectangle) - void
DrawImageUnscaled(System::Drawing::Image ^ image,
System::Drawing::Rectangle rect);
```

```
DrawImageUnscaledAndClipped(Image, Rectangle) - void
DrawImageUnscaledAndClipped(System::Drawing::Image ^
image, System::Drawing::Rectangle rect);
```

Bufunksiyaining barcha parametrlarita‘lilqilingan.

16. DrawLine. Bufunksiyaberilgan parametrlarasosidachiziqchizadi. Uning 4 taturimavjud.

```
DrawLine(Pen, Int32, Int32, Int32, Int32) - void
DrawLine(System::Drawing::Pen ^ pen, int x1, int y1,
int x2, int y2);
```

```
DrawLine(Pen, Point, Point) - void
DrawLine(System::Drawing::Pen ^ pen,
System::Drawing::Point pt1, System::Drawing::Point
pt2);
```

```
DrawLine(Pen, PointF, PointF) - void
DrawLine(System::Drawing::Pen ^ pen,
System::Drawing::PointF pt1, System::Drawing::PointF
pt2);
```

```
DrawLine(Pen, Single, Single, Single, Single) - void
DrawLine(System::Drawing::Pen ^ pen, float x1, float
y1, float x2, float y2);
```

Bufunksiyaining barcha parametrlarita‘lilqilingan.

22. DrawLines. Bufunksiyaberilgan parametrlarasosida ulanadigan chiziqchizadi. Uning 2 taturimavjud.

```
DrawLines(Pen, Point[]) - void
DrawLines(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::Point> ^ points);
```

```
DrawLines(Pen, PointF[]) - void
DrawLines(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::PointF> ^ points);
```

Bufunksiyaining barcha parametrlaritaʼlilqilingan.

17. DrawPath.

Bufunksiyaberilgan parametrlarasosidabogʻlanganchiziqlarvae grichiziqla rqatorinichizadi. Businfmeros xoʻrboʻlishim umkin emas.

```
DrawPath(Pen, GraphicsPath) - void
DrawPath(System::Drawing::Pen ^ pen,
System::Drawing::Drawing2D::GraphicsPath ^ path);
```

18. DrawPie. Bufunksiyaberilgan parametrlarasosida Koordinata jufti, kengligi, balandligi va ikkita radial chiziq bilan belgilangan ellips bilan belgilangan nok shaklini chizadi. Uning 4 taturimavjud.

```
DrawPie(Pen, Int32, Int32, Int32, Int32, Int32, Int32) - void
DrawPie(System::Drawing::Pen ^ pen, int x, int y, int
width, int height, int startAngle, int sweepAngle);
```

```
DrawPie(Pen, Rectangle, Single, Single) - void
DrawPie(System::Drawing::Pen ^ pen,
System::Drawing::Rectangle rect, float startAngle,
float sweepAngle);
```

```
DrawPie(Pen, RectangleF, Single, Single) - void
DrawPie(System::Drawing::Pen ^ pen,
System::Drawing::RectangleF rect, float startAngle,
float sweepAngle);
```

```
DrawPie(Pen, Single, Single, Single, Single, Single, Single) - void
DrawPie(System::Drawing::Pen ^ pen, float x, float y,
float width, float height, float startAngle, float
sweepAngle);
```

Bufunksiyaining barcha parametrlaritaʼlilqilingan.

19. DrawPolygon.

Bufunksiyaberilgan parametrlarasosidakoʻp shaklini chizadi. Uning 4 taturimavjud.

```
DrawPolygon(Pen, Point[]) - void
DrawPolygon(System::Drawing::Pen ^ pen, cli::array
<System::Drawing::Point> ^ points);
```

DrawPolygon(Pen, PointF[]) - void

```
DrawPolygon(System::Drawing::Pen ^ pen, cli::array  
<System::Drawing::PointF> ^ points);
```

Bufunksiyaining barcha parametrlarita'liqlingan.

20. DrawRectangle.

Bufunksiyaberilgan parametrlarasosidato'rt burchak shaklini chizadi.

Uning 3 taturim mavjud.

```
DrawRectangle(Pen, Int32, Int32, Int32, Int32) - void
```

```
DrawRectangle(System::Drawing::Pen ^ pen, int x, int  
y, int width, int height);
```

```
DrawRectangle(Pen, Rectangle) - void
```

```
DrawRectangle(System::Drawing::Pen ^ pen,  
System::Drawing::Rectangle rect);
```

```
DrawRectangle(Pen, Single, Single, Single, Single) - void
```

```
DrawRectangle(System::Drawing::Pen ^ pen, float x,  
float y, float width, float height);
```

Bufunksiyaining barcha parametrlarita'liqlingan.

21. DrawRectangles.

Bufunksiyaberilgan parametrlarasosida ulanadigan to'rt burchak shaklini chizadi. Uning 3 taturim mavjud.

```
DrawRectangles(Pen, Rectangle[]) - void
```

```
DrawRectangles(System::Drawing::Pen ^ pen, cli::array  
<System::Drawing::Rectangle> ^ rects);
```

```
DrawRectangles(Pen, RectangleF[]) - void
```

```
DrawRectangles(System::Drawing::Pen ^ pen, cli::array  
<System::Drawing::RectangleF> ^ rects);
```

Bufunksiyaining barcha parametrlarita'liqlingan.

22. DrawString.

Bufunksiyaberilgan parametrlarasosidashrift ob'ektlar bilan belgilangan joy dabelgilangan matn chizadi (joylashtiradi). Uning 6 taturim mavjud.

```
DrawString(String, Font, Brush, PointF) - void
```

```
DrawString(System::String ^ s, System::Drawing::Font  
^ font, System::Drawing::Brush ^ brush,  
System::Drawing::PointF point);
```

```
DrawString(String, Font, Brush, PointF, StringFormat) - void
```

```
DrawString(System::String ^ s, System::Drawing::Font  
^ font, System::Drawing::Brush ^ brush,  
System::Drawing::PointF point,  
System::Drawing::StringFormat ^ format);
```

```
DrawString(String, Font, Brush, RectangleF) - void
DrawString(System::String ^ s, System::Drawing::Font
^ font, System::Drawing::Brush ^ brush,
System::Drawing::RectangleF layoutRectangle);
```

```
DrawString(String, Font, Brush, RectangleF, StringFormat) - void
DrawString(System::String ^ s, System::Drawing::Font
^ font, System::Drawing::Brush ^ brush,
System::Drawing::RectangleF layoutRectangle,
System::Drawing::StringFormat ^ format);
```

```
DrawString(String, Font, Brush, Single, Single) - void
DrawString(System::String ^ s, System::Drawing::Font
^ font, System::Drawing::Brush ^ brush, float x,
float y);
```

```
DrawString(String, Font, Brush, Single, Single, StringFormat) -
void DrawString(System::String ^ s,
System::Drawing::Font ^ font, System::Drawing::Brush
^ brush, float x, float y,
System::Drawing::StringFormat ^ format);
```

Bufunksiyaining barcha parametrlarita‘lilqilingan.

23. EndContainer.

Bufunksiyaberilgan parametrlarasosidajoriy grafik konteynerni yopadiva  
BeginContainer()

usuli uchun ko‘rsatkich orqali saqlangan bug grafik holatini chizadi.

```
EndContainer(GraphicsContainer) - void
EndContainer(System::Drawing::Drawing2D::GraphicsCont
ainer ^ container);
```

Bufunksiyaining barcha parametrlarita‘lilqilingan.

24. EnumerateMetafile.

Bufunksiyaberilgan parametrlarasosidamatn nimeta faylga uzatadivabelgilan  
ngan joydacha chizadi (joylashtiradi). Uning 36 taturim mavjud.

```
EnumerateMetafile(Metafile, Point,
Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::Point destPoint,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback);
```

```
EnumerateMetafile(Metafile, Point,
Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
```

```

^   metafile,      System::Drawing::Point   destPoint,
System::Drawing::Graphics::EnumerateMetafileProc   ^
callback, IntPtr callbackData);
    EnumerateMetafile(Metafile,                Point,
Graphics+EnumerateMetafileProc, IntPtr, ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^   metafile,      System::Drawing::Point   destPoint,
System::Drawing::Graphics::EnumerateMetafileProc   ^
callback,                IntPtr                callbackData,
System::Drawing::Imaging::ImageAttributes         ^
imageAttr);
    EnumerateMetafile(Metafile, Point, Rectangle, GraphicsUnit,
Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^   metafile,      System::Drawing::Point   destPoint,
System::Drawing::Graphics::EnumerateMetafileProc   ^
callback,                IntPtr                callbackData,
System::Drawing::Imaging::ImageAttributes         ^
imageAttr);
    EnumerateMetafile(Metafile, Point, Rectangle, GraphicsUnit,
Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^   metafile,      System::Drawing::Point   destPoint,
System::Drawing::Rectangle                srcRect,
System::Drawing::GraphicsUnit                srcUnit,
System::Drawing::Graphics::EnumerateMetafileProc   ^
callback, IntPtr callbackData);
    EnumerateMetafile(Metafile, Point, Rectangle, GraphicsUnit,
Graphics+EnumerateMetafileProc, IntPtr, ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^   metafile,      System::Drawing::Point   destPoint,
System::Drawing::Rectangle                srcRect,
System::Drawing::GraphicsUnit                unit,
System::Drawing::Graphics::EnumerateMetafileProc   ^
callback,                IntPtr                callbackData,
System::Drawing::Imaging::ImageAttributes         ^
imageAttr);

```

```

EnumerateMetafile(Metafile, Point[],
Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::Point> ^
destPoints,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback);
EnumerateMetafile(Metafile, Point[],
Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::Point> ^
destPoints,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData);
EnumerateMetafile(Metafile, Point[],
Graphics+EnumerateMetafileProc, IntPtr, ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::Point> ^
destPoints,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData,
System::Drawing::Imaging::ImageAttributes ^
imageAttr);
EnumerateMetafile(Metafile, Point[], Rectangle, GraphicsUnit,
Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::Point> ^
destPoints, System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback);
EnumerateMetafile(Metafile, Point[], Rectangle, GraphicsUnit,
Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::Point> ^
destPoints, System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit srcUnit,

```

```

System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData);
    EnumerateMetafile(Metafile, Point[], Rectangle, GraphicsUnit,
Graphics+EnumerateMetafileProc, IntPtr, ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::Point> ^
destPoints, System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit unit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData,
System::Drawing::Imaging::ImageAttributes ^
imageAttr);
    EnumerateMetafile(Metafile, PointF,
Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::PointF destPoint,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback);
    EnumerateMetafile(Metafile, PointF,
Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::PointF destPoint,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData);
    EnumerateMetafile(Metafile, PointF,
Graphics+EnumerateMetafileProc, IntPtr, ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::PointF destPoint,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData,
System::Drawing::Imaging::ImageAttributes ^
imageAttr);
    EnumerateMetafile(Metafile, PointF, RectangleF, GraphicsUnit,
Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::PointF destPoint,
System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit,

```

```

System::Drawing::Graphics::EnumerateMetafileProc ^
callback);
    EnumerateMetafile(Metafile, PointF, RectangleF, GraphicsUnit,
Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::PointF destPoint,
System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData);
    EnumerateMetafile(Metafile, PointF, RectangleF, GraphicsUnit,
Graphics+EnumerateMetafileProc, IntPtr, ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::PointF destPoint,
System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit unit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData,
System::Drawing::Imaging::ImageAttributes ^
imageAttr);
    EnumerateMetafile(Metafile, PointF[],
Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::PointF> ^
destPoints,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback);
    EnumerateMetafile(Metafile, PointF[],
Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::PointF> ^
destPoints,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData);
    EnumerateMetafile(Metafile, PointF[],
Graphics+EnumerateMetafileProc, IntPtr, ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::PointF> ^

```



```

destPoints,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback,          IntPtr          callbackData,
System::Drawing::Imaging::ImageAttributes ^
imageAttr);
    EnumerateMetafile(Metafile, PointF[], RectangleF, GraphicsUnit,
Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::PointF> ^
destPoints, System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback);
    EnumerateMetafile(Metafile, PointF[], RectangleF, GraphicsUnit,
Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::PointF> ^
destPoints, System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData);
    EnumerateMetafile(Metafile, PointF[], RectangleF, GraphicsUnit,
Graphics+EnumerateMetafileProc, IntPtr, ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, cli::array <System::Drawing::PointF> ^
destPoints, System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit unit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData,
System::Drawing::Imaging::ImageAttributes ^
imageAttr);
    EnumerateMetafile(Metafile, Rectangle,
Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::Rectangle destRect,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback);

```

```

EnumerateMetafile(Metafile, Rectangle,
Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::Rectangle destRect,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData);
EnumerateMetafile(Metafile, Rectangle,
Graphics+EnumerateMetafileProc, IntPtr, ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::Rectangle destRect,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData,
System::Drawing::Imaging::ImageAttributes ^
imageAttr);
EnumerateMetafile(Metafile, Rectangle, Rectangle, GraphicsUnit,
Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::Rectangle destRect,
System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback);
EnumerateMetafile(Metafile, Rectangle, Rectangle, GraphicsUnit,
Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::Rectangle destRect,
System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData);
EnumerateMetafile(Metafile, Rectangle, Rectangle, GraphicsUnit,
Graphics+EnumerateMetafileProc, IntPtr, ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::Rectangle destRect,
System::Drawing::Rectangle srcRect,
System::Drawing::GraphicsUnit unit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData,

```

```

System::Drawing::Imaging::ImageAttributes ^
imageAttr);
    EnumerateMetafile(Metafile, RectangleF,
Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::RectangleF destRect,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback);
    EnumerateMetafile(Metafile, RectangleF,
Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::RectangleF destRect,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData);
    EnumerateMetafile(Metafile, RectangleF,
Graphics+EnumerateMetafileProc, IntPtr, ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::RectangleF destRect,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData,
System::Drawing::Imaging::ImageAttributes ^
imageAttr);
    EnumerateMetafile(Metafile, RectangleF, RectangleF,
GraphicsUnit, Graphics+EnumerateMetafileProc) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::RectangleF destRect,
System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback);
    EnumerateMetafile(Metafile, RectangleF, RectangleF,
GraphicsUnit, Graphics+EnumerateMetafileProc, IntPtr) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::RectangleF destRect,
System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit srcUnit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData);

```

```

EnumerateMetafile(Metafile, RectangleF, RectangleF,
GraphicsUnit, Graphics+EnumerateMetafileProc, IntPtr,
ImageAttributes) - void
EnumerateMetafile(System::Drawing::Imaging::Metafile
^ metafile, System::Drawing::RectangleF destRect,
System::Drawing::RectangleF srcRect,
System::Drawing::GraphicsUnit unit,
System::Drawing::Graphics::EnumerateMetafileProc ^
callback, IntPtr callbackData,
System::Drawing::Imaging::ImageAttributes ^
imageAttr);

```

25. Equals.

Belgilanganob'ektjoriyob'ektgatengyokiyo'qliginianiqlaydi.

```
Equals(Object) -
```

Belgilanganob'ektjoriyob'ektgatengyokiyo'qliginianiqlaydi.

26. ExcludeClip.

Bufunksiyaberilganparametrlarasosidako'rsatilganmaydonniistisnoqilish uchunushbugrafikaningkliphududiniyangilaydi. Uning 2 taturimavjud.

```
ExcludeClip(Rectangle) - void
```

```
ExcludeClip(System::Drawing::Rectangle rect);
```

```
ExcludeClip(Region) - void
```

```
ExcludeClip(System::Drawing::Region ^ region);
```

Bufunksiyainingbarchaparametrlarita'lilqilingan.

27. FillClosedCurve.

Bufunksiyaberilganparametrlarasosidayopiqegrichiziqisplaynnibo'yyalg anholdachizadi. Uning 6 taturimavjud.

```
FillClosedCurve(Brush, Point[]) - void
```

```
FillClosedCurve(System::Drawing::Brush ^ brush,
cli::array <System::Drawing::Point> ^ points);
```

```
FillClosedCurve(Brush, Point[], FillMode) - void
```

```
FillClosedCurve(System::Drawing::Brush ^ brush,
cli::array <System::Drawing::Point> ^ points,
System::Drawing::Drawing2D::FillMode fillmode);
```

```
FillClosedCurve(Brush, Point[], FillMode, Single) - void
```

```
FillClosedCurve(System::Drawing::Brush ^ brush,
cli::array <System::Drawing::Point> ^ points,
System::Drawing::Drawing2D::FillMode fillmode, float
tension);
```

```

    FillClosedCurve(Brush, PointF[]) - void
FillClosedCurve(System::Drawing::Brush ^ brush,
cli::array <System::Drawing::PointF> ^ points);
    FillClosedCurve(Brush, PointF[], FillMode) - void
FillClosedCurve(System::Drawing::Brush ^ brush,
cli::array <System::Drawing::PointF> ^ points,
System::Drawing::Drawing2D::FillMode fillmode);
    FillClosedCurve(Brush, PointF[], FillMode, Single)- void
FillClosedCurve(System::Drawing::Brush ^ brush,
cli::array <System::Drawing::PointF> ^ points,
System::Drawing::Drawing2D::FillMode fillmode, float
tension);

```

Bufunksiyaining barcha parametrlari ta'riflangan.

28. FillEllipse.

Bufunksiyaberilgan parametrlar asosida ellipsni bo'laydigan holda chizadi.  
Uning 4 taturim mavjud.

```

    FillEllipse(Brush, Int32, Int32, Int32, Int32) - void
FillEllipse(System::Drawing::Brush ^ brush, int x,
int y, int width, int height);
    FillEllipse(Brush, Rectangle) - void
FillEllipse(System::Drawing::Brush ^ brush,
System::Drawing::Rectangle rect);
    FillEllipse(Brush, RectangleF) - void
FillEllipse(System::Drawing::Brush ^ brush,
System::Drawing::RectangleF rect);
    FillEllipse(Brush, Single, Single, Single, Single) - void
FillEllipse(System::Drawing::Brush ^ brush, float x,
float y, float width, float height);

```

Bufunksiyaining barcha parametrlari ta'riflangan.

29. FillPath.

Bufunksiyaberilgan parametrlar asosida bog'langan va egrichiziqlar bo'laydigan holda chizadi.

```

    FillPath(Brush, GraphicsPath) - void
FillPath(System::Drawing::Brush ^ brush,
System::Drawing::Drawing2D::GraphicsPath ^ path);

```

Bufunksiyaining barcha parametrlari ta'riflangan.

30. FillPie.

Bufunksiyaberilgan parametrlar asosida bir juft koordinatalar, kenglik,

balandlikvaikkitaradialchiziqbilanbelgilanganellipsnibo‘yalganholdachizadi. Uning 3 taturimavjud.

```
FillPie(Brush, Int32, Int32, Int32, Int32, Int32, Int32) - void
FillPie(System::Drawing::Brush ^ brush, int x, int y,
int width, int height, int startAngle, int
sweepAngle);
```

```
FillPie(Brush, Rectangle, Single, Single) - void
FillPie(System::Drawing::Brush ^ brush,
System::Drawing::Rectangle rect, float startAngle,
float sweepAngle);
```

```
FillPie(Brush, Single, Single, Single, Single, Single, Single) -
void FillPie(System::Drawing::Brush ^ brush,
float x, float y, float width, float height, float
startAngle, float sweepAngle);
```

Bufunksiyainingbarchaparametrlarita‘lilqilingan.

31. FillPolygon.

Bufunksiyaberilganparametrlarasosidamaydonnibo‘yalganholdachizadi. Uning 4 taturimavjud.

```
FillPolygon(Brush, Point[]) - void
FillPolygon(System::Drawing::Brush ^ brush,
cli::array <System::Drawing::Point> ^ points,
System::Drawing::Drawing2D::FillMode fillMode);
```

```
FillPolygon(Brush, Point[], FillMode) - void
FillPolygon(System::Drawing::Brush ^ brush,
cli::array <System::Drawing::PointF> ^ points);
```

```
FillPolygon(Brush, PointF[]) - void
FillPolygon(System::Drawing::Brush ^ brush,
cli::array <System::Drawing::PointF> ^ points,
System::Drawing::Drawing2D::FillMode fillMode);
```

```
FillPolygon(Brush, PointF[], FillMode) - void
FillRectangle(System::Drawing::Brush ^ brush, int x,
int y, int width, int height);
```

Bufunksiyainingbarchaparametrlarita‘lilqilingan.

32. FillRectangle.

Bufunksiyaberilganparametrlarasosidato‘rtburchaknibo‘yalganholdachizadi. Uning 4 taturimavjud.

```
FillRectangle(Brush, Int32, Int32, Int32, Int32) - void
FillRectangle(System::Drawing::Brush ^ brush,
System::Drawing::Rectangle rect);
```

```

FillRectangle(Brush, Rectangle) - void
FillRectangle(System::Drawing::Brush ^ brush,
System::Drawing::RectangleF rect);
FillRectangle(Brush, RectangleF) - void
FillRectangle(System::Drawing::Brush ^ brush, float
x, float y, float width, float height);
FillRectangle(Brush, Single, Single, Single, Single) - void
FillRectangles(System::Drawing::Brush ^ brush,
cli::array <System::Drawing::Rectangle> ^ rects);

```

Bufunksiyaining barcha parametrlarita' l'ilqilingan.

33. FillRectangles.

Bufunksiyaberilgan parametrlarasosida ulanadigan to'rt burchakni bo'lgan yalga n holda chizadi. Uning 2 ta turim mavjud.

```

FillRectangles(Brush, Rectangle[]) - void
FillRectangles(System::Drawing::Brush ^ brush,
cli::array <System::Drawing::RectangleF> ^ rects);
FillRectangles(Brush, RectangleF[]) - void
FillRegion(System::Drawing::Brush ^ brush,
System::Drawing::Region ^ region);

```

Bufunksiyaining barcha parametrlarita' l'ilqilingan.

34. FillRegion.

Bufunksiyaberilgan parametrlarasosida soha ni bo'lgan yalga n holda chizadi.

```

FillRegion(Brush, Region) - void
FillRegion(System::Drawing::Brush ^ brush,
System::Drawing::Region ^ region);

```

Grafika sinfining 34 ta usullarini varinatlaris osidakeltirdik.

Bu usullarni va xususiyatlar nita svirlarni qurish orqaliko'rib chiqamiz.

### Graphics

**sinfi usulari asosidat asvirlarni qurish.** Tasvirlarni qurish uchun **PictureBox** komponentasidan foydalanamiz.

**Chiziqchizish.** Bunday komponentaning berilgan joyida, rangli chiziqchizish niko'ramiz. Buning uchun 1 ta komponenta, 1 ta tugmava 4 ta `textBox`

komponentalarini oynaga qulay qilib joylashtiramiz.

1-qadam.

Oynaning

`Form1_Load`

hodisasida quyidagi algoritmni joylashtiramiz.

```

this->Text = "DrawLine - chiziq chizish";
button1->Text = "CHIZISH";

```

2-qadam.

KomponentavateTextBoxxususiyatlarinisozlashamallarnibajarishmumkin.

3-qadam. Chiziqchizishuchunnuqtalargaatabint  
myPoint[4];o'zgaruvchisiolamiz.

4-qadam. Komponentaning Paint  
deghanhodisasigaquyidagialgoritmnijoylashtiramiz.

```
e->Graphics->DrawLine(System::Drawing::Pens::Red,  
myPoint[1], myPoint[2], myPoint[3],  
myPoint[4]);
```

5-qadam.

Tugmaningbutton1\_Clickhodisasidaquyidagialgoritmnijoylashtirami  
Z.

```
myPoint[1] = Convert::ToInt32(textBox1->Text);  
myPoint[2] = Convert::ToInt32(textBox2->Text);  
myPoint[3] = Convert::ToInt32(textBox3->Text);  
myPoint[4] = Convert::ToInt32(textBox4->Text);  
pictureBox1->Refresh();
```

Bundamatndansontipigao'tishamalidanfoydalanilganvaharbirtugma  
bosilgandakomponentaningchizmasio'zgaribturadi.

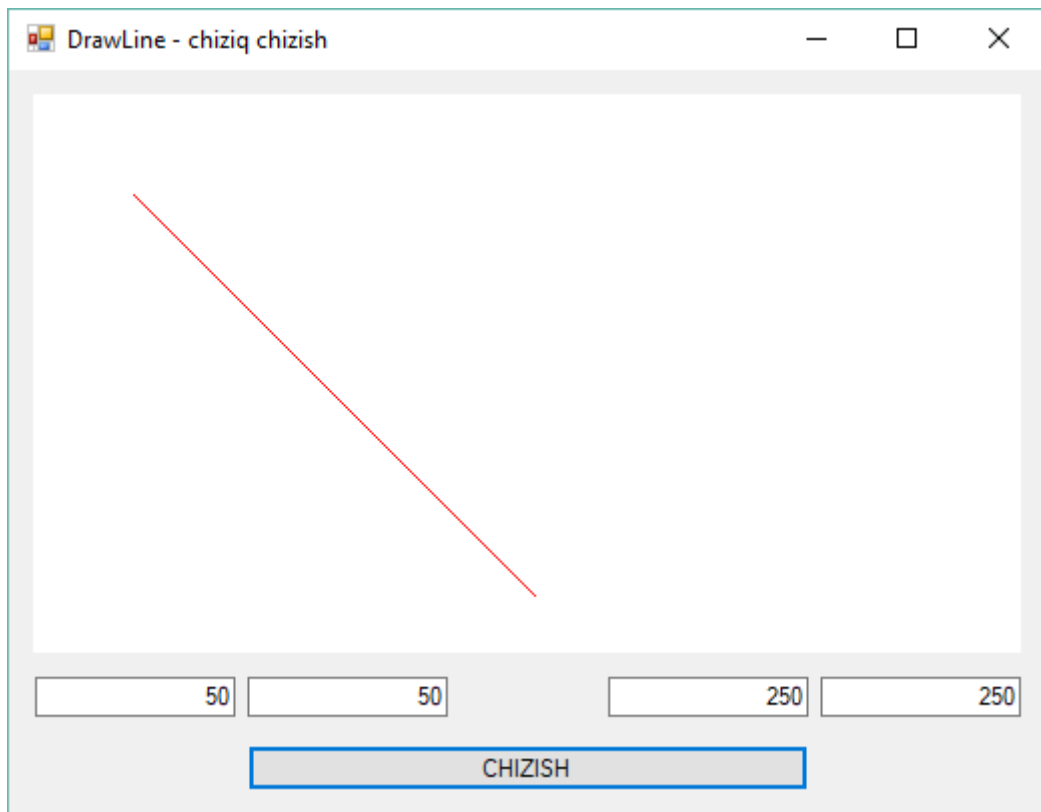
Agarloyhaniishgatushursangiz, textBoxlargakeraklisonlarnikiritib,  
chiziqchizishmumkin.

Komponentaningo'lchamidankattachiziqqlarnichizishmumkinammoko'ri  
nmaydi. Unibrshqarishlozim,

ya'nikeraklikattalikdanoshibketgandafoydalanuvchigamuloqotoynasibila  
nxabarberishmumkin, buni  
textBoxlarningfokusio'zgargandaamalgaoshirishmaqsadgamuvofiq.

Chiziqchizishningnatijasi 12.1-rasmdakeltirilgan.





12.1-rasm. Chiziqchizishnatijasi.

**Uchburchakchizish.** Bundak komponentaning berilgan joyida, rangli chiziqlar orqali uchburchak chizishni ko'ramiz. Buning uchun 1 ta komponenta, 1 ta tugmava 6 ta TextBox komponentalarini oynaga qulay qilib joylashtiramiz. Chunki 3 ta chiziqning tutashtirish, uchburchak hosil bo'ladi.

1-qadam. Oynaning Form1\_Load hodisasida quyidagi algoritmni joylashtiramiz.

```
this->Text = "DrawLine - uchburchak chizish";
button1->Text = "CHIZISH";
```

2-qadam.

Komponenta va TextBox xususiyatlarini sozlash amallarni bajarish mumkin.

3-qadam. Chiziqchizish uchun nuqtalarga a ta b int myPoint[6]; o'zgaruvchisi o'lamiz.

4-qadam. Komponentaning Paint degan hodisasi ga quyidagi algoritmni joylashtiramiz.

```
e->Graphics->DrawLine(System::Drawing::Pens::Red,
                        myPoint[1], myPoint[2], myPoint[3],
myPoint[4]);
e->Graphics->DrawLine(System::Drawing::Pens::Black,
                        myPoint[3], myPoint[4], myPoint[5],
myPoint[6]);
```

```
e->Graphics->DrawLine(System::Drawing::Pens::Blue,
                        myPoint[1], myPoint[2], myPoint[5],
myPoint[6]);
```

5-qadam.

Tugmaningbutton1\_Clickhodisidaquyidagialgoritmnijoylashtirami  
Z.

```
myPoint[1] = Convert::ToInt32(textBox1->Text);
myPoint[2] = Convert::ToInt32(textBox2->Text);
myPoint[3] = Convert::ToInt32(textBox3->Text);
myPoint[4] = Convert::ToInt32(textBox4->Text);
myPoint[5] = Convert::ToInt32(textBox3->Text);
myPoint[6] = Convert::ToInt32(textBox4->Text);
pictureBox1->Refresh();
```

Agar loyihani ish gatushursangiz, textBoxlarga kerakli sonlarni kiritib,  
uch burchakni chizish mumkin.

**Ellipischizish.** Bunda komponentaning berilgan joyida,  
rangli ellipischizish niko‘ramiz. Buning uchun 1 ta komponenta, 1  
ta tugmava 4 ta textBox  
komponentalarini oynaga qulay qilib joylashtiramiz.

1-qadam. Oynaning Form1\_Load  
hodisida quyidagialgoritmni joylashtiramiz.

```
this->Text = "Ellipse - chizish";
button1->Text = "CHIZISH";
```

2-qadam.

Komponenta va textBox xususiyatlarini sozlash amallarni bajarish mumkin.

3-qadam. Chiziqchizish uchun nuqtalarga a ta b int  
myPoint[4]; o‘zgaruvchisi olamiz.

4-qadam. Komponentaning Paint  
degan hodisiga quyidagialgoritmni joylashtiramiz.

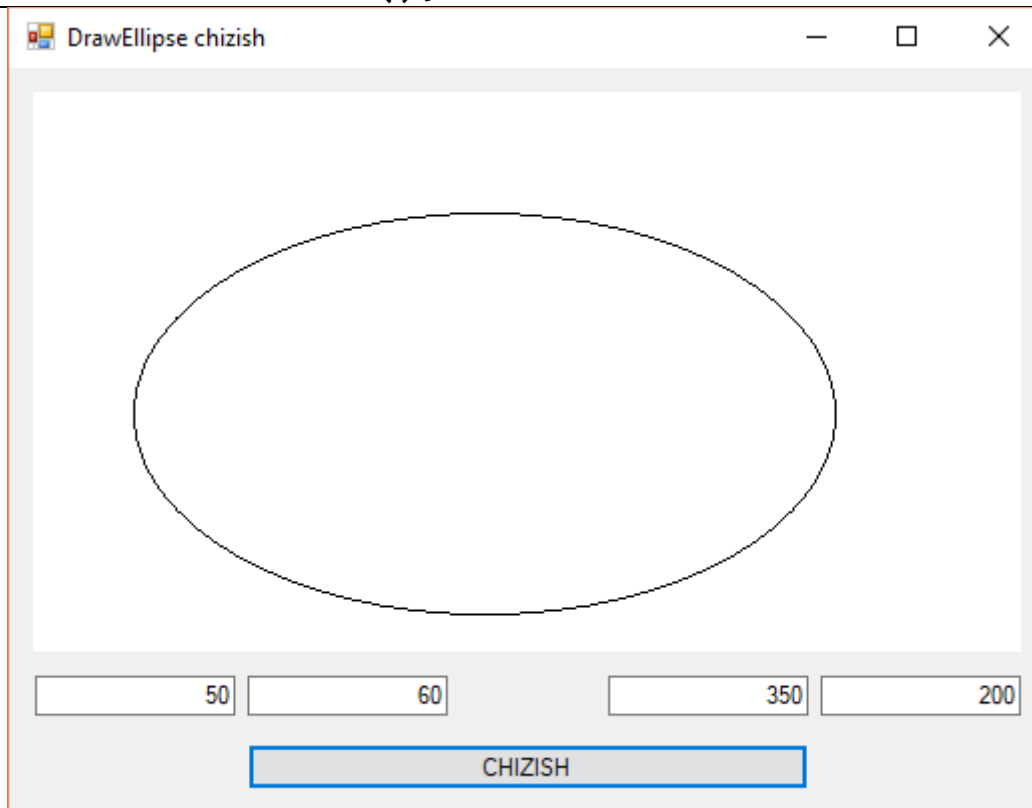
```
Pen ^ pen = gcnw Pen(Color::Black);
e->Graphics->DrawEllipse(pen,
                        myPoint[1], myPoint[2], myPoint[3],
myPoint[4]);
```

5-qadam.

Tugmaningbutton1\_Clickhodisidaquyidagialgoritmni joylashtirami  
Z.

```
myPoint[1] = Convert::ToInt32(textBox1->Text);
myPoint[2] = Convert::ToInt32(textBox2->Text);
```

```
myPoint[3] = Convert::ToInt32(textBox3->Text);
myPoint[4] = Convert::ToInt32(textBox4->Text);
pictureBox1->Refresh();
```



12.2-rasm. DrawEllipsechizishnatijasi.

Yuqoridakeltirilganchizishlarorqaliboshqaixtiyoriyshakllarnihamchi zishmumkin. Turlishakllarnichizibbo‘yyashniko‘ramiz.

**Turlishakllarnibo‘yyash.**Bundakkomponentaningberilganjoyida, bo‘yyalganshakllarnichizishniko‘ramiz. Buninguchun**Brush** – mo‘yqalamdanfoydalanib, 1 takomponenta, 1 ta label va 1 ta Combox komponentalarinioynagaqulayqilibjoylashtiramiz.

1-qadam. Oynaning Form1\_Load hodisasidaquyidagialgoritmnijoylashtiramiz.

```
this->Text = "Shaklni bo‘yash";
label1->Text = "Kerakli shaklni
tanlang:";
comboBox1->Text = "Shakllar";
comboBox1->Items-
>Add("To‘rtburchak");
comboBox1->Items->Add("Ellipsis");
comboBox1->Items->Add("Doira");
comboBox1->Items->Add("Qirqilgan
doira ");
```

2-qadam.

KomponentavaComboBoxxususiyatlarinisozlashamallarnibajarishmumkin.

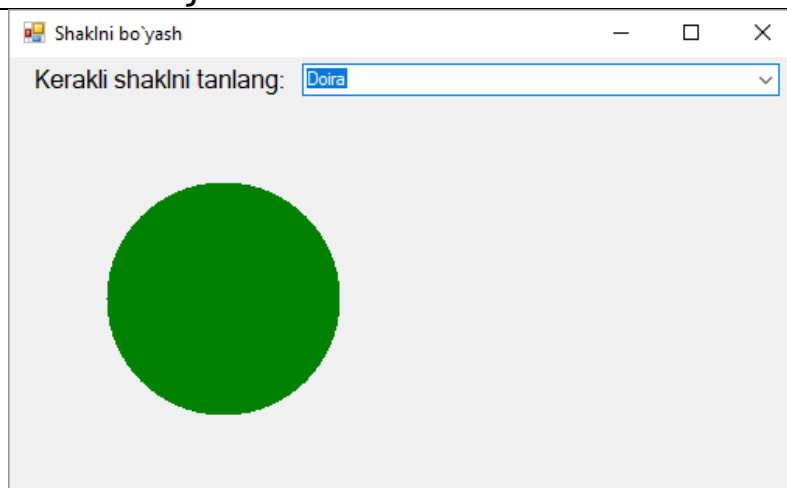
3-qadam.

ComboBox

komponentasiningcomboBox1\_SelectedIndexChangedhodisasidaquyidagialgoritmnijoylashtiramiz.

```
Graphics ^grp = pictureBox1->CreateGraphics();
Brush ^ brsh = gcnew SolidBrush(Color::Green);
grp->Clear(SystemColors::Control);

switch (comboBox1->SelectedIndex){
    case 0:
        grp->FillRectangle(brsh,50,50,150,150);
break;
    case 1:
        grp->FillEllipse(brsh,50,50,300,150); break;
    case 2:
        grp->FillEllipse(brsh,50,50,150,150); break;
    case 3:
        grp->FillPie(brsh,50,50,150,150,150,100);
break;
    default:
grp->Clear(SystemColors::Control);break;
}
```



12.3-rasm. Turlishakllarnibo'yyashnatijasi.

**Ko'pburchaklarnichizish.** Bundakomponentaningberilganjoyida, bo'yyalganfoydalanuvchishaklarnichizishniko'ramiz.

Buninguchun **Brush** – mo‘yqalamdan foydalanib, 1 takomponenta, 1 ta button komponentasini oynaga qulay qilib joylashtiramiz.

1-qadam. Oynaning Form1\_Load hodisasida quyidagi algoritmni joylashtiramiz.

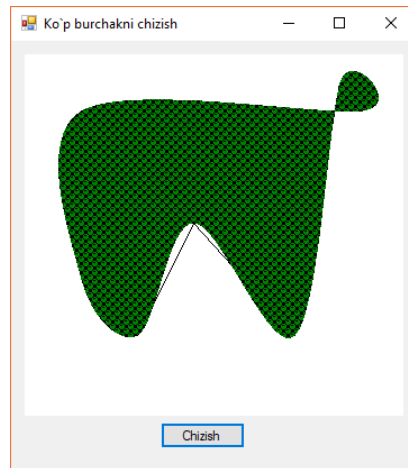
```
this->Text = "Ko‘p burchakni  
chizish";  
button1->Text = "Chizish";
```

2-qadam. Komponentava button xususiyatlarini sozlash amallarni bajarish mumkin.

3-qadam. button komponentasining button1\_Click hodisasida quyidagi algoritmni joylashtiramiz.

```
Graphics^ graf = pictureBox1->CreateGraphics();  
Pen^ gPen = gcnew Pen( Color::Black, 1);  
HatchBrush^ HBrush = gcnew HatchBrush(  
HatchStyle::Sphere, Color::Green, Color::Black );  
  
Point point1 = Point(50, 50);  
Point point2 = Point(50, 200);  
Point point3 = Point(100, 250);  
Point point4 = Point(150, 150);  
Point point5 = Point(240, 250);  
Point point6 = Point(280, 25);  
Point point7 = Point(300, 50);  
array<Point> ^Points =  
{point1, point2, point3, point4, point5, point6, point7};  
  
graf->DrawPolygon( gPen, Points );  
graf->FillClosedCurve( HBrush, Points );
```

Bu algoritmdagi HatchBrush sinfi ishlatish uchun using namespace System::Drawing::Drawing2D; nomlar fazosini qo‘shib qo‘yish kerak. Algoritmda 7 nuqtalarni birlashtirish orqali shakl chiziladi.



12.4-rasm. Turlishakllarnibo‘yyashnatijasi.

**Hodisalarorqalishaklchizish.** Hamma "Paint" kabidasturlarnibiladi. Uningengajoyibxususiyatlaridanbirisichqonchabilanchiziqqlarchizishdir. Buniamalgaoshirishuchunkomponentningsichqonchabilashishlash "**Mou sDown**",

"**MausUp**" i "**MouseMove**" hodisalaridan foydalanish mumkin.

Dasturning algoritmi quyidagicha:

foydalanuvchi sichqonchani chaptugmasinibosgandakursororqasigajud ako‘pkichikkvadratlar chizilaboshlaydi.

Bukvadratlar hajmikodda ko‘rsatilgan. Bundantashqari, "tugma" tugmachasini rasimgatushiradiganshaklgako‘chirishingizkerak.

Bundakomponentaning berilgan joyida, sichqonchahodisalariorqalishakllarnichizishniko‘ramiz.

Buning uchun **Brush** – mo‘yqalamdan foydalanib, 1 ta komponenta, 1 ta button komponentasini oynaga qulay qilib joylashtiramiz.

1-qadam. Oynaning Form1\_Load hodisasida quyidagi algoritmni joylashtiramiz.

```
button1->Text = "Tozalash";
this->Text = "Shakl chizish";
```

2-qadam. Komponentava button xususiyatlarini sozlash amallarni bajarish mumkin.

3-qadam. button komponentasining button1\_Click hodisasida quyidagi algoritmni joylashtiramiz. Bu algoritm komponentaning tozalash uchun ishlatiladi.

```
Graphics ^ grp = pictureBox1->CreateGraphics();
grp->Clear(SystemColors::Window);
```

4-qadam.

Komponentaning sichqonchahodisalarini qaytaishlash uchun mantiqiy biro' zgaruvchi olinadi, uning qiymati [0] bo'lsin. `Bool Drow = false;`

5-qadam.

Sichqonchak komponentaning ustiga kelganda mantiqiy o'zgaruvchining qiymati 1 ga o'zgaradi. Chunki chizishni boshlash uchun. Buni amalga oshirish uchun sichqonchani `pictureBox1_MouseDown` hodisiga `Drow = true;` ni yozib qo'yamiz.

6-qadam.

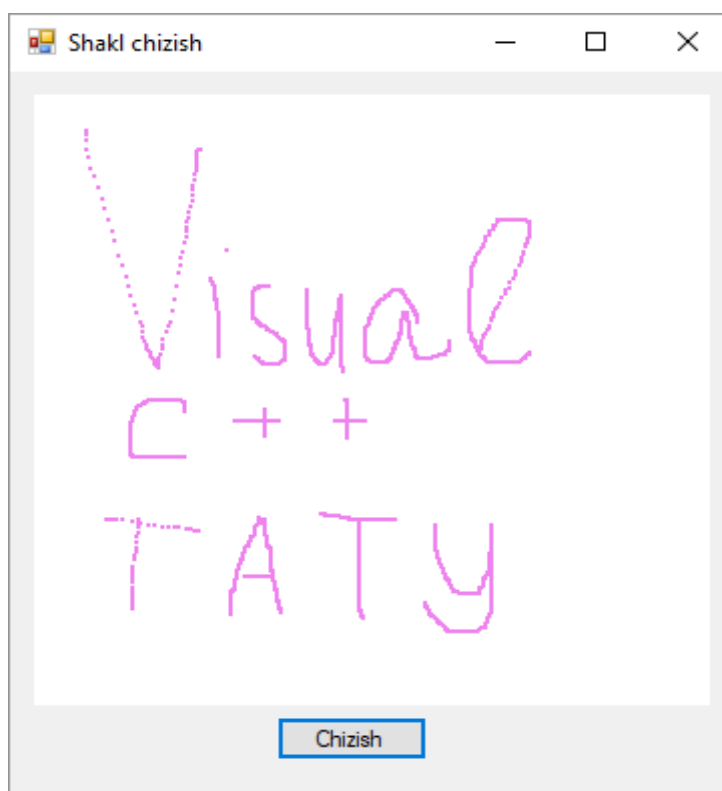
Sichqonchak komponentaning ustidan ketganda mantiqiy o'zgaruvchining qiymati 0 ga o'zgaradi. Chunki chizishni tugatish uchun. Buni amalga oshirish uchun sichqonchani `pictureBox1_MouseUp` hodisiga `Drow = false;` ni yozib qo'yamiz.

7-qadam.

Sichqonchak komponentaning ustiga kelganda chizishi uchun quyidagi algoritmni kiritamiz.

```
Graphics^ graf = pictureBox1->CreateGraphics();
if (Drow == true){
    graf->FillEllipse(Brushes::Violet, e->X, e->Y,
3,3); // mo'y qalam qalinligi
}
```

Algoritmda shakllarni hosil qilish uchun juda kichik ellipschalardan foydalanamiz.



## 12.5-rasm. Sichqonchabilanturlishakllarnichizish.

**Rasmlarnio‘zgartirish.** Buninguchun fayllarni yuklash **openFileDialog** va fayllarni saqlash **saveFileDialog** komponentalaridan foydalanamiz.

Buninguchun **Brush** – mo‘yqalamdan foydalanib, 1 ta komponenta, 3 ta button komponentasini oynaga qulay qilib joylashtiramiz. Button larning birinchisi, rasmni tahrirlashni tozalashni amalga oshiradi, ikkinchisi, tahrirlangan rasmni saqlash uchun, uchinchi, rasmni tahrirlash uchun yuklashni amalga oshiradi. `using namespace System::Drawing::Drawing2D;` ni qo‘shish b qo‘yish kerak.

1-qadam.

Rasmlarnio‘zgartirish uchun ishlatiladigan o‘zgaruvchilarni e‘lon qilamiz.

```
private: Bitmap ^ bmp_for_draw;
private: Point start_point;
private: bool Draw;
public: Pen^ pen_for_draw;
private: String ^ full_name_of_image;
```

2-qadam.

Dasturoynasining `Form1_Load` hodisasi gadastlab kisozlamalarni o‘rnatamiz.

```
this->Text = "Rasmni tahrirlash";
button1->Text = "Tozalash";
button2->Text = "Rasmni saqlash";
button3->Text = "Rasmni yuklash";
pen_for_draw = gcnew Pen(Color::Black, 4);
pen_for_draw->StartCap =
System::Drawing::Drawing2D::LineCap::Round;
pen_for_draw->EndCap =
System::Drawing::Drawing2D::LineCap::Round;
```

Bundak komponentalarni sozlash va bir qalamni yaratib olish, uning `StartCap`, `EndCap` xususiyatlarini o‘rnatish amalga oshiriladi.

3-qadam.

Komponentdagi rasmni tozalash uchun `button1_Click` hodisasi ga algoritmni yozamiz.

```
Graphics^ grp = pictureBox1->CreateGraphics();
grp->Clear(SystemColors::Window);
```

4-qadam.

Sichqoncha komponentaning ustidakelgandachizishi uchun `pictureBox1_MouseDown` hodisasi ga quyidagi algoritmni kiritamiz



```

if (e->Button ==
System::Windows::Forms::MouseButtons::Left){
    Draw = true;
    start_point = e->Location;
}

```

5-qadam.

Sichqonchakomponentaningustidaketgandachizishnito‘xtatishuchun pictureBox1\_MouseUp hodisasigaquyidagialgoritmnikiritamiz

```

if (e->Button ==
System::Windows::Forms::MouseButtons::Left){
    Draw = false;
}

```

6-qadam.

Komponentagakeraklirasmniyuklashuchun button3\_Clickhodisasigaquyidagialgoritmniyoziadi

```

OpenFileDialog ^ open_dialog = gcnew
OpenFileDialog();
open_dialog->Filter = "Image
Files (*.BMP;*.JPG;*.GIF;*.PNG)|*.BMP;*.JPG;*.GIF;*.PN
G|All files (*.*)|*.*";
if (open_dialog->ShowDialog() ==
System::Windows::Forms::DialogResult::OK){
    try{
        full_name_of_image = open_dialog->FileName;
        bmp_for_draw = gcnew Bitmap(open_dialog-
>FileName);
        //pictureBox1->Size = bmp_for_draw->Size;
        pictureBox1->SizeMode =
PictureBoxSizeMode::StretchImage;
        pictureBox1->Image = bmp_for_draw;
        pictureBox1->Invalidate();
    } catch(Exception^ e) {
        System::Windows::Forms::DialogResult rezult =
MessageBox::Show("Tanlangan faylni ochib bo'lmaydi "+
e->ToString(),"Diqqat", MessageBoxButtons::OK,
MessageBoxIcon::Error);
    }}

```

Bundagiasosinuosonshundaniboratki,  
agarrasmningo‘lchamlarikomponentaningo‘lchamlaribilanbixilbo‘lsa,

yaxshichizadi,  
foydalanuvchiekraniningproporsiyasiniolibchizadi.

birxilbo'lmasa,

7-qadam. Komponentadagirasmnisaqlashuchun  
button2\_Clickhodisasigaquyidagialgoritmniyoziladi

```
if (pictureBox1->Image != nullptr){
String ^format = full_name_of_image-
>Substring(full_name_of_image->Length - 4, 4);
SaveFileDialog ^savedialog = gcnew SaveFileDialog();
savedialog->Title = "Rasmni saqlash ...";
savedialog->OverwritePrompt = true;
savedialog->CheckPathExists = true;
savedialog->Filter = "Image Files(*.BMP)|*.BMP|Image
Files(*.JPG)|*.JPG|Image Files(*.GIF)|*.GIF|Image
Files(*.PNG)|*.PNG|All files (*.*)|*.*";
savedialog->ShowHelp = true;
// If selected, save
if (savedialog->ShowDialog() ==
System::Windows::Forms::DialogResult::OK){
    try{
bmp_for_draw->Save(savedialog->FileName,
System::Drawing::Imaging::ImageFormat::Jpeg);
} catch(Exception ^ e) {
System::Windows::Forms::MessageBox::Show("Impossible
to save image"+e->ToString(), "FATAL ERROR",
MessageBoxButtons::OK, MessageBoxIcon::Error);
} } }
```

8-qadam.

Sichqonchakomponentaningustidakelgandachizishiuchunquyidagialgorit  
mnikiritamiz.

```
if (e->Button ==
System::Windows::Forms::MouseButton::Left){
Graphics ^graf = Graphics::FromImage(pictureBox1-
>Image);
graf->DrawLine(pen_for_draw, start_point, e-
>Location);
start_point = e->Location;
pictureBox1->Invalidate();
}
```

Dasturni ishlatganda quyidagi oyna chiqadi.



12.6-rasm. Rasmlarni tahrirlash.

**Matnlarni tasvir kabijoylashtirish.** Bunday komponentaning belgilan gan joyida, aniqo' lchamlivarangli matn ni joylashtirish niko' ramiz. Buning uchun komponentaga 3 ta textbox va 1 ta botton komponentlarini qulay qilib joylashtiramiz. Birinchisi matn yozish uchun, qolganiki tasimatn ni joylashtirish nuqtasi uchun.

1-qadam. Textbox nirejimini multiline rejimigava scrolBars xususiyatiga Vertical qiymati o' rnatiladi.

2-qadam. Oynaning Form1\_Load hodisasida quyidagi algoritmni joylashtiramiz.

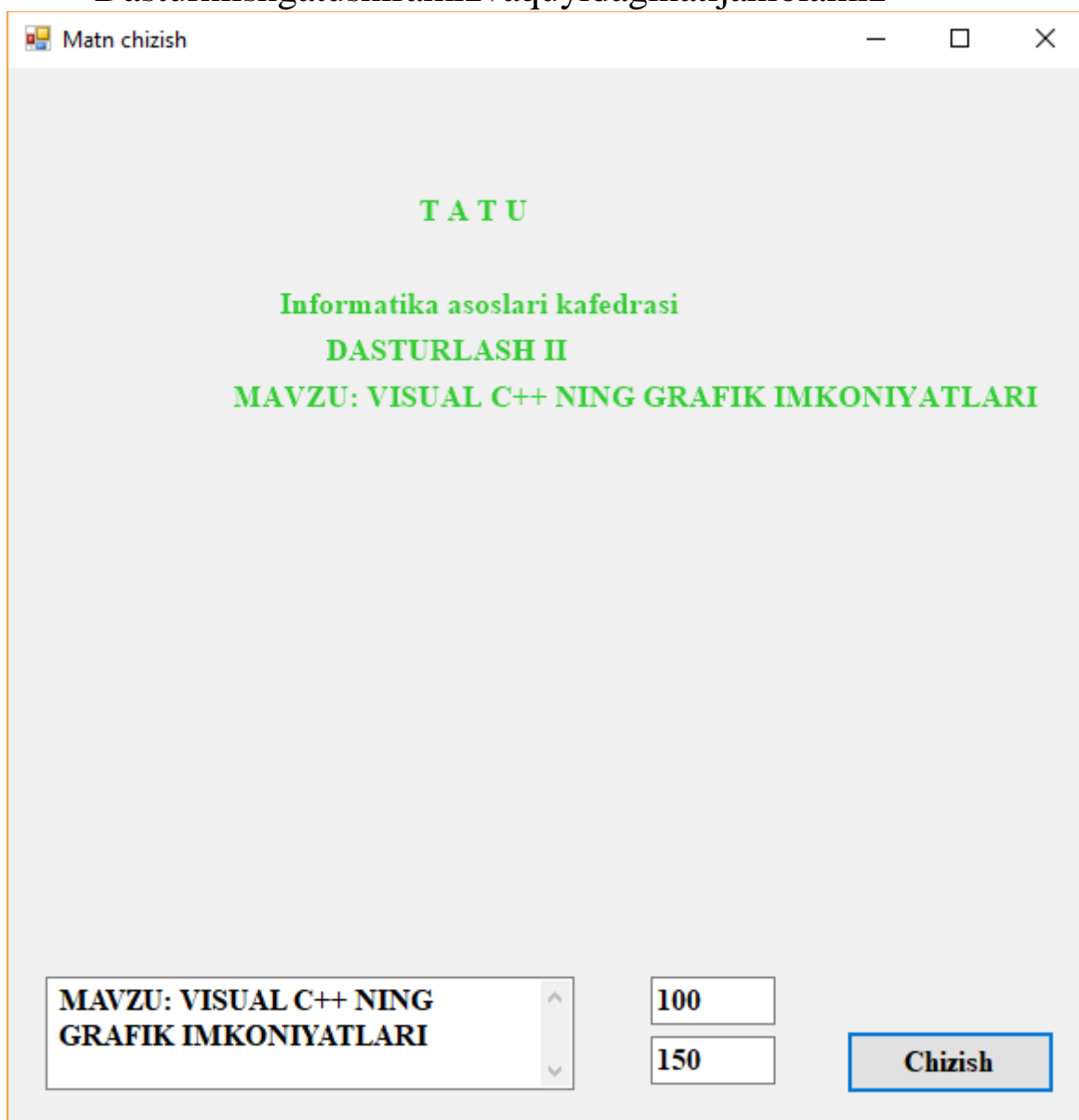
```
Font = gcnew System::Drawing::Font("Times New Roman",  
12, FontStyle::Bold);  
button1->Text = "Chizish";  
this->Text = "Matn chizish";
```

3-qadam. Komponentava button xususiyatlarini sozlash amallarni bajarish mumkin.

4-qadam. button komponentasining button1\_Click hodisasida quyidagi algoritmni joylashtiramiz. Bualgoritm komponentaning tozalash uchun ishlatiladi.

```
String^ Text = String::Format("{0}", textBox1->Text);
Brush^ brsh = gcnew SolidBrush(Color::LimeGreen);
Graphics^ G = pictureBox1->CreateGraphics();
G->TextRenderingHint =
System::Drawing::Text::TextRenderingHint::AntiAlias;
int xP = Convert::ToInt16(textBox2->Text);
int yP = Convert::ToInt32(textBox3->Text);
G->DrawString(Text, Font, brsh, xP, yP); //
joylashtirish manzili
```

Dasturni ishga tushiramiz va quyidagini taniyamiz



12.7-rasm. Matnlarni joylashtirish.

**Matnlirasmlarni yaratish.** Matnlirasmlarni yaratish uchun 1 takomponenta, 4 ta label, 4 ta button, 4 ta textbox larni qulay qilib joylashtiriladi.

1-qadam. 2 ta Textbox nirejimini multiline rejimigava scrolBars xususiyatiga Vertical qiymatio‘rnatiladi.

2-qadam. Oynaning Form1\_Load hodisasidaquyidagialgoritmnijoylashtiramiz.

```
this->label1->Text = "| Yuqoridagi  
matn";  
  
label2->Text = "|";  
label3->Text = "| Pastdagi matn";  
label4->Text = "|";  
button1->Text = "Tozalsh";  
button2->Text = "Chizish";  
button3->Text = "Saqlash";  
button4->Text = "Rasm yuklash";
```

3-qadam.

Komponentavabuttonxususiyatlarinisozlashamallarnibajarishmumkin.

4-qadam. TextBox3 va TextBox4 nimosravishda TextBox1 va TextBox2 larorqasigatashlaymiz.

5-qadam. Formaning formBorderStyle xususiyatiga SizableToolWindow nio‘rnatamiz.

Formanio‘lchamlarinio‘zgartirmaslikuchun, buniboshqachahamamalgaoshirishmumkin.

6-qadam. TextBox3 va TextBox4 nishrifo‘lchamlarini 12, TextBox1 va TextBox2 larboshqakattaroqo‘lchamnio‘rnatamiz. Chunkidasturishgatushgadaasosiyko‘rinadigannmatnlarko‘rinmaydigan matnlargako‘chiriladivaundankomponentagako‘chiriladi.

7-qadam. Zarurriyo‘zgaruvchilarnianiqlashmiz.

```
private: Bitmap^ bmp_for_draw;  
private: String^ full_name_of_image;
```

8-qadam. Oynaning Form1\_Load hodisasidavomidanquyidagialgoritmnijoylashtiramiz.

```
Text = "Matnni rasmga joylashtirish";  
Font = gcnew System::Drawing::Font("Times New Roman", 32, FontStyle::Bold);  
this->pictureBox1->SizeMode =  
PictureBoxSizeMode::StretchImage;
```

9-qadam. Komponentagakeraklirasmniyuklashuchun button4\_Clickhodisasigaquyidagialgoritmniyoziyadi

```
OpenFileDialog^ open_dialog = gcnew OpenFileDialog();  
open_dialog->Filter = "Image
```

```

Files(*.BMP;*.JPG;*.GIF;*.PNG)|*.BMP;*.JPG;*.GIF;*.PNG|All files (*.*)|*.*";
if (open_dialog->ShowDialog() ==
System::Windows::Forms::DialogResult::OK)      {
    try{
        full_name_of_image = open_dialog->FileName;
        bmp_for_draw = gcnw Bitmap(open_dialog->FileName);
        pictureBox1->Image = bmp_for_draw;
        pictureBox1->Invalidate();
    }catch(Exception^ e) {
        System::Windows::Forms::DialogResult result =
MessageBox::Show("Fayl xato tanlandi"+e->ToString(),"Warning",MessageBoxButtons::OK,
MessageBoxIcon::Error);
    } }

```

10-qadam. Komponentadagirasmnisaqlashuchun  
button3\_Clickhodisasigaquyidagialgoritmniyoziladi

```

if (pictureBox1->Image != nullptr){
    String^ format = full_name_of_image->Substring(full_name_of_image->Length - 4, 4);
    SaveFileDialog^ savedialog = gcnw SaveFileDialog();
    savedialog->OverwritePrompt = true; // Agar shu nomli fayl bo'lsa
    savedialog->CheckPathExists = true; // Agar noto'g'ri nom kiritilsa
    savedialog->ShowHelp = true;
    savedialog->Filter = "Image Files(*.BMP)|*.BMP|Image Files(*.JPG)|*.JPG|Image Files(*.GIF)|*.GIF|Image Files(*.PNG)|*.PNG|All files (*.*)|*.*";
    if (savedialog->ShowDialog() ==
System::Windows::Forms::DialogResult::OK){
        try {
            Bitmap^ MBB = gcnw Bitmap(pictureBox1->Image);
            MBB->Save(savedialog->FileName,
System::Drawing::Imaging::ImageFormat::Jpeg);
        } catch(Exception^ e){
            MessageBox::Show("Rasmni saqlashda xatolik",

```

```
"FATAL ERROR", MessageBoxButtons::OK,  
MessageBoxIcon::Error);  
} } }
```

11-qadam. Komponentadagirasnitozalashuchun  
button1\_Clickhodisasigaquyidagialgoritmniyoziladi

```
Bitmap^ BM = gcnew Bitmap(pictureBox1->Image);  
Graphics^ C =  
System::Drawing::Graphics::FromImage(BM);  
//Graphics C = CreateGraphics();  
C->Clear(pictureBox1->BackColor);  
pictureBox1->Image = BM;
```

12-qadam. Komponentagamatnlarnijoylashtirishuchunuchun  
button2\_Clickhodisasigaquyidagialgoritmniyoziladi

```
System::Drawing::Font^ FontTemp = gcnew  
System::Drawing::Font("Times New Roman", 32,  
FontStyle::Bold);  
textBox4->Text = textBox1->Text;  
textBox3->Text = textBox2->Text;  
SaveFileDialog^ savedialog = gcnew SaveFileDialog();  
String^ Text = String::Format("{0}", textBox4->Text);  
String^ Txt = String::Format("{0}", textBox3->Text);  
Brush^ brsh = gcnew SolidBrush(Color::White);  
Bitmap^ MBB = gcnew Bitmap(pictureBox1->Image);  
Graphics^ G =  
System::Drawing::Graphics::FromImage(MBB);  
Graphics^ Q = G;  
G->TextRenderingHint =  
System::Drawing::Text::TextRenderingHint::AntiAlias;  
G->DrawString(Text, FontTemp, brsh, 75, 2);  
Q->TextRenderingHint =  
System::Drawing::Text::TextRenderingHint::AntiAlias;  
Q->DrawString(Txt, FontTemp, brsh, 5, 225);  
pictureBox1->Image = MBB;
```

Dasturniishgatushiramizvaquyidaginatijaniolamiz



12.8-rasm. Matnlarni rasmga joylashtirish.

## Graphics

sinfi usulari asosida tasvirlarni qurishning turli usullari qarab chiqdik, bu amallarni yordamida rasmlarni tahrirlash, ishlov berish, matnlarni joylashtirish kabi ixtiyoriy amallarni bajarish mumkin.

**Chart komponenta xususiyat va hodisalari.** Bu komponenta Data tab bo‘limiga joylashgan bo‘lib, asosan ma’lumotlarni infografiklarni yaratish uchun ishlatiladi. Infografika uchun ma’lumotlarni to‘plam kerak.

Bu komponentaning xususiyat va hodisalarini boshqa komponentalarni ikki bo‘lim, xuddi o‘shalar kabi bo‘limlarga bo‘lingan. Ularning maxsuslarini keltirib o‘tamiz.

1. **BorderSkin** xususiyatlar gruppasi bo‘lib, unda komponentaning yangi niqobga solish mumkin. Niqob deganda, uning yangi ko‘rinishini o‘rnatish mumkin. Rang (color), rasm (image), stil (style), kengligi (width) kabi xususiyatlar mavjud. Bu xususiyatlarni o‘rnatish muammokeltirib chiqarmaydi. Oldingi o‘rnatilganlaringizdabunday xususiyatlardan foydalanmasiz. Shuningdek, asosiy niqob SkinStyle bo‘lib, komponentaning asosiy ko‘rinishini o‘zgartirish uchun xizmat qiladi. Uning mos qiymatlarro‘yxati mavjud, shundan keraklisini tanlab olish mumkin.

2. **Palette** (palitra) — xususiyat yordamida komponentaning infografika ko‘rsatadigan shaklini tanlash mumkin. Uning mos qiymatlarro‘yxati mavjud.



3. PaletteCustomColors — bunda ham komponentaning infografikaniko‘rsatadigan shaklidanlash mumkin. Ammo foydalanuvchi o‘zining rangidanlashimim mumkin. Bundamaxsus muloqot oynasiasosidapalitra gaturliranglarni qo‘shish mumkin.

4. Annotations — bunda komponentaga izohlarni yozish mumkin. Buning o‘zining xususiyatlarimaxsus muloqot oynasiyordamida o‘rnatiladi.

5. ChartAreas — bu xususiyat orqali komponentaga bir nechta chart infografika joylashtirish mumkin. Buning ham o‘zi maxsusiyatlarini mos muloqot oynasibilano‘rnatish lozim.

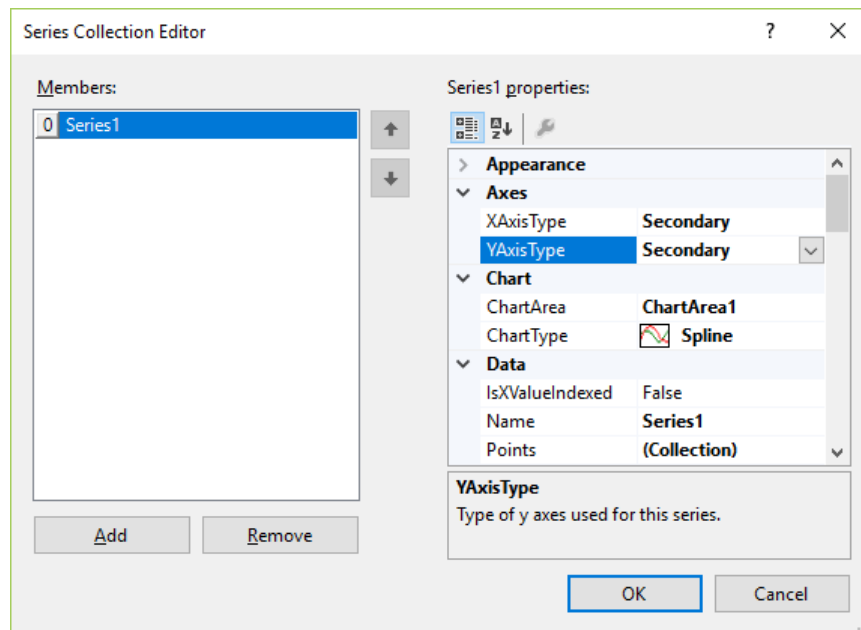
6. Legends — infografika gaketirilgan qiymatlarining joylashish maydoni. Buni ham maxsus muloqot oynasiasosidatahrir qilish mumkin.

7. Series — bu xususiyat asosiy bo‘lib, infografikaning qiymatlarini belgilovchi, har bir qiymat tegishlilikini bildiradi. Buning uchun maxsus muloqot oynasimavjud. Buning o‘ziga liqroq to‘xtalibo‘tamiz.

8. Titles — komponentalarga joylashtirilgan infografikalarga sarlavha qo‘yish uchun ishlatiladi.

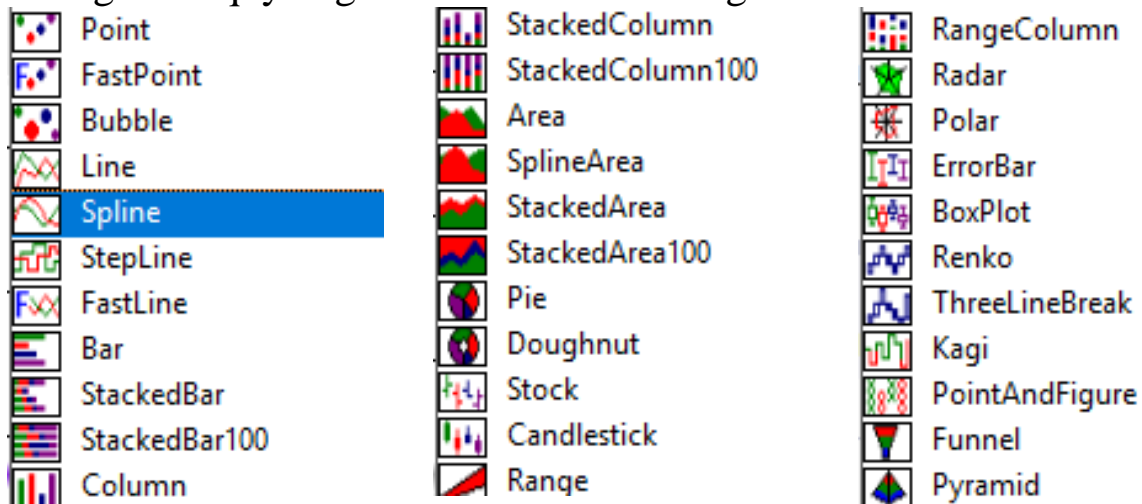
Uning maxsus muloqot oynasiorqali ishlov berish orqali shirnatish mumkin.

Series — bu komponentaga qiymatlarni qo‘shish va unitasvirlash uchun xizmat qiladi. Komponentaning ichiga joylashtirilgan sohanitahrirlash uchun ishlatiladi. Unga bosganda quyidagi muloqot oynasichiqadi.



12.9-rasm. Tahrirlashoynasi.

Buoyndayangiqiyatlar maydoni qo‘shish uchun [Add] tugmasi va unio‘chirish uchun [Remove] tugmasi ishlatiladi. Infografikaning asosiy xususiyatlarini boshqarish uchun o‘ng tomondagi xususiyatlar panelidan foydalanish mumkin. Unda infografika ni tahrirlash uchun zarur bo‘lgan barcha xususiyatlar bor. Ulardan biri bu ChartType bo‘lib, infografikaning turlarini belgilash uchun xizmat qiladi. Uning turlari quyidagi 12.10-rasmda keltirilgan.



12.10-rasm. Infografikaning turlari

Shuningdek, infografikaning ma‘lumotlari, yozuvlari, qiymatlari, maydoni, qiymat turlari, chegaralar bilan ishlash xususiyatlarini ham mavjud. Odatdagi xususiyatlar ni dasturlash orqali dastur fragmentlarida o‘rnatish foydalanish dastur chiga qulay hisoblanadi. Ammo vizual dasturlashning imkoniyatidan foydalanish uchun buni hamishl atishni o‘rganish lozim.

Funksiyalarnigrafiklariniqurish.

Infografikakomponentasigamosravishdafunksiyalarnigrafikginichizishusullariko‘ribchiqamiz.

Matematikfunksiyalarnigrafiklarinichizishuchunavvalshufunksiyalarnibirsingayaratibolamiz.

```
private value class MyFunction
{
  private:
    double _value;
  public:
    double getValuePow(double x) {
      return Math::Pow(x, 2);
    }
    double  getValueX4(double x){
      return x*x*x*x;
    }
    double  getValuekxa(double x, int k, int a){
      return k*x+a;
    }
};
```

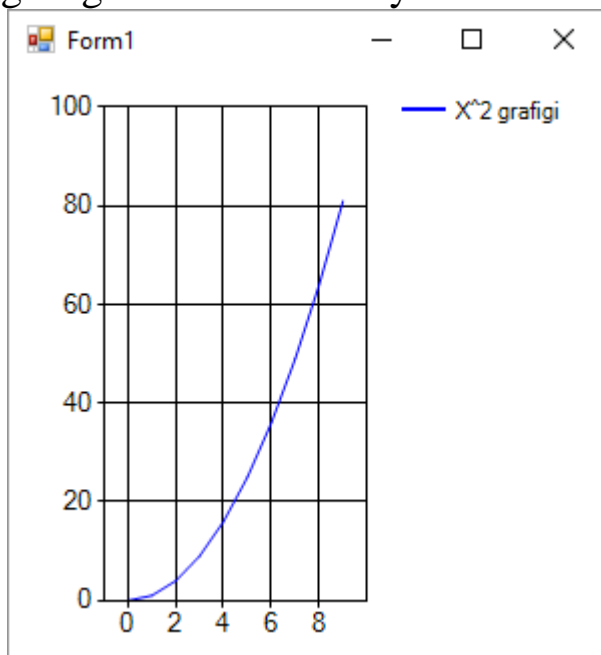
Buko‘rinishdamatematikaningbarchafunksiyalariniyaratibolishyo kito‘g‘ridanto‘g‘rifoydalanishmumkin.

Chart komponentasigagrafiknichizishuchun Form1\_Load hodisasigaquyidagichaalgoritmniyozamiz.

```
chart1->Series->Clear();
Series^ series1 = gcnew Series(L"X^2 grafigi");
// rangni tanlash
series1->Color = Color::Blue;
series1->IsVisibleInLegend = true;
series1->IsXValueIndexed = true;
// infografikani turini tanlash
series1->ChartType = SeriesChartType::Line;
// qiymatlar qatlamini qo‘shish
chart1->Series->Add(series1);
// qiymatlarni
MyFunction^ func = gcnew MyFunction();
for (double i = 0; i < 10; i++) {
  series1->Points->AddXY(i, func-
```

```
>getValuePow(i));
}
```

Agardasturniishlatsak,  
 bir  $x^2$  funksiyaning grafini chizishimkoniyatini beradi.



12.11-rasm.  $x^2$  funksiyaning grafigi.

Birvaqtningo‘zidabirnechta funksiyaning grafiklarini chizish uchun yuqorida aniqlangan sinfdan foydalanib, Form1\_Load  
 hodisasiga quyidagicha algoritmni yozamiz.

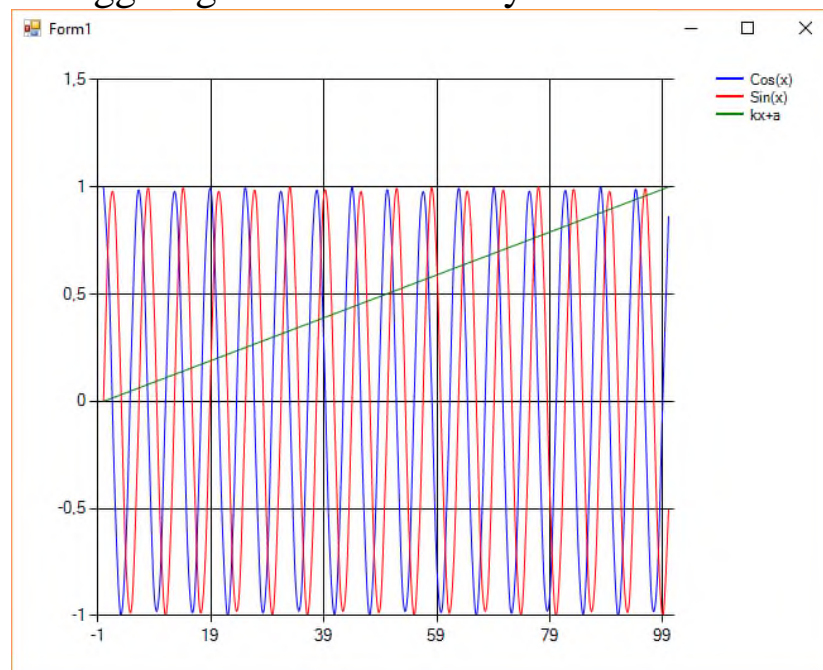
```
chart1->Series->Clear();
Series^ series1 = gcnw Series(L"Cos(x)");
Series^ series2 = gcnw Series(L"Sin(x)");
Series^ series3 = gcnw Series(L"kx+a");
// rangni tanlash
series1->Color = Color::Blue;
series2->Color = Color::Red;
series3->Color = Color::Green;
// infografikani turini tanlash
series1->ChartType = SeriesChartType::Spline;
series2->ChartType = SeriesChartType::Spline;
series3->ChartType = SeriesChartType::Spline;
// qiymatlar qatlamini qo‘shish
chart1->Series->Add(series1);
chart1->Series->Add(series2);
chart1->Series->Add(series3);
// qiymatlarni
```

```

MyFunction^ func = gcnw MyFunction();
for (double i = 0; i <= 100; i++) {
    series3->Points->AddXY(i, func-
>getValueKxa(i*0.01,1,0));
    series1->Points->AddXY(i, Math::Cos(i));
    series2->Points->AddXY(i, Math::Sin(i));
}

```

Agardasturniishlatsak,  
 birfunksiyalarning grafiginichizishimkoniyatiniberadi.



12.12-rasm. Funktsiyalarning grafigi.

Gistogrammagrafiqlarinichizishuchunmassivlardanyokiixtiyoriyto  
 ‘plamlardanfoydalanishmumkin. Buninguchun 3 tamassivolamiz,  
 Form1\_Load hodisasigaquyidagichaalgoritmniozamiz

```

chart1->Series->Clear();
    Title^ title = gcnw Title("Sotilgan texnikalar
soni ");
    chart1->Titles->Add(title);
    Series^ series1 = gcnw Series(L"Telefon");
    Series^ series2 = gcnw Series(L"Kompyuter
texnikasi");
    Series^ series3 = gcnw Series(L"Avtomashina");
    // rangni tanlash
    series1->Color = Color::Blue;
    series2->Color = Color::Red;
    series3->Color = Color::Green;

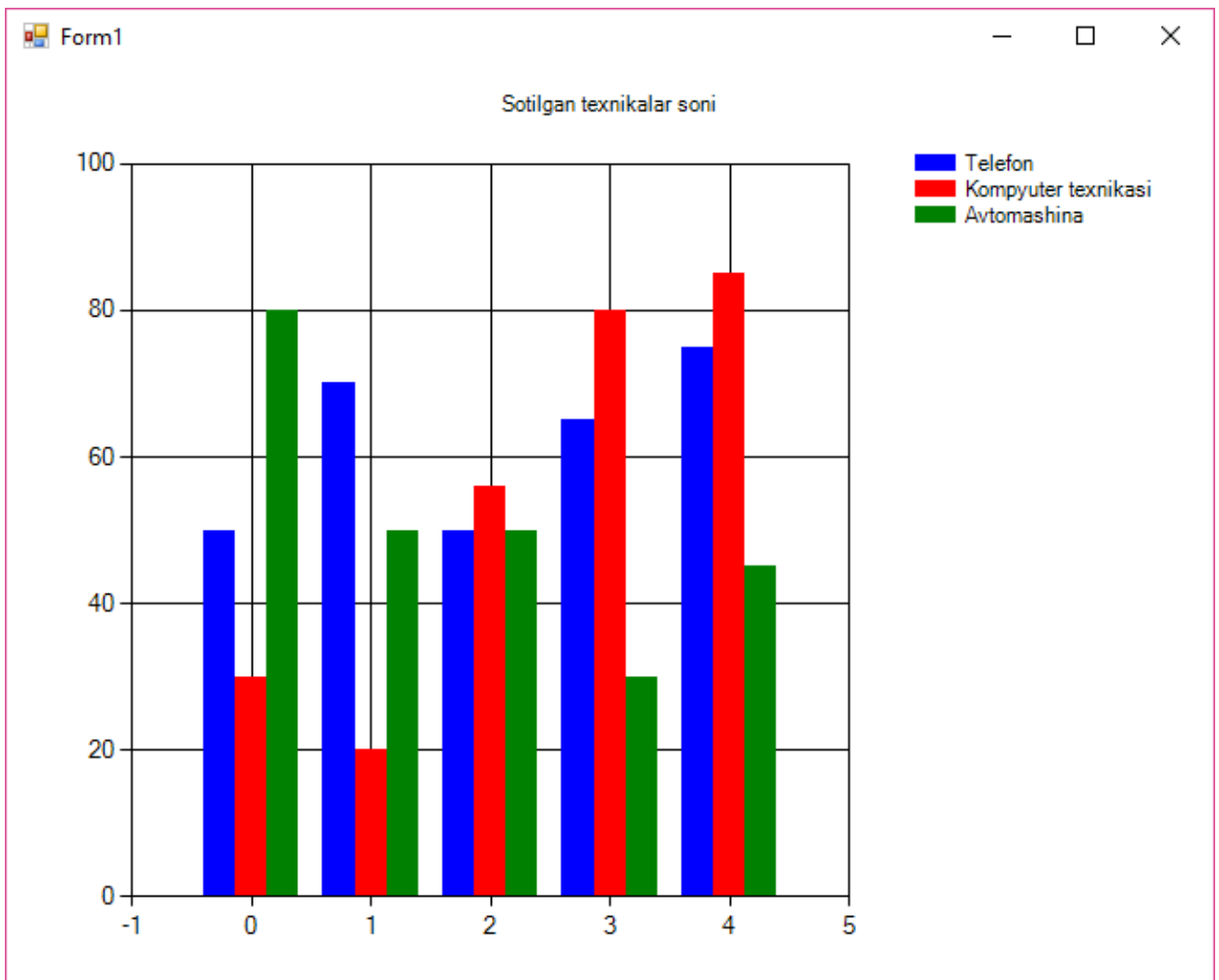
```

```

// infografikani turini tanlash
series1->ChartType = SeriesChartType::Column;
series2->ChartType = SeriesChartType::Column;
series3->ChartType = SeriesChartType::Column;
// qiymatlar qatlamini qo'shish
chart1->Series->Add(series1);
chart1->Series->Add(series2);
chart1->Series->Add(series3);
// qiymatlarni
array<int>^ arr1 = {30,20,56,80,85};
array<int>^ arr2 = {50,70,50,65,75};
array<int>^ arr3 = {80,50,50,30,45};
for (int i = 0; i < arr1->Length; i++) {
    series3->Points->AddXY(i, arr3[i]);
    series1->Points->AddXY(i, arr2->GetValue(i));
    series2->Points->AddXY(i, arr1->GetValue(i));
}

```

Agardasturni ishlatsak,  
Gistogrammagrafi ginichizishimkoniyatiniberadi.



12.13-rasm. Gistogrammagrafiklarini chizish

Grafiklarini alohida soha shachizish uchun massivlardan yoki ixtiyoriy to'plamlardan foydalanish mumkin. Buning uchun 3 ta massiv olamiz, Form1\_Load hodisasiga quyidagicha algoritmni yozamiz

```

chart1->Series->Clear();
ChartArea^ chartArea1 = gcnew ChartArea();
ChartArea^ chartArea2 = gcnew ChartArea();
ChartArea^ chartArea3 = gcnew ChartArea();
chartArea1->Name = "1";
chartArea2->Name = "2";
chartArea3->Name = "3";
this->chart1->ChartAreas->Add(chartArea1);
this->chart1->ChartAreas->Add(chartArea2);
this->chart1->ChartAreas->Add(chartArea3);
Title^ title = gcnew Title("Sotilgan texnikalar
soni ");
chart1->Titles->Add(title);
Series^ series1 = gcnew Series(L"Telefon");

```

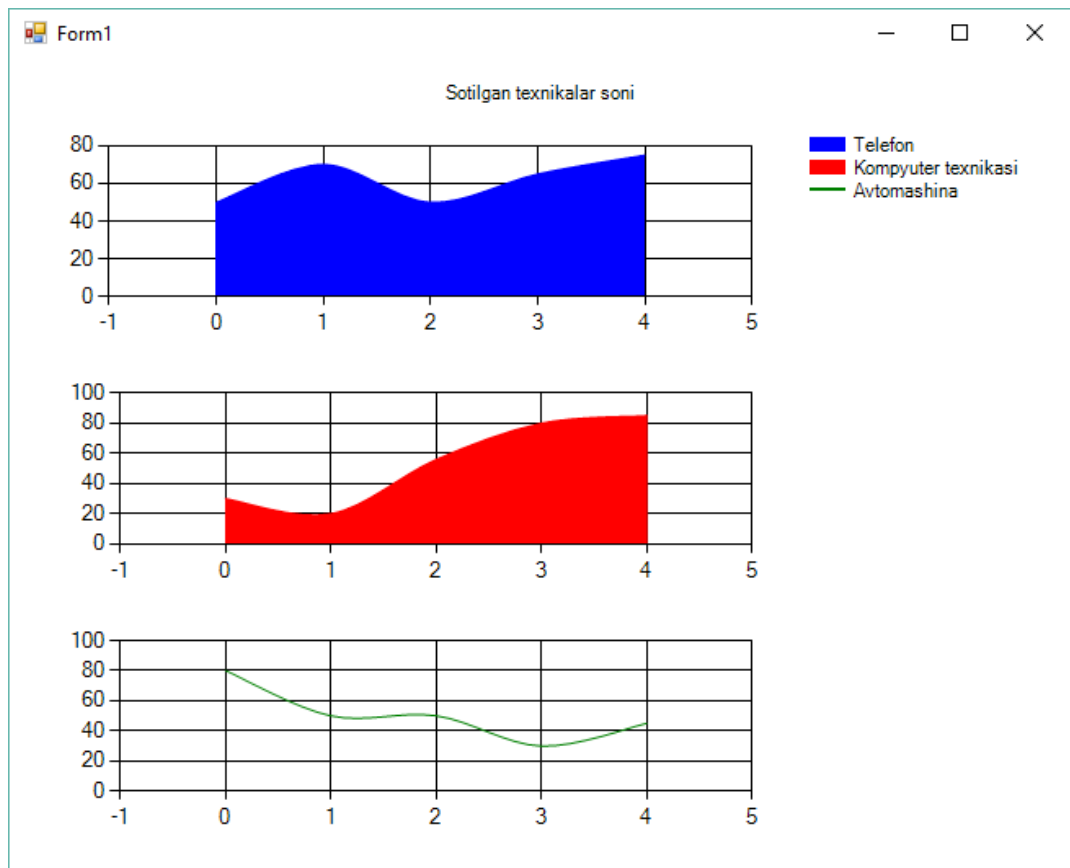
```

Series^ series2 = gcnew Series(L"Kompyuter
texnikasi");
Series^ series3 = gcnew Series(L"Avtomashina");
// rangni tanlash
series1->Color = Color::Blue;
series2->Color = Color::Red;
series3->Color = Color::Green;
// infografikani turini tanlash
series1->ChartType = SeriesChartType::SplineArea;
series2->ChartType =
SeriesChartType::SplineRange;
series3->ChartType = SeriesChartType::Spline;
series1->ChartArea = chartArea1->Name;
series2->ChartArea = chartArea2->Name;
series3->ChartArea = chartArea3->Name;
// qiymatlar qatlamini qo'shish
chart1->Series->Add(series1);
chart1->Series->Add(series2);
chart1->Series->Add(series3);
// qiymatlarni
array<int>^ arr1 = {30,20,56,80,85};
array<int>^ arr2 = {50,70,50,65,75};
array<int>^ arr3 = {80,50,50,30,45};
for (int i = 0; i < arr1->Length; i++) {
    series3->Points->AddXY( i, arr3[i]);
    series1->Points->AddXY( i, arr2-
>GetValue(i));
    series2->Points->AddXY( i, arr1-
>GetValue(i));
}

```

Dasturfragmentini tahlil qilish orqali yangi infografika sohalarni qo'shi-  
sh va ularni qiymatlar o'plam bilan bog'lashni ko'rish imkonligi mavjud.  
Agar dasturni ishlatib, Gistogramma grafigini chizish imkoniyatini beradi.





12.14-rasm. Bitta Chart gabirnechtasohanijoylashtirish.

### Chart

komponentasiyordamidafoydlanuvchiningixtiyoriyma'lumotlariniinfografikasiniyaratishmumkin.

Bujudakattaimkoniyatlikomponentahisoblanadi.

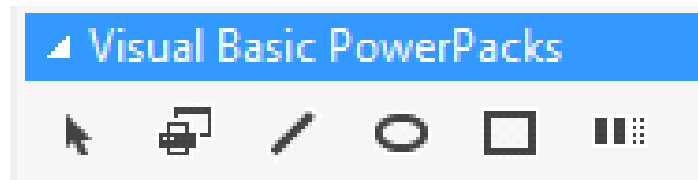
Bunixammasinio'rganishuchunko'proqamaliyotqilishlozim.

Yuqoridauningba'ziimkoniyatlarinikeltirganholdakomponentaningxususiyatlarivahodisalaribo'yichanazariyvaamaliyyma'lumotlarnikeltirdik.

**Visual Basic Power Packs**  
**komponentalarivaularniishlatish.**Bukomponentalarguruhialohidabo'lib, ularni Microsoft Visual Basic Power Packs debatashadi. Buniyangivarintlarinihaminternetdanolishvao'rnatishmumkin.






Asosiyvazifasiformanibezashuchunishlatiladi, shuninguchunbarchakomponentalargaborbo'lganhodisalarmavjud.

Bu Microsoft Visual Basic Power Packs nomlarfazosidajoylashganbo'lib, Visual Basic Power Packs elementlaruchunsinflarimavjud. Visual Basic Power to'plamlarelementlariqo'shimcha Windows formalarielementlariuchunmo'ljallangan. Ulardastlabbepulplugin-smayliksifatidaishlatilganvaendi Visual Studio tarkibigakiritilgan.



12.15-rasm. Visual Basic Power Packs komponentalari.

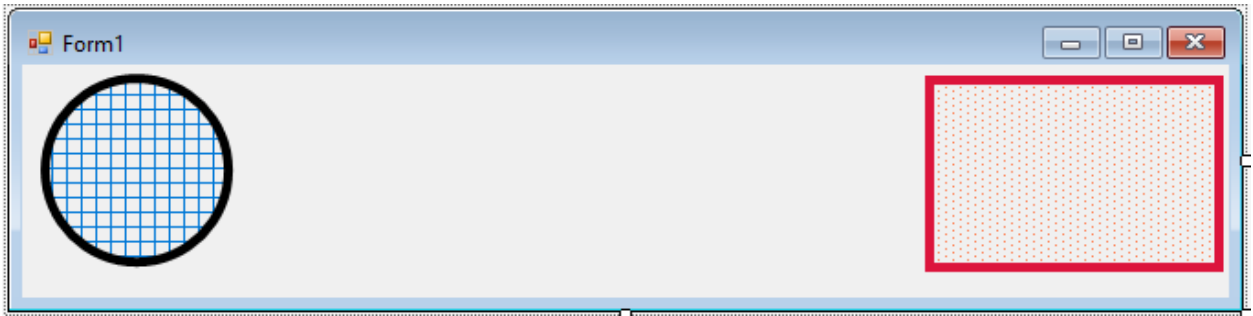
12.3-jadval. Visual Basic Power Packs komponentalari

Rasmi	Komponentanomi	vazifasi
	PrintForm	Formanichopqilishgaruxsatberishniboshqaradi
	LineShape	Gorozantal, vertikal, diognalchiziq larniboshqarishnitaʻminlaydi
	OvalShape	Ovalkoʻrinishdagiobʻektlarniboshqarishnitaʻminlaydi
	RectangleShape	Toʻrtburchakkoʻrinishdagiobʻektlarniboshqarishnitaʻminlaydi
	DataRepeater	Talabasosidaformatlanganmaʼlumotlarnikoʻrsatadi

OvalShape komponentasiningbaʼzixususiyatlarivahodisalari

1. BackColor – komponentaningorqarangini oʻrnatadivaqiyatlariranglartoʻplami.
2. BackStyle - komponentaningorqastilini oʻrnatadivaqiyatlari 2 tastatikberilgan.
3. BorderColor - komponentaningchegararangini oʻrnatadivaqiyatlariranglartoʻplami.
4. BorderStyle - komponentaningchegarastilini oʻrnatadivaqiyatlari 6 tastatikberilgan.
5. BorderWight - komponentaningchegararangqalinligini oʻrnatadivaqiyatlarisonlar.
6. FillColor - komponentaningaktivboʻlgandagirangni oʻrnatadivaqiyatlariranglar.
7. Location – komponentaningjoylashuvoʻrnini oʻrnatish..
8. SelectionColor - komponentaningtanlanganboʻlgandagirangni oʻrnatadivaqiyatlariranglar.

Birloyihayaratingvaunda 12.16-rasmkabiformanitayyorlang.



12.16-rasm. Loyihako‘rinishi.

Yaratilgan `ovalShape1` ni formabo‘yicha xarakterlanish nima masalasini ko‘ramiz. Sizing formadagi 2 ob‘ektni yaratishda ularning xususiyatlarini bilgan bo‘lgansiz.

Buning uchun loyihaga bir `Timer1` ob‘ektidan joylashtiramiz. Tez bajarilishi uchun uning interval xususiyatiga `1` qiymatni kiritamiz. `Timer1` ob‘ektining ustiga `chqonchani` kikkimartabosib, `timer1_Tick` hodisasiga quyidagicha algoritmni kiritamiz.

```

if(ovalpos <= (this->Size.Width - ovalShape1-
>Size.Width)){
    ovalpos += 5;
    ovalShape1->Location = Point(ovalpos,ovalShape1-
>Location.Y);
}
}

```

Bunda `ovalpos` qiymati `0` ga teng bo‘lgan butun son tipidagi o‘zgaruvchi. `Location` – bu ob‘ektning joylashini belgilaydigan juft qiymatga ega `Point` qiymatni qabul qiladi.

Shuningdek, `Form1_Load` hodisasida quyidagi algoritmni yozamiz.

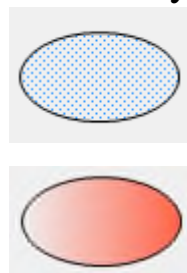
```





ovalpos = 0;
timer1->Start();

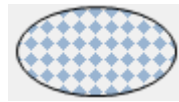
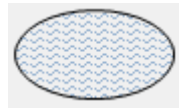
```

Agarda sturni ishga tushirsangiz `ovalShape1` ob‘ektining harakatini ko‘rasiz.

Shakllarning `FillColor`, `FillGradientColor`, `FillGradientStyle`, `FillStyle` xususiyatlaridan foydalanib, turli xil tugmalarni ham yaratish mumkin.



<code>FillColor</code>	 <code>MenuHighlight</code>
<code>FillGradientColor</code>	 <code>Maroon</code>
<code>FillGradientStyle</code>	<code>ForwardDiagonal</code>
<code>FillStyle</code>	<code>Percent20</code>
<code>FillColor</code>	 <code>Control</code>
<code>FillGradientColor</code>	 <code>Tomato</code>
<code>FillGradientStyle</code>	<code>Horizontal</code>
<code>FillStyle</code>	<code>Solid</code>



FillColor		ActiveCaption
FillGradientColor		White
FillGradientStyle		Horizontal
FillStyle		Wave
FillColor		ActiveCaption
FillGradientColor		OrangeRed
FillGradientStyle		Horizontal
FillStyle		SolidDiamond

Berilgan button tugmanibezashmasalasini qaraymiz. Buning uchun forma gabit tuga majoyla shtiramiz va uning xususiyatlarini quyidagicha aniqlaymiz.

```
using namespace System::Drawing::Drawing2D;
using namespace System::Drawing::Text;
// ...
this->button1->Location = System::Drawing::Point(115,
438);
this->button1->Name = L"button1";
this->button1->Size = System::Drawing::Size(121, 49);
this->button1->TabIndex = 0;
this->button1->Text = L"button1";
this->button1->UseVisualStyleBackColor = true;
```

Tugmaning button1\_Paint hodisasiga o'tib, quyidagi algoritimni yozamiz.

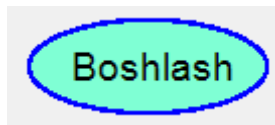
```
Pen^ pen = gcnew Pen(Color::Blue, 10);
Brush^ brush = gcnew SolidBrush(Color::FromKnownColor
(KnownColor::Control));
SolidBrush ^ brushinside = gcnew SolidBrush(Color::
Aquamarine);
Graphics ^ gr = e->Graphics;
gr->FillRectangle(brush, 0, 0, button1-
>Width, button1->Height);
gr->FillEllipse(brushinside, 0, 0, button1->Width,
button1->Height);
gr->DrawEllipse(pen, 0, 0, button1->Width, button1-
>Height);
GraphicsPath^ path = gcnew GraphicsPath();
button1->Region = gcnew
System::Drawing::Region(path);
String^ text = "Boshlash";
System::Drawing::Font^ drawfont = gcnew
```

```

System::Drawing::Font("Arial", 14);
SolidBrush^ exbrush = gcnew SolidBrush(Color::Black);
RectangleF rect = RectangleF(button1->Width/2-
40,button1->Height/2-10,button1->Width, button1-
>Height);
gr->DrawString(text, drawfont, exbrush,rect);

```

Bundachizishuchunqalam - Pen , mo'yqalam - Brush, to'liqbo'yashuchunmo'yqalam - SolidBrush yaratibolamiz. Keralishakllarnichizishob'ektininitugmaningo'lchamigamoslabtayyorlay miz. Tugmaningustigamatnjoylagtirishuchunyanabirob'ektyaratamiz, ungamatn, matnrangi, formativaegallashsohasiniko'rsatib, DrawString bilantugmaningustigachizamiz.



12.17-rasm. Tugmanibezashgamiisol.

Yangiyaratilgantugmachalarorqalishakllarnioqimorqaliboshqarish masalasiniqaraymiz. BuninguchunLoyihaformasiga, 1 ta PictureBox, 2 ta button tugmaniyuqoridagidekqilibjoylashtiramiz. PictureBox ning BackColor xususiyatigaoqrangnio'rnatimiz. Shaklningxarakatlantirishuchununingtayanchnuchtasiniko'chirib, oldingisioqbilanbo'yaladivakeyingisihosilqilinadi, shundayqilibiteratsiyadavometaveradi.

Ikkitaberilgantugmalarniyuqoridagidekqilib, moslabolamiz.

Oqimniboshqarishuchunavvalbirtugmaniyuqoridagidekqilib, yaratibolamiz.

Butugmaniyuqoridagidekqilib, yaratibolamiz. PictureBox bo'yabharakatlanishvaoldinkeltirilganiteratsiyaniamalgaoshirishuchunk erak. Funksiyaningalgoritmiquyidagicha:

```

public: void Go(){
    int R = 30;
    // button2->BeginInvoke(gcnew
    InvokeDelegateSetEnabled(this,
    &Form1::setEnabledToButton), false);
    Bitmap ^ image = gcnew Bitmap(pictureBox1->Width,
    pictureBox1->Height);
    Graphics^ g = Graphics::FromImage(image);
    for (int x = 0; x <= pictureBox1->Width - 2 * R;
    x += 2){
        g->Clear(Color::White);
    }
}

```

```

        g->FillEllipse(Brushes::Aqua, x, 0, 2
* R, 2 * R);
        pictureBox1->BeginInvoke(gcnew
InvokeDelegateSetImage(this,
&Form1::setImageToPictureBox), image);
        Thread::Sleep(10);
    }
    for (int y = 0; y <= pictureBox1->Height - 2 * R;
y += 2){
        g->Clear(Color::White);
        g->FillEllipse(Brushes::Aqua,
pictureBox1->Width - 2 * R, y, 2 * R, 2 * R);
        pictureBox1->BeginInvoke(gcnew
InvokeDelegateSetImage(this,
&Form1::setImageToPictureBox), image);
        Thread::Sleep(10);
    }
    for (int x = pictureBox1->Width - 2 * R; x >= 0;
x -= 2){
        g->Clear(Color::White);
        g->FillEllipse(Brushes::Aqua, x,
pictureBox1->Height - 2 * R, 2 * R, 2 * R);
        pictureBox1->BeginInvoke(gcnew
InvokeDelegateSetImage(this,
&Form1::setImageToPictureBox), image);
        Thread::Sleep(10);
    }
    for (int y = pictureBox1->Height - 2 * R; y >= 0;
y -= 2)
    {
        g->Clear(Color::White);
        g->FillEllipse(Brushes::Aqua, 0, y, 2
* R, 2 * R);
        pictureBox1->BeginInvoke(gcnew
InvokeDelegateSetImage(this,
&Form1::setImageToPictureBox), image);
        Thread::Sleep(10);
    }
// button2->BeginInvoke(gcnew

```

```
InvokeDelegateSetEnabled(this,
&Form1::setEnabledToButton), true);
    }
```

Bu funksiyani ishlatish uchun qiymatni to'g'ri dan to'g'ri berib bo'lmay  
dishing uchun birdelgant (funksiyako'rsatkichsaqllovchi)  
va qiymatni funksiyalar yaratamiz.

```
delegate void InvokeDelegateSetImage(Bitmap^ image);
public: void setImageToPictureBox(Bitmap^ image){
    pictureBox1->Image =
image;
    }
```

Formaning ochiq joyiga oqim yaratib olamiz.

```
using namespace System::Threading;
// ...
private: Thread^ myThread;
private: Thread^ myThreadOne;
```

Birichitugmabosilgandamaz kuroqimlarni ishlatish uchun button1\_Click hodisasi ga quyidagi algoritmi yoziladi.

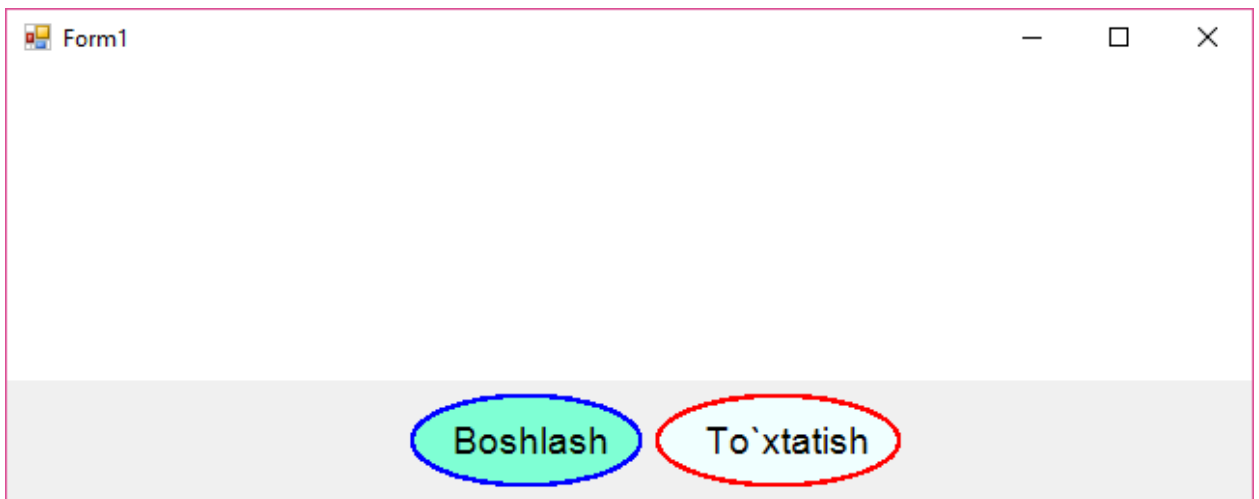
```
myThread = gcnew Thread(gcnew
ThreadStart(this, &Form1::Go));
myThread->Start();
myThreadOne = gcnew Thread(gcnew
ThreadStart(this, &Form1::GoOne));
myThreadOne->Start();
```

Oqimni saqlab turish uchun yaratilgan funksiyada  
Thread::Sleep(10); dan foydalanilgan.

Oqim ishlatishgan paytdan ishlaydi,  
ammo forma yopilganda uning ishini to'xtatish lozim. Shuning uchun oqimni  
to'xtatish uchun button2\_Click va Form1\_FormClosing  
hodisalariga quyidagi algoritmi kiritamiz.

```
try{
    myThread->Abort();
    myThreadOne->Abort();
}
catch(...){}
```

Loyihani ishlatish hirsangiz, quyidagi 12.18- rasm hosil bo'ladi.



### 12.18- rasm. Oqimlarasosidaxarakatniboshqarish

Dasturlashboshlashtugmasinibosgandaikkita paralel bo'lgan shakllari ning harakatini ko'rasiz. To'xtatish bosilganda ularni to'xtatishni ko'rasiz. Xarakat davom etgan vaqtda ham formani yopiqishni tugallash mumkin.

Umuman olganda grafik bilan ishlash uchun geometriyani yaxshi bilasht alab qilinadi. Visual C++ da grafik asinfi, uning usullari va turlari shakllarni chizish usullari, Chart va Shape sinflarining xususiyatlarini va usullarini ko'rib chiqdik. Bularni ilqizigaxosma'lumotlarda ishlatishni tavsiya qilamiz.

## NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.



1. Chiziqlar va shakllar chizish uchun maxsus qanday usullarni o'rganish lozim
2. DrawLine, DrawArc, DrawClosedCurve, DrawPolygon va DrawRectangle o'z ichiga oladigan sinflarni ayting.
3. Grafik asinfi xususiyatlarini sanab bering.
4. Grafik asinfi koordinatalarini uchun ishlatiladigan o'lchov birligini ayting.
5. GraphicsUnit qiymatlarini sanab bering.



6. -

butunchizmayuzasinitozalaydivabelgilanganfonrangibilanto‘ldiradiganus ulningnominiayting.

7. Bufunksiyaberilganparametrlarasosidatasvirnibelgilanganjoydaasl kattaligidanfoydalanibchizadi. Funksiyanominiayting.

8. Chiziqchizishniqandayamalgaoshiriladi.

9. Uchburchakchizishniqandayamalgaoshiriladi

10. Ellipischizishniqandayamalgaoshiriladi

11. Turlishakllarnibo‘yyashniqandayamalgaoshiriladi

12. Ko‘pburchaklarnichizishniqandayamalgaoshiriladi

13. Hodisalarorqalishaklchizishniqandayamalgaoshiriladi

14. Rasmlarnio‘zgartirishniqandayamalgaoshiriladi

15. Matnlarnitasvirkabijoylashtirishniqandayamalgaoshiriladi

16. Matnlirasmlarniyaratishniqandayamalgaoshiriladi

17. Chart komponentaxususiyativahodisalarahaqidagapiribbering.

18. BorderSkin nima.

19. Qaysixususiyatyordamidakkomponentaninginfografikaniko‘rsatad iganshaklinitanlashmumkin

20. Qaysixususiyatasosiybo‘lib, infografikaningqiymatlarinibelgilovchi, harbirqiymattegishliligini bildiradi

21. ChartType infografikaningturlarinibelgilashuchunxizmatqiladi. Uningqiymatlarinibilasizmi.

22. Birinfografikadanechttagachagistogrammajoylashtirishmumkin.

23. Chart

komponentasiyordamidafoydlanuvchiningixtiyoriyma‘lumotlariniinfografikasiniyaratishmumkinmi

24. Nimalardastlabbepulplugin-smayliksifatidaishlatilganvaendi Visual Studio tarkibigakiritilgan

25. Visual Basic Power Packs komponentalarinisanabbering.

26. Visual Basic Power Packs

komponentalariningasosiyvazifalarinimadaniborat.

27. Komponentaningchegararangqalinliginio‘rnatadivaqiymatlarison lardaniboratbo‘lganxususiyatnominiayting.

28. Shakllarning FillColor, FillGradientColor, FillGradientStyle, FillStyle xususiyatlaridanfoydalanib, niamqilishmumkin.

29. Tugmningko‘rinishinio‘zgartirishuchunnimaamallarbjario‘kera k.

30. Rasmlarniharakatlantirishdaoqimlardanqandayfoydalanamiz.



---

---

---

---

---

---

---

---

---

---

3. Intellektual turli ma'lumotlarni kiritishni talab qiluvchi funksiyalarga afigniyaratish dastur tuzing.  
Loyihabo'yicha qisqacha asosiy fragmentlarni yozing.

---

---

---

---

---

---

---

---

---

---

---

---

4. Tugmalarni bezash orqali kichik kalkulator dasturini tuzing.  
Loyihabo'yicha qisqacha asosiy fragmentlarni yozing.

---

---

---

---

---


---

---

---

_____
<p>5. Barchaturdagimuloqotoynalarinio‘zichigaolgan, interaktivsavoljavobtiziminiyaratingvajavoblargayasamaharakatl anishvositalaridanfoydalanishdasturtuzing. Loyihabo‘yichaqsqachaasosiyfragmentlarniyozing.</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p>
<p>6. Barchavazifalarnio‘rganganingizbo‘yichaeslabko‘ring, nimalarnio‘rgandingizvabirtakrorlang.</p> <p>7. Barchabilimvako‘nikmalarasosidakichikloyihatayyorlang.</p>

### 3.5. OLE,MFC texnologiyalari. Ko‘poynalimuhitlar bilan ishlash.

 Visual C++ningoperatsiontizim, amaliydasturlaribilano‘zarobog‘lanishuchun MFC va OLE texnologiyalariningimkoniyativavariantlari, ularningo‘zarohamkorligi, ko‘poynalidasturlarniyaratishbo‘yichanazariybilimlarkeltirilgan.. Bilimlarnimustahkamlashuchun 15 tanazariysavolberilgan.

*✍***Kalitso‘zlar.**Operatsiontizim, amaliydastur, integratsiya, o‘zarobog‘lanish, MFC, OLE, SDI, MDI, ko‘poynalidastur..

*✔***Bilishshartbo‘lgantushunchalar.**Operatsiontizimvaamaliydasturl ardaishlash, sinfvasinfob’ekti, xususiyat, hodisa, forma, komponentadasturlashgaoiddastlabkitushunchalarhamdaC++tiliniqo‘llab quvvatlovchimuhitdaishlashnibilishlozim.

**Bilibolasiz.** Visual C++ muhitidasturlash, Visual C++ning MFC texnologiyasiimkoniyatlarivavariantlari, OLE texnologiyasiimkoniyatlarivavariantlari, ularningo‘zarobog‘lashuchunloyihayaratish, ko‘poyalidasturlarniyaratishuchun SDI, MDI konstruktorlariusularivaba‘zibirxususiyatlari, hodisalarior‘rganishingiz mumkin.

Reja

1.MFC texnologiyasi.

2.OLE texnologiyasi.

3.Visual C++ muhitidako‘poyalimuhitniyaratish.

**MFC texnologiyaci.** Microsoft Foundation Class paketi (MFC) Microsoft tomonidanishlabchiqilganbo‘lib, C++ kutubxonalariyordamida Microsoft Windows uchun GUI dasturlarishlabchiqishniengillashtirishuchunmo‘ljallangan. Undakutubxonalariningvasinflariningsonijudako‘p.

MFC texnologiyasi

MFC kutubxonasi, uningasosiyraqobatchisibo‘lgan Borland VCL kabi, GUI bilanishlashniosonlashtiradidasturasoslariniyaratish, ya‘nima’lumbirtartibasosidaavtomatikravishdayaratiladi.

Foydalanuvchiinterfeysivabutunlayuningamaliyvatexnikxizmatko‘r satishuchunmuntazamharakatvahodisalarbajariladi(masalan, oynaamallari,

elementlarvadasturo‘zgaruvchilarichkiob’ektlaro‘rtasidama’lumotlarniy uborish, vahokazo.). Dasturasoslariniishlabchiqqandanso‘ng, dasturchifaqatkodnimaxsussharakatvahodisalaratalabqilinadiganjoylargaki ritishikerak.

Bunday Framework yaxshibelgilangantuzilishgaegabo‘lishikerak, shuninguchun visual C++ uniyaratishvao‘zgartirishuchunyordamchikomponentalarniberadi.

Bundantashqari, MFC Windows API vazifalariturliob’ektgayo‘naltirilganqatlamlariniberadi, ularbilanishlashuchunbirozosondir.

Buqatlamko‘pajralmasob’ektlarniifodalaydi (Windows, vidjet, fayllar, instrumentlar, sinflar, komponentalarvaboshqalar.). Sinflarsifatidajoriyhodisalariniyopishvaxotirajratish/bo‘shatishkabimunt azamamallarnibajaradi.

Dasturdoirasigakodqo‘shish. MFC Framework uchundasturkodikiritishuchunikkiyo‘lbor.

Birinchisikutubxonadanmerosmexanizmidanfoydalanadi:

asosiy dastur doirasida tuzilmalar merosqilibolingansinflar sifatidan amoyon bo'ladi. Businflar dasturidama'lum nuqtalarda chaqiriladigan ko'pgina virtual funksiyalarni ta'minlaydi. Bu funksiyalarni yanada belgilab (ko'phollarda bazaviy sinflar vazifasini chaqirish kerak), dasturda o'z kodidashu nuqtalarda bajarishni qo'shish mumkin.

Ikkinchi usul o'ynahodisalarini qo'shish uchun ishlatiladi. Yordamchi instrument Windows — message maps (inglizcha: message map) bilan bog'liq sinflar qoliblarida [message ID — pointer to handler] juftlarini o'z ichiga olgan maxsus massivlarni yaratadi. Qo'shish yoki o'rnatish yordamchi instrument asosida tegishli xabar orqali o'zgarishlar qilinadi.

MFC ning yaratilish tarixi. MFC ning birinchi versiyasi 1992-yilda Microsoft kompaniyasining 16-bitli C/C++ kompilyatorining gettinchi versiyasi bilan birgacha qilingan. API funksiyalari yordamchi dailovalarni ishlab chiqarotganlar uchun MFC paketini yaratishni o'z ichiga olgan paket yaratildi.

1. Dasturlash jarayonini boshqarish. MFC ninge'tiborli xususiyatlaridan biri "AFX" prefiks bo'lib, ko'plab funksiyalar, makroslar nomlarida va "STDAFX" standart sarlavha sifatida ishlatiladi.

Rivojlanishning dastlabki bosqichida, keyinchalik MFC nomi "dastur" deb nomlangan. AFX doirasida renga yitmalari Microsoft Foundation sinflar nomini o'zgartirish uchun qaror (MFC) kodi AFX murojaatlarni o'zgartirish uchun juda kechqilingan.

2. C++ kompilyatori bilan ishlash uchun Borland tomonidan ishlab chiqilgan Object Windows Library (OWL) shu davrda joriy etilgan raqobatchi mahsulot edi. Oxir oqibat Borland rivojlanishdagi o'xtadiva MFC bilan ishlash uchun kutubxona, foydalanish uchun qisqam uddatli litsenziya sotib oldi. Lekin uning mahsulotlari MFC uchun o'liq qo'llab-quvvatlash amalga oshirilmadi.

3. MFC Microsoft markazida Microsoft .Net Framework tanlangan. Biroq, shunga qaramay, MFC hali ishlab chiqilganlar orasida mashhur.

Birinchi lardan bo'lib, Rossiyaning bcgsoft kompaniyasida sotib olingan. MFC yangi dizaynli interfeysni qo'llab-quvvatlash va takomillashtirilgan interfeys bilan bog'liq bir nechta boshqo'zgartirishlarni kiritgan va nazorat qilmoqda. Visual Studio 2008 tizimidan ba'zibir komponentalarni olib, businflar asosida MFC ajralmas qismi qilib yaratildi.

## 13.1-jadval. MFC mahsulotlarivavarintlari.

<b>Mahsulot nomi</b>	<b>MFC varianti</b>	<b>Joriy qilingan sana</b>
Microsoft C/C++ 7.0	MFC 1.0	1992
Visual C++ 1.0	MFC 2.0 (arxitekturasi yangilangan)	
Visual C++ 1.5	MFC 2.5 (ODBC i dragand-dropni qo'llabquvvatlovchi texnologiyajoriy qilingan)	
Visual C++ 1.52c	MFC 2.5 (MS Windows 3.x uchun yaratilgan oxirgi versiya)	
Visual C++ 2.0	MFC 3.0 (ko'p masalaliva Unicode ni qo'llabquvvatlash joriy qilindi)	
Visual C++ 2.1	MFC 3.1	
Visual C++ 2.2	MFC 3.2	
Visual C++ 4.0	MFC 4.0 (mfc40.dll - Windows 95 tarkibi g akiritilgan)	1995y. avgust
Visual C++ 4.1	MFC 4.1	
Visual C++ 4.2	MFC 4.2 (mfc42.dll - Windows 98 tarkibi g akiritilgan)	1998y. mart
eMbedded Visual C++ 3.0	MFC 4.2 (mfc42.dll)	
Visual C++ 5.0	MFC 4.21 (mfc42.dll), MFC 4.2 ning yangilanganligi	
Visual C++ 6.0	MFC 6.0 (mfc42.dll)	1998
eMbedded Visual C++ 4.0	MFC 6.0 (mfcce400.dll)	
Visual C++ .NET 2002 (Visual	MFC 7.0 (mfc70.dll), .NET 1.0	2002 y. fevral

C++ 7.0)		
Visual C++ .NET 2003 (Visual C++ 7.1)	MFC 7.1 (mfc71.dll), .NET 1.1	2003 y. aprel
Visual C++ 2005[2] (Visual C++ 8.0)	MFC 8.0 (mfc80.dll), .NET 2.0	2005 y. Mart
Visual C++ 2008[2] (Visual C++ 9.0)	MFC 9.0.21022 (mfc90.dll), .NET 3.5	2007 noyabr
Visual C++ 2008[2] with Feature Pack	MFC 9.0.30411 (mfc90.dll)	2008 y. Aprel
Visual C++ 2008[2] SP1	MFC 9.0.30729 (mfc90.dll)	2008y. avgust
Visual C++ 2008[2] Security Update (KB971092 )	MFC 9.0.30729.4148 (mfc90.dll)	2009y.Iyul
Visual C++ 2010	MFC 10.0.30319.1 (mfc100.dll), .NET 4.0	2010 y. aprel
Visual C++ 2010 SP1	MFC 10.0.40219.1 (mfc100.dll), .NET 4.0	2011y.mart
Visual C++ 2010 + MS11-025	MFC 10.0.30319.415 (mfc100.dll), .NET 4.0	2011 y. aprel
Visual C++ 2012 (Visual	MFC 11.0.50727.1 (mfc110.dll), .NET 4.5	2012 y. iyul



C++ 11.0)		
Visual C++ 2012 Update 1 (Visual C++ 11.0)	MFC 11.0.51106.1 (mfc110.dll), .NET 4.5	2012y. noyabr
Visual C++ 2012 Update 3 (Visual C++ 11.0)	MFC 11.0.60610.1 (mfc110.dll), .NET 4.5	2012y. dekabr
Visual C++ 2013 (Visual C++ 12.0)	MFC 12.0.21005.1 (mfc120.dll), .NET 4.5.1	2013
Visual C++ 2013 Update 2 (Visual C++ 12.0)	MFC 12.0.30501.0 (mfc120.dll), .NET 4.5.1	2014
Visual C++ 2015 (Visual C++ 14.0)	MFC 14.0.23026.0 (mfc140.dll), .NET 4.6	2015
Visual C++ 2015 Update 1 (Visual C++ 14.0)	MFC 14.0.23506.0 (mfc140.dll), .NET 4.6.1	2015 y. noyabr
Visual C++ 2015 Update 2 (Visual C++ 14.0)	MFC 14.0.23918.0 (mfc140.dll)	2016 y. mart
Visual C++ 2015	MFC 14.0.24210.0 (mfc140.dll)	2016y. iyun



multimediyakontentibilanishlashdaamalgaoshirishg'oyasikengqo'llaniladi. Biroq OLE texnologiyasi "kattamijoz" arxitekturasidan, ya'niortiqchahisoblashresurslarigaegabo'lgantarmoqkompyuteridanfoydalanadi.

Buamalgaoshirishdafaylturiyokidasturmijozmashinamavjudbo'lishikerak. Masalan, OLE Microsoft Excel jadvallaridanfoydalansa, Excel foydalanuvchimashinasigao'rnatilishikerak.

OLE	1.0	Microsoft	Windows
operatsiontiziminingoldingiversiyalaridaishlatiladigan			DDE
(dinamikma'lumotlaralmashinuvi)		texnologiyasiasosida	1990-
yildachi qarilgan.			DDE

texnologiyasiikkitaishlaydigandasturo'rtasidama'lumotlarniuzatishsoniv  
ausullaribilancheklanganbo'lsa-da, OLE  
ikkitahujjatorasidagifaolulanishlarustidaishlayoldivahattobirturdagihujjat  
niboshqaturdagihujjatgajoylashtirdi.

OLE  
serverlarvamijozlarvirtualfunksiyajadvallaryordamidatizimikutubxonalar  
bilano'zarobog'lanadi.

Bujadvallardatizimkutubxonasiserveryokimijozbilano'zaroaloqaqilishuc  
hunfoydalanishimumkinbo'lganfunksiyalargamosko'rsatgichlarmavjud.  
Kutubxona OLESRV.DLL (server) va OLECLI.DLL (mijozhaqida)  
dastlaboperatsiontizimitomonidantaqdimetiladigan  
WM\_DDE\_YeXECUTE xabaryordamidabir-  
biribilano'zaromoshuvikerak.

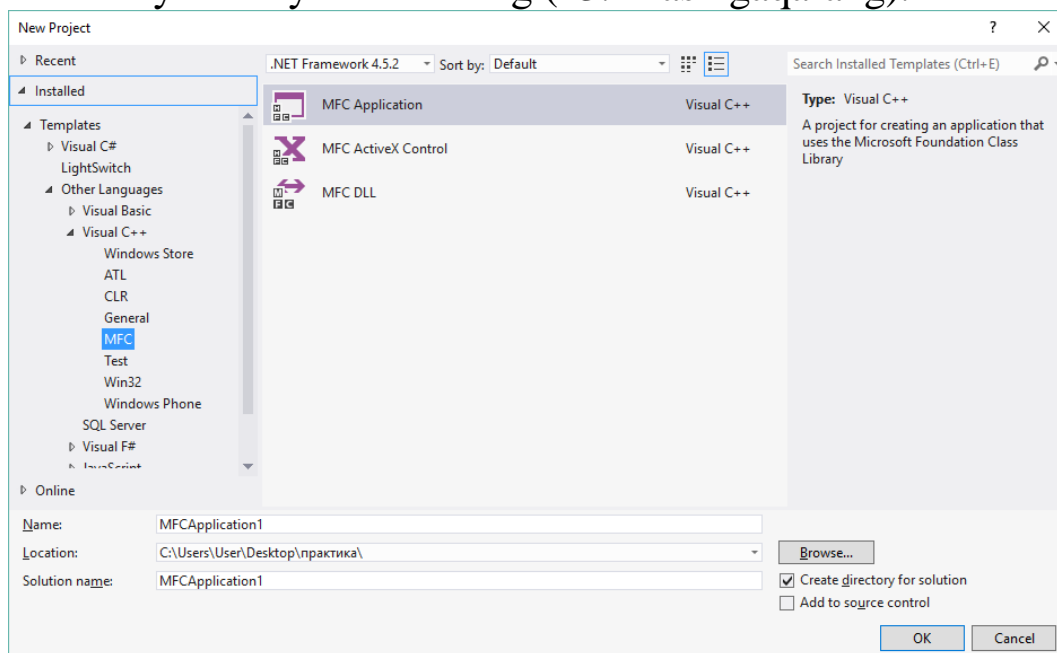
OLE 1.1 keyinchalikdasturiykomponentlarbilanishlashuchun COM  
(butlovchiob'ektmodeli) arxitekturasigaaylandi. Keyinchalik,  
DCOMsifatidama'lumbo'ldi.

Buferga OLE ob'ektijoylashtirilganda Windows formatlarida  
(bitmap yoki metafile kabi) saqlanadihamdao'zformatidasaqlanadi.  
Uformatniqo'llab-quvvatlash OLE dasturi  
clipboardgako'chirilganboshqahujjatningbirqisminijoylashvafoydalanuv  
chihujjatdaunisqaqlashimkoniniberadi.

Keyingievoluyusionqadam OLE 2.0 edi,  
buavvalgiversiyabilanbirxilmaqsvavazifalarnisaqlabqolganholda OLE  
2.0 VTBL yordamida COMarxitekturasigaqo'shilganbo'lgan.  
Yangixususiyatlari Drag-and-Drop avtomatlashtirish, IN-  
joyfaollashtirishvatuzilganlarnisaqlashtexnologiyalarinio'zichigaoladi.

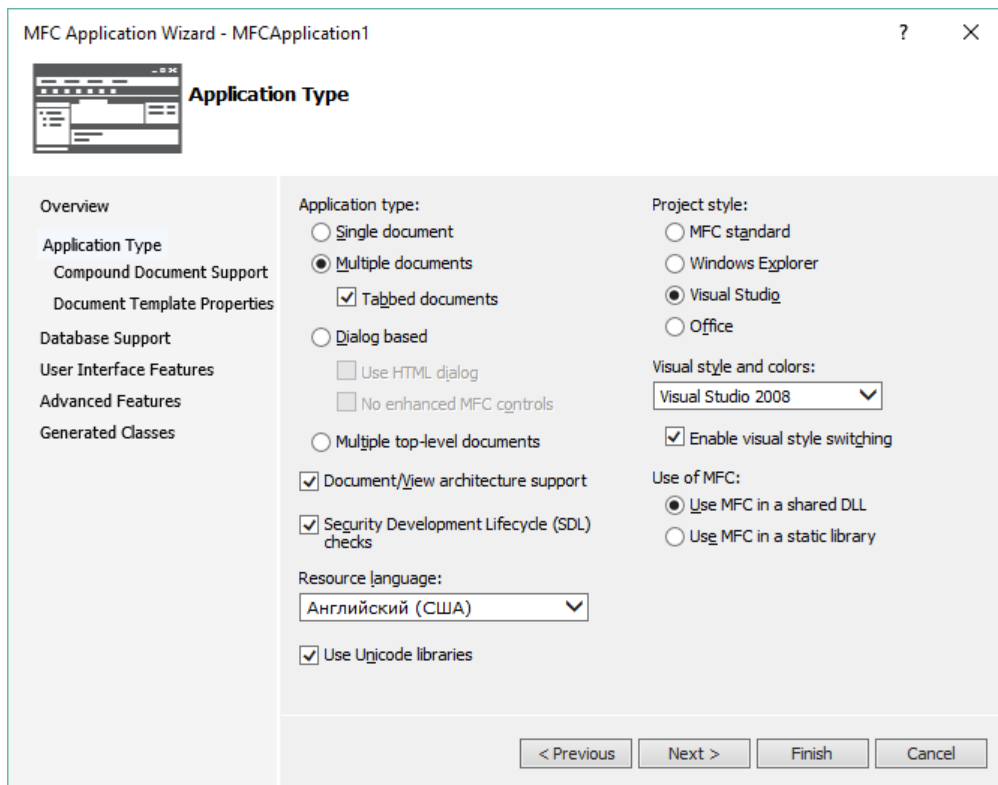
Visual C++dasturlarida OLE konteyneridanfoydalanish.

Visual C++ OLE moslamalarnifoydalanishmisolinita 'riflaydidebatalmish OLE konteynerb yordamidailovalar MFC platformasidaamalgaoshiriladi. OLE ob'ektlarengtez-tezmurakkabhujjatqismlarisifatidaishlatiladi. Hujjatlarga Microsoft Office ilovalari (Excel, Word, PowerPoint vaboshqalar) yordamidayaratilgan fayllarkiradi. Biroq, Microsoft Office hujjatlariningo'ziboshqailovalarda OLE ob'ektlarisifatidaishlatilishimumkin. Misolsifatida, Visual Studio yangi MFC dasturloyihasiniyaratibko'ring (13.1-rasmgaqarang).



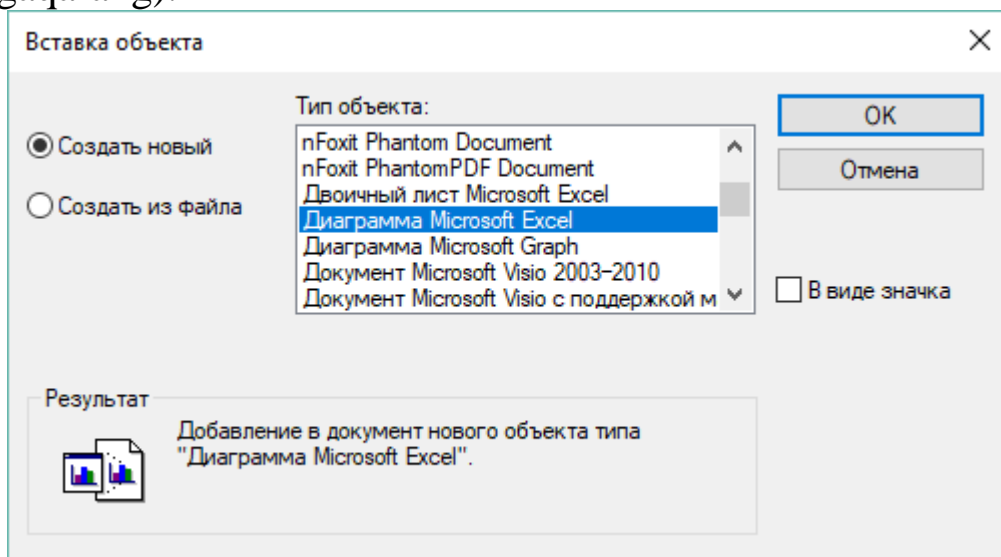
13.1-rasm. MFC application yaratish.

Keyingiqadammuloqotoynasida, kelajakdaqo'llashasoslarinitanlash. Joriydastur "hujjat-ko'rinish" arxitekturasidanfoydalanibyaratilishikerak. Ilovainterfeysibittahujjatyokiko'phujjatbo'lishimumkin. Shundanso'ng "keyingi" tugmasinibosibvakompozithujjatlarnikonteynersifatidaqo'llab-quvvatlashvariantinitanlashlozim. Bundantashqari, faolhujjatkonteynerxususiyatio'rnatishkerak. (13.2-rasmgaqarang)



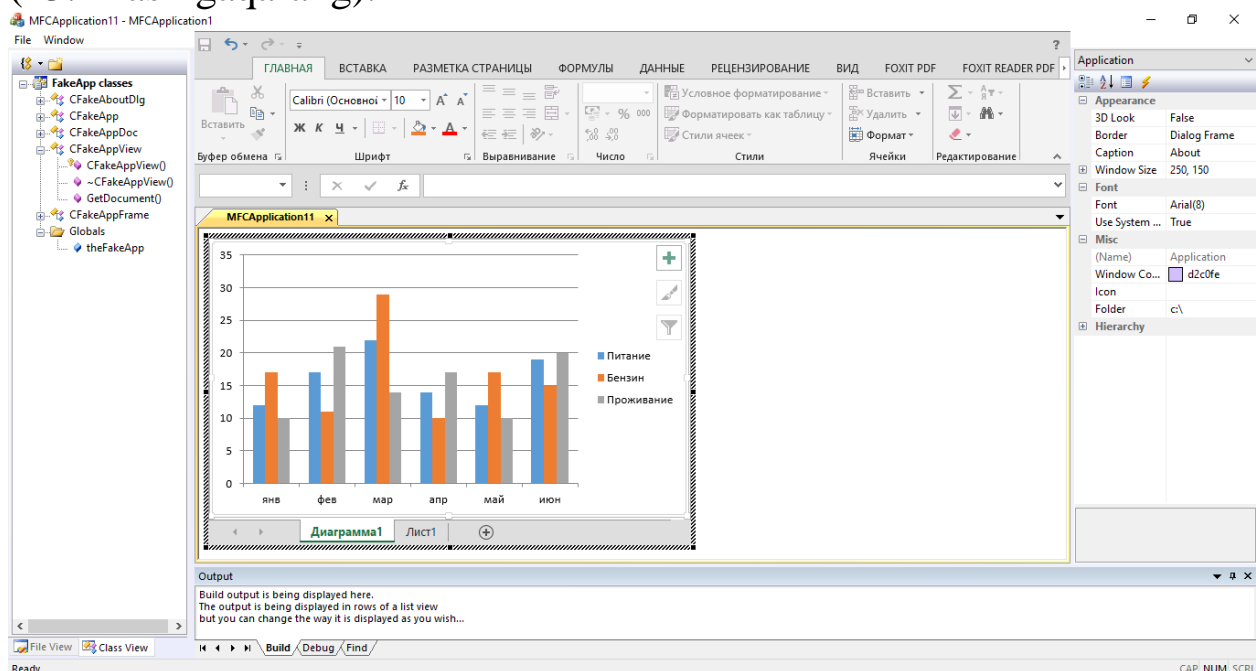
13.2-rasm. Parametrlarni sozlash.

Shundansoʻng "Finish" tugmasini bosish kerak, yoki keyingi sahifalardan oʻtib, zarur hollarda loyihaning parametrlarini oʻzgartirish mumkin. Natijada siz kompilyatsiya uchun ishlatish mumkin, yangi loyihani hisoblanadi. Dasturoynasida Bosh menyubandini tanlash orqali "Edit Insert New Object..." orqali OLE ob'ektlarining roʻyxati bilan muloqot oynasini hosil qilinishi (13.3-rasmga qarang).



13.3-rasm. OLE asosida bogʻlanish.

Misolsifatida Microsoft Excel grafikob'ektinitanlasangiz, dasturoynasiob'ektkiritilgandanso'ngshundayko'rinishgaegabo'ladi (13.4-rasmgaqarang).



13.4-rasm. Microsoft Excel bilan bog'lanish.

Bu OLE ob'ektfal, unio'zgartirish mumkin: grafikuslubio'zgartirish, ma'lumotlarga o'zgarishlar qilish, vakonteyner ichidan uqta formatini o'rnatish ham mumkin.

OLE birkonteyner serialization xususiyatiga ega, bir fayl uchun ma'lumotlarni saqlash va keyin fayl uni qaytarib o'qish mumkin. Ushbu ob'ektni faylga saqlash uchun "File-Save as" nitanlab, dasturning asosiy menyu sidavamuloqot oynasidagi fayl nomi niko'rsatiladi.

OLE konteyneridan foydalanib, mavjud Microsoft Office hujjat fayllarini, masalan, Excel elektron jadvalarini yoki PowerPoint taqdimotlarini, shuningdek Word hujjatlarini ham ko'rishingiz va tahrirlashingiz mumkin. Buning uchun yaratilgan dasturni ishga tushirganda "Edit / Insert New Object" nitanlang. Bu orqali barcha imkoniyatlarni amalga oshirish mumkin.

**Visual C++ muhitidako'poyalimuhitni yaratish.** Visual C++ muhitidako'poyalidasturlarni yaratish uchun SDI va MDI ilovalar konstruktorlari mavjud. Hozirda MDI ilovalar konstruktorlari qaliki chikko'poyalidasturi yaratamiz. Bunda quyidagi qadamlardan foydalaniladi.

1. Yangi formani yaratiladi.
2. Formaning xususiyatlarini o'zgartiriladi.

IsMdiContainer: True  
StartPosition: CenterScreen  
Text: **Notice**

3. Yangiforma qo'shish uchun, asosiy menyusida, click Project -> Add Windows Form...

4. Yagona hujjatgan ommo'ratish

5. Add tugmasini bosing

6. Asboblarning umumiy elementlarini bo'limidan Rich TextBoxni tanlang

7. Xususiyatlar oynasida quyidagicha xususiyatlarni o'zgartiring:

(Name): **rtbNotice**

Dock: Fill

Modifiers: Public

8. Birinchi shaklni ko'rsating

9. Formada o'tingvayangi ikki martasi chiqonchanibosing

10. Faylning yuqori qismida yagona hujjat header faylini tahrirlang:

```
#include "SingleDocument.h"
```

11. Kodning pastgatushingva quyidagicha hodisa amalga oshirish kodini yozing.

```
System::Void  
mnuFileNew_Click(System::Object^ sender,  
System::EventArgs^ e)  
{  
    SingleDocument ^ document = gcnew  
    SingleDocument;  
    document->Text = "Untitled";  
  
    document->MdiParent = this;  
    document->Show();  
  
    mnuFileClose->Enabled = true;  
}
```

12. Formaga qaytiladi.

13. Asboblari panelidani yangi birtugmani o'rnatish.

14. Xususiyatlar oynasining hodisalariga o'tib, tugmani hodisalarini bosing. click maydonini bosish va o'ng tomonida, mnufilenew\_click tanlang

15. Formada o'tingvayangi ikki martasi chiqonchanibosing

16. Quyidagicha tadbir amalga oshiring:

```

System::Void
mnuFileOpen_Click(System::Object^
sender, System::EventArgs^ e)
{
    if( dlgFileOpen->ShowDialog() ==
System::Windows::Forms::DialogResult::OK
)
    {
        for each( String ^ strFile in
dlgFileOpen->FileNames )
        {
            SingleDocument ^ document =
gcnew SingleDocument;

            document->rtbNotice-
>LoadFile(strFile);

            document->MdiParent = this;
            document->Show();
        }
    }
}

```

17. Formagaqaytiladi.

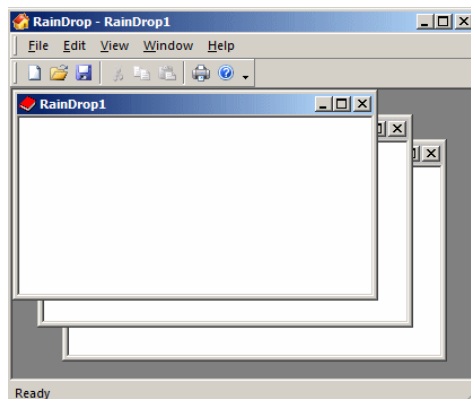
18. Asboblarpneldanyangibirtugmanio‘rning.

19. Xususiyatlaroynasininghodisalarigao‘tib,  
tugmanihodisalarinibosing. click maydoninibosingvao‘ngtomonida,  
mnufilenew\_click tanlang

20. mnufileopen\_click tanlang

21. Formagaqayting.

Natijadayaratilganloyihaniishgatushirsangizquyidagicha,  
dasturhosilqilinadi.





## NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

1. MFC nima, uqandayimkoniyat.
2. MFC va API ningbog‘liqxususiyatlarinibilasizmi?
3. MFC ningengasosiyxususityainima.
4. Visual Studio 2008 tizimidanqandaykomponentalarniolib, qaysisinflarasosida MFC ajralmasqismiqlibyaratildi.
5. MFC 9.0.21022 (mfc90.dll), .NET 3.5 ningimkoniyatlarinisanabbering.
6. MFC 14.10.25008.0 (mfc140.dll), .NET 4.6.2 ningimkoniyatlarinisanabbering.
7. OLEtexnologiyasiningimkoniyatinisanabbering.
8. OLE murakkabhujjatlarniqaytaishlashuchunishlatiladi, qaysiinterfeysiorqaliturlibog‘liqbo‘lmagantizimlaro‘rtasidama’lumotlarniuzatishuchunfoydalanishmumkin.
9. Qaysitexnologiyavauningmetodi Microsoft Windows operatsiontiziminingoldingiversiyalaridaishlatiladigan DDE (dinamikma’lumotlaralmashinuvi) texnologiyasiasosida 1990-yildachi qarilgan
10. OLE 1.0 vaOLE 2.2 texnologiyalarinifarqinitushuntiribbering.
11. MFC vaOLEtexnologiyalaribog‘liqmivanimauchun.
12. Ko‘poyalimuhitlarningavfzalliginimada?
13. Formaxususiyatlarinio‘zgartirishnimauchunkerak.
14. Dock xususiyatininimauchun Fill qiymatgao‘zgartiriladi.
15. mnufileopen\_click hodisasiningvazifasinitushuntiribbering.

### 3.6.Kichikloyihalarniyaratish

#### Visual

*C++ningloyihayaratishusullarivauskunalarpanelidagielementlardanfoydalanishasosidakichikloyihalarniyaratishtalablarivausullari, loyihadafoydalaniladiganalgoritmlarniifodalashvaloyihalashasosidatahlilqilishni, hamdahisobotlarniyaratishusullarivaelementlaribo‘yichanazariybilimlar keltirilgan. Bilimlarnimustahkamlashuchun 15 tanazariysavolberilgan.*

***✍ Kalitso‘zlar.*** Operatsiontizim, amaliydastur, kichikloyiha, axborottizimi, IDEF, DFD, Hayotsikli, algoritm, EPSmetodologiyasi, BRMImetodologiyaisi, hisobot, hisobotpaneli.

***☑ Bilishshartbo‘lgantushunchalar.*** Visual

C++ningloyihayaratishusullarivauskunalarpanelidagielementlardanfoydalanish, ishlash, sinfvasinfob’ekti, xususiyat, hodisa, forma, komponentadasturlashgaoiddastlabkitushunchalarhamdaC++tiliniqo‘llabquvvatlovchimuhitdaishlashnibilishlozim.

***🔗 Bilibolasiz.*** Kichikloyihalarni loyihalashtirishusullari, hayotdavri, loyihadagialgoritmlarningifodalashninguchullarivatalablari, Visual C++ muhitidasturlash, Visual C++ninghisobotlarniyaratishkonstruktorlari usularivaba‘zibirxususiyatlari, hodisalarior‘ganishingiz mumkin.

## **REJA**

1. Visual C++ muhitidaturli sohagaoid masalalarni yechish uchun kichik loyihalarni loyihalashtirish usullari.

2. Kichik loyihalalar algoritmlarini yozish usullari.

3. Visual C++ muhitida hisobot shaklini tayyorlash va chop etish elementlari.

## **KIRISH**

Visual C++ muhitidaturli sohagaoid masalalarni yechish uchun kichik loyihalarni loyihalashtirish usullari asosan axborottizimlarni loyihalashtirish usullarini nazarda tutiladi. Shuning uchun axborottizimining ob’ektlariva usullari, algoritmlariga, qo‘llanish sohasi va chegaralariga qarab loyihani katta, kichik loyihalarda yish mumkin.

Shuning uchun kichik loyihadeganda funksionalimkoniyatlarisanoqlibo‘lganaxborottizimlaritushuniladi.

Axborottizimlarining loyihalashtirish uchun axborottizimining hayotdavrini yaxshibilish va tushunish kerak.

Chunki axborottizimni loyihalashda uchun hayot davri muhim ahamiyatga ega.

Axborottizimining hayot davri. Hayot davri (HD) tushunchasi axborottizimlarini loyihalash metodologiyasining asosiy tushunchalaridan biridir. Axborottizimining hayoti ydari -

bu axborottizimni yaratish fikridan boshlanib,

ubutunlay foydalanihdan chiqqan paytda tugaydigan uzluksiz jarayondir.

Hayot davrining tuzilishini belgilovchi asosiy standart GOST ISO/IEC

12207-02 hisoblanadi. Standartgako‘ra, hayotdavrstrukturasi uchta jarayon guruhiga asoslangan:

1. **Asosiy** (buyurtma, ishlabchiqish, etkazibberish, sinov, texnik xizmat ko‘rsatish);

2. **Yordamchi** (asosiy jarayonlarning bajarilishini ta‘minlash uchun):

- Hujjatlar uchun -

barcha manfaatdor shaxslar tomonidan zarur bo‘lgan hujjatlarni ishlabchiqish, tahrirlash, tarqatish va saqlash bo‘yicha ishlar;

- konfiguratsiya niboshqarish (configuration management)

quyidagilarni o‘z ichiga oladi:

tizimdagidasturiy ob‘ektlarning holatini aniqlash va o‘rnatish;

ob‘ektlarning o‘zgarishini va chiqarilishini niboshqarish;

ob‘ektlarning to‘liqligi, mosligi va to‘g‘riligini ta‘minlash;

ob‘ektlarni saqlash, tashish va etkazibberishni niboshqarish;

- sifatini ta‘minlash -

yaratilayotgan tizim va amalga oshirilayotgan hayotdavi jarayonlarining belgilangan talablarga javob berishini ta‘minlash ishlarini;

- tekshirish -

loyiha amalga oshirilayotgan dastur yaratilgan muvaqqat natijalarning belgilangan talablarga javob berishini tekshirish uchun tegishli shaxs (buyurtmachilik, etkazibberuvchi yoki mustaqil tomon) ning ishi. Shartnoma, jarayon, talablar, loyihaning tizimi, qurilish tizimining hujjatlarni tekshirishni ajrata;

- sertifikatlash -

talablarning va yakuniy mahsulotning tizimning funksional maqsadiga to‘liq mosligini tekshirish uchun tegishli shaxsning ishi;

- qo‘shimcha tahlil - har qanday ish (tizim)

holatini yoki natijalarini baholash uchun ish;

- audit - mustaqil (loyihaning isbatidan)

ekspertlarning korxonada faoliyatining qabul qilingan talablar, rejalarga shartnomada shartlariga muvofiqligini aniqlash ishlarini;

- muammoni hal qilish -

loyihani amalga oshirish vaqtida topilgan muammolarni tahlil qilish va tuzatish ishlarini;

3. **Tashkiliy:**

- loyihani boshqaruvi - rejalashtirish va jarayonlarni niboshqarish, shu jumladan monitoring,

tekshirish va hisobot bilan tugallangan ishlarini baholash;

- loyihani infratuzilmasini yaratish -

boshqariladigan jarayon uchun zarur bo‘lgan infratuzilmani tashkilot va

amirlash uchun ishlaydi. Infrastruktura texnika dasturiy vositalar, vositalar, usullar, standartlar va tizimni ishlab chiqish, ishlatish yoki xizmat ko'rsatish shartlari bo'lishi mumkin;

- yaxshilash - hayot davri jarayonlarini baholash, kuzatish va yaxshilash uchun ishlash;

- o'quv-kadrlar tayyorlashni rejalashtirish va o'tkazish, shu jumladan o'quv materiallarini ishlab chiqish. Shubhan birga,

xodimlarni faqat tizimni boshqaradigan oxirgi foydalanuvchilarga, balki tizimni ishlab chiquvchilari gaham tegishlidir. Masalan,

ishlab chiquvchi lartash kilotda qabul qilingan texnologiyalar va dasturiy vositalar bo'yicha o'qitilishi,

hatto oxirgi foydalanuvchilarni tizim bilan ishlashni to'g'ri amalga oshirish va o'qitishga o'rgatish kerak. Qanchalik paradoksal bo'lsin,

lekin o'qitishning to'g'ri usul va metodlarini o'rgatish ham kerak.

AT yaratish uchun asosiy mezon quyidagilarni o'z ichiga oladi:

- tizimning funksional va tashkiliy tuzilmalari;
- texnika dasturiy vositalar majmuasining tarkibi va tuzilishi;
- ishlatiladigan uskunalar;
- axborotni qayta ishlash texnologiyalari;
- axborot bazasini yuritish tarkibi, tuzilishi va texnologiyasi;
- kirish va chiqish formalari;
- ma'lumotlarni qayta ishlash algoritmlari.

Axborot tizimni ishlab chiqishdan oldin mijoz va ishlab chiquvchi tizim daqanday funksional ko'rinishini va tizim ichida qanday funksional o'zaro munosabatlartashkili tushunish kerak.

Funksional modelni ishlab chiqishda (funktional talablarni belgilashda) ko'pgina muammolar paydo bo'lishi mumkin:

- mijoz axborot tizimi oldiga qanday vazifalar qo'yilganini aniqlifoda eta olmay di. Ko'pincha mijoz ham talabni mavajunishakllantirish qanday bilmaydi;

- buyurtmachilarning vakillari (turlidarajadagirahbarlar, texnologiya mutaxassislari, oddiy foydalanuvchilar) kelajak tizimi haqida o'z qarashlariga ega va ko'pinchaularning tizim gabo'lg antalablari o'zaro farqqiladi. Ushbu holat, ayniqsa, ishlab chiqilayotgan tizim bir nechta avtomatlashtirish ob'ektlarida amalga oshirilganda odatiy holdir;

-  
mijozko 'pinchazamonaviy hisoblashtizimlarining imkoniyatlarini bilmayd  
iva avtomatlashtirish jarayonini qo'ldabajariladigan elementar faoliyatni ko  
mpyuterlarga oddiy uzatish deb hisoblash g'ayri iladi. Shubilan birga,  
ulardan yangi texnologiyalar kelishi bilan tashkilot ichida g'ibiznes-  
jarayonlarni optimallashtirish haqida o'ylamaydilar;

-  
mijozba'zi funksiyalarni virtual mashinalar bajarishi mumkinligiga ishonma  
ydi.

Funksional model qurish, bu muammolarni eng avval hal qilish kerak.  
Buzarayon batafsil ko'rib chiqamiz.

Uni ishlab chiqishda ishni mavjud tashkilotning modeli avvalo ishtav  
siflari, farmoyishlari, hisobotlari,  
me'yoriy hujjatlarini vaboshqalar asosida quriladi.

Modelni tahlil qilish zaif tomonlar qo'lda,  
yangi jarayonlarning afzalliklari qanday bo'lishini vakorxonalar (kompaniya,  
bo'lim)

ning mavjud tashkilotiga qanday chuquro'zgarishlar kiritilishi tushunishga  
imkon berishi lozim.

Faoliyatni samarasiz tashkilotning belgilarini bo'lishi mumkin:

- foydasiz, nazoratdan tashqari vduplikativ ishlar;
- natijasiz ishlar;
- samarasiz hujjat oqimi (to'g'ri hujjat o'z vaqtida to'g'ri joyda emas)

vaboshqalar.

Modelni topilgan kamchiliklar modelini yaratishda tuzatilib,  
korxonaning yangi tashkilot modelini taklif qilinishi ham mumkin.

Bu modelni muammoni hal qilishning muqobil yo'llarini tahlil qilish va eng yaxs  
hisobni tahlil uchun ishlatiladi.

SADT metodologiyasi (Structured Analysis and Design  
Technique - struktural tahlil va loyiha lashtexnikasi) -  
tizimning funksional modelini qurish uchun mo'ljallangan usullar,  
qoidalar va protseduralar majmuidir.

Bu metodni ishlab chiqishni XX asrning 60-yillari o'rtalarida  
Duglas Ross (AQSh) boshlab bergan. Keyin, Softech, Inc  
tahlilchilardan tomonidan SADT  
yaxshilandi va muammolarni keng ko'lami hal qilish uchun ishlatiladi. Telefon  
tarmog'ida sturiya minot, diagnostika,  
uzoq muddatli va strategik rejalashtirish,  
kompyuter yordamida ishlab chiqarish va loyiha lashtirish,

kompyutertizimikonfiguratsiya, kadrlartayyorlash, vamoliyaviyvalogistikaboshqarishSADTsamaraliilovalaryaratilaboshlandi.

KengsohalarSADTmetodikasiningko'pqirraliligivaqudratiniko'rsatadi. Amerika

qo'shmaShtatlariMudofaavazirliginingintegratsiyalashgankompyuterquv vatishlabchiqarish (ICAM) dasturiSADTningfoydaliliginitanoldi. Bu 1981-yildauningbirqismini "IDEF0" (Integrated Computer Aided Manufacturing, ICAM) debnomlab, dasturiya'tminotniishlabchiqishbo'yicha Federal standart sifatidachopetishgaolibkeldi.

BunomostidaSADTdanharbiyvasanoattashkilotlaridaminglabmutaxassislarfoydalanishgan. MilliystandartlarvatexnologiyalarinstitutidaIDEF0 standartiningso'nggiversiyasi 1993-yilningdekabridachiqarildi.

Shunita'kidlashkerakki, IDEF0 RossiyavaRespublikamizGosstandartomonidanfoydalanishuchuntavsiyaetildivamilliydavlatidoralaridafaolqo'llaniladi.

Ushbumetodologiyaaxborottiziminingfunktionaljihatinitavsiflashdama'lumotlaroqimigayo'naltirilganusullar (DFD) bilanraqobatlashadi. Aksincha, IDEF0 imkoniyatlariko'yidagilardanberadi:

- harqandaytizimnitavsiflash, nafaqataxborottizimlari (dfddasturiya'tminotnitasvirlashuchunmo'ljallangan);

- ungaqo'yiladiganyakuniyatalablarnibelgilashdanoldintizimvauningtashqi muhittavsifiniyaratish. Boshqaso'zlarbilanaytganda, bumetodologiyasiyordamida, sizastasekinqurishvauningbajarilishinitasavvurqilishqiyinbo'lsaham, tizimtahlilqilishmumkin.

Shundayqilib, IDEF0 kengdoiradagitizimlarniyaratishningdastlabkibosqichlaridafoydalanishmumkin.

Shubilanbirgamavjudtizimlarningvazifalarinitahlilqilishvaularnitakomillashtirishechimlariniishlabchiqishuchunfoydalanishmumkin.

IDEF0 metodikasigrafikjarayonnitasvirlashtiligaasoslangan. IDEF0 notatsiyadagi model iyerarxiktartiblivao'zarobog'langandiagrammalarto'plamidir. Harbirdiagrammatizimtavsifiningbirligibo'lib, alohidavaraqdajoylashgan.

Model (AS-IS, TO-BE yoki SHOULD-BE) diagrammalar 4 turnio‘zichigaolishimumkin

- kontekstdiagrammasi;
- parchalanganda diagrams;
- tugundaraxtdiagrammalari;
- faqatEHMuchungrafiklar (FEO).

Diagrammalardaraxttuzilishiningyuqorisibo‘lmishkontekstdiagramma (top-level diagramma) tizimningmaqsadi (asosiyvazifasi) vauningtashqimuhitbilano‘zarota‘siriniko‘rsatadi.

Harbirmodelfaqatbittakontekstgrafikkaegabo‘lishimumkin.

Asosiyfunksiyaniavsiiflagandanso‘ngfunktionaldekompozitsiyaamalgaosshiriladi, ya‘niasosiyfunksiyani tashkiletuvchifunksiyalaraniqlanadi.

Bundantashkari,

o‘rganilayotgantizimningkeraklidarajadadetalashtirilgungaqadarfunksiyalarsubfunksiyalargabo‘linadivahokazo.

Sistemaningharbirshundayfragmentinitasvirlovchidiagrammalardekompozitsiondiagrammalardeyiladi.

Harbirdekompozitsiyaseansidanso‘ngeksperttekshiruvseanslario‘tkaziladi

domenekspertlarirealjarayonlaryaratilgandiagrammlargamoskelishiniko‘rsatadilar. Topilganharqandaynomuvofiqliklarbartarafetiladi, so‘ngrajarayonlarniyanadabatafsildavometiradi.

Tugundaraxtdiagrammasidafunksiyalar (isho‘rinlari) ningiyerarxikmunosabatlariko‘rsatilgan, lekinularo‘rtasidagimunosabatlaremas. Ulardanbirnechtasibo‘lishimumkin, chunkidaraxtharqandaychuqurlikkavaharqandaytugundanqurilishimumkin.

EHMdiagrammalaritizimdayuzberadiganjarayonlarningmuqobilko‘rinishiniko‘rsatishuchunmodelningalohidaqismlariniko‘rsatishuchunmo‘ljallangan (masalan, tashkilotboshqaruvinuqtainazaridan).

**Kichikloyihalaralgorimlariniyozishusullari.** Tizimvazifalarinibelgilabvaaxborotasoslariniishlabchiqqandanso‘ng, keyingiqadamtizimidaaxborotoqimiharakatilo'yihalashtirishiborat. Tizimningmodelikeraklifunksiyalargaqandayerishishvaunita‘minlashuchunqandayma'lumotlarishlatilishiniko‘rsatadi. Shundayqilib, modelbevositatizimningfunktionalvaaxborotmodellarigaasoslanadi.

BumodellarodatdaIDEF0 vaDFDdekompozitsiyadiagrammalariningoxirgidarajalaridako‘rsatilganf

unksiyalar (jarayonlar) uchunquriladi. DFDuchundiagrammayokidiagrammako‘rinishidagimodelielementarjara yonlarniminispektlarko‘rinishidatasvirlashdanko‘rayaxshiroqechimdir.

Bundayvaziyatlardafunksionalimkoniyatnitasvirlashningengentan iqlivamashhurusullarivauslubiyatibor:

- algoritmlarningblokdiagrammalari;
- JadvalningEPC ;
- BPMNmetodologiyasi.

AlgoritmlarvaBPMNdiagrammalariningoqimlarian‘anaviytizimlar nitizimliyondashuvtamoyillarigaasoslanganholdamodellashtirishuchunen gmoskeladivafunksiyalarnielementarbosqichlarga (protseduralar, operatorlar, harakatlarvahokazolarga) aniqbo‘linishibilantavsiflanadi.) ma‘lumketma-ketlikda (algoritmbo‘yicha) bajariladi. EPC diagrammalarvaBPMNdiagrammalarbirharakatyokiularningmajmuijrosi tizimidasadirbo‘lganvoqealarbog‘liqbo‘lganvoqeagayo‘naltirilgantiziml arinimodellashtirishuchunyaxshivositahisoblanadi. Bundantashqari, buusulvauslubiyatlardannafaqatxulq-atvorlimodellashtirishuchun, balkifunksionalmodellashtirishuchunhamfoydalanishmumkin.

Ulardama‘lumamallarni (mantiqiysimvollarni) bajarishshartlariniko‘rsatishgaimkonberuvchielementlarningmavjudligiti zimfunksiyalariningmantiqivaketma-ketliginiyaxshiroqtushunishgaimkonberadi.

Algoritmlarnitasvirlashdauzoqvaqt davomida flowcharts (Basic Flowchart) muvaffaqiyatliishlatilgan. AlgoritmlarningblokdiagrammalariniquyurishGOST 19.701-90 (ISO 5807-85) dasturiyhujjatlarningyagonatizimibilantartibgasolinadi. Dasturlar, ma‘lumotlarvatizimlaruchunalgoritmlarning sxemalarimavjud.

Ushbudavlatstandarti "ISO 5807-85" xalqarostandartiasosidatashkiletgan. Axborotniqaytaishlash – ma‘lumotlar, dasturvatizimoquvchilari, dasturtarmoqjadvallari vatizimresurslarivadvallariuchunhujjatsimvollari v akonvensiyalarimoslashtirilgan.

GOST 19.701-90 gako‘ra, diagramma- bumuammonihalqilishningta‘rifi, tahliliyoki usulininggrafiktasviri. Tizimningstatikvadinamikjihatlarniniko‘rsatishuchundiagrammalardanfoydalanishingiz mumkin.

Davlatstandartidaberilgansimvollardanquyidagiturdagisxemalardafoydal anishmumkin:



- ma'lumotlarsxemalari - ma'lumotlarniqaytaishlashketma-ketliginivaularningtashuvchilarinianiqlaydi;

- dasturdiagrammalari - dasturdagiamallarketma-ketliginiko'rsatish (aslidabuan'anaviyma'nodaalgoritmlarningasosiyxususiyati);

- tizimningishlashsxemasidatizimdagiamallarvama'lumotlaroqimlariko'rsatiladi;

- dasturningo'zaroishlashdiagrammalari - dasturlar (modullar) ningaktivlashtirishyo'liniko'rsatishvaularningtegishlima'lumotlarbilano'zarohamkorligiuchun;

- tizimresursdiagrammalari - ma'lumotlarbloklarivaqaytaishlashbloklarikonfiguratsiyasiniko'rsatish.

Yuqoridakeltirilganturdagisxemalardanko'rinibturibdiki, ulardannafaqatxulq-atvorob'ektinimodellashtirishuchun, balkifunksional, axborotvakomponentliloyihalashvazifalariuchunhamfoydalanishmumkin

Tizimningfunktionalmodeliniquyidagistrukturaviyyondashuvninga sositamoyillari

dekompozitsiyavaiyerarxikalgoritmtamoyillaridanfoydalaniladi.

Funktionalmodeli

buturlidarajadagidetallargaegabo'lgano'zarobog'langandiagrammalarmajmuibo'lib, harbiryangidetaldarajasibilantizimyanadato'liqlashadi.

Diagrammalargrafikbelgilashningquyidagielementlarinio'zichigaolishimumkin:

- data symbollar - ma'lumotlarningmavjudligini, axborotlarningturiniyokima'lumotlarnikiritishvachiqarishusuliniko'rsatish;

- jarayonsimvollari - ma'lumotlarustidabajariladiganamallar;

- chiziq symbols

jarayonlarva/yokisaqlashvositalariorasidagima'lumotlaroqimlarini, shuningdekjarayonlarorasidaginazoratoqimlariniko'rsatadi;

-maxsusbelgilar

diagrammalarniyoziyishvao'qishnisonlashtirishuchunishlatiladi.

Semantikmazmungaako'rabo'linishdantashqariharbirturkumsimvollari (maxsuskategoriylardantashqari)

asosiyvao'zigaxossimvollargabo'linadi.

Asosiybelgijarayonyokima'lumotlartashuvchininganiqturinoma'lumbo'lgandayokihaqiqiy ma'lumotlartashuvchisi (jarayoni)

nita 'riflashgahojatbo' lmagandaqo' llaniladi.

Jarayonyokima' lumotlartashuvchininganiqturima' lumbo' lgandavadiagrammadako' rsatilishikerakbo' lgandao' zigaxossimvoldanfoydalaniladi.

Hodisalijarayonzanjiri (EPC) - biznes-jarayonlarnimodellashtirish, tahlilqilishvaqaytatashkiletish (funktionalmodellashtirish) uchunishlatiladigandiagrammaturo. ShubilanbirgaEPC diagrammalaridanfunktionalarniamalgaoshirishdatizimningalohidaqismlarifunktsionalliknimodellashtirishdafoydalanishvaan' anaviyoqimlarnialmashtirishuchunxizmatqilishmumkin.

EPSmetodiAvgust-VilgelmSheertomonidan 1990-yillarboshidaintegrallashganaxborotizimlariarxitekturasimetodikasigaoi dishiningbirqismisifatidaishlabchiqilgan.

EPC notatsiyasidajarayon (funksiya) diagrammasihodisavafunktionalarningtartibliirikmasidir. Harbifunksiyauchunungahamrohbo' lgandastlabkivaoxirgihodisalar, ishtirokchilar, ijrochilar, moddiyvaaxborotoqimlarianiqlanishihamdaquyidarajalargabo' linishimumkin.

DFDKabi, EPC metodologiyasiturlinotatsiyalarni (elementlarningsintaksisivasemantikasini) qo' llabtalqinqilishbilanrivojlanmoqda. EPC (ARIS, Microsoft Visio, Business Studio, Bflow) yordamidabiznes-jarayonlarnimodellashtirishqobiliyatiniamalgaoshirganmetodikamuallifi ham, dasturiyta' minotishlabchiqaruvchilarihambungaqo' lurishgan. Blokdigrammasigao' xshashlikbo' yichasimvollar (elementlar) grafikyozuvtlarinimaqsadgamuvofiqguruhlasmumkin. Quyidagijadvaldaadabiyotvadasturiyta' minotdaengko' puchraydiganEPC larivaularningmuqobiltasvirlarimavjud.

EPC yordamidadekompozitsiyajarayonlarnimodellashtirishvaiyerarxikalgoritmlarjarayoninimodellashtirishningklassiktamoyillarigaamalqiladiyu. Dekompozitsiya, alohidadiagrammalardako' rsatishbilan, IDEFO diagrammalardaoldindanbelgilanganjarayonlardaishlashgao' xshashfunktionalaryalaruchunbajariladi.

1. EPC jarayoniningdiagrammasikamidabirboshlashhodisasibilanboshlashkerak vakamidabirtugashhodisasibilanyakunlandi.

2.

Jarayon rivojlanib borgan sari hodisalar va funksiyalar almashinib turish kerak.

3. Diagrammavazifalaritavsiyasoniko‘p bilan 20 tabo‘lishi mumkin. Agar grafik funksiyalar soni 20 dan sezilarli darajada oshsa, yuqoridarajadagi jarayonlarni o‘g‘rita‘kidlangan va modelni sozlash kerak bo‘ladi.

4.

Hodisalar va funksiyalar jarayonning borishini aksettiruvchi qat‘iy bittakiruvchivabittachiquvchialoqani (nazorat oqimini) o‘zichiga olishi kerak.

5. Grafik bir tegishsiz elementlar bo‘lmasligi kerak.

6.

Yuqori modul diagrammasidafunksiya atrofidagi hodisalar va mantiqiy operatorlar funksiyadekompozitsiyadiagrammasidaboshlang‘ich / natijaviy hodisalar va operatorlar bo‘lishi kerak.

7.

Har bir birlashma kamida ikki takiruvchi ulanishga va faqat bittachiquvchi, ayirmalar

faqat bittakiruvchialoqavakamida ikki takiruvchibo‘lishi kerak.

Operatorlar bir vaqtning o‘zida bir nechta takiruvchivachiquvchibog‘lanishlar ga ega bo‘la olmaydi.

8.

Mantiqiy operatorlar faqat funksiyalarni yoki faqat hodisalar ni birlashtirishiy oki birlashtirmasligi mumkin.

Bir vaqtning o‘zida funksiyavahodisanibirlashtira olmaydi.

**Visual**

**C++**

**muhitida hisobot shaklini tayyorlash va chop etish elementlari.** Buning uchun ma’lumotlar bazasini bilishtirish talab qilinadi.

Quyidagichamasalaberilgan bo‘lsin.

Masala.

Universitet ma’lumotlar bazasiberilgan bo‘lsin.

Undatalab avasemestri nomli jadvallar bo‘lsin.

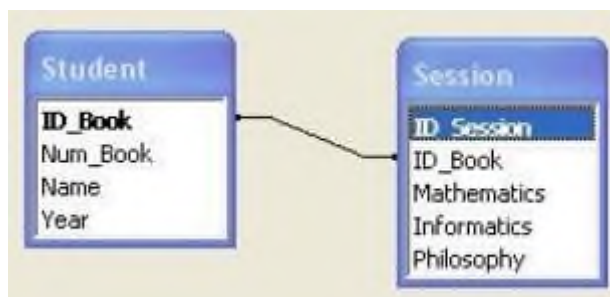
14.1-jadval. Talabajarjadvalining tuzilishi

Maydon nomi	tipi	Izoh
ID_Book	Butunson (int)	Birlamchikalitmaydon
Num_Book	matnli (10 belgi)	Talabaning greytingdaftari raqami
Name	matnli (30 belgi)	Talabaning ismi sharifi
Year	Butunson (int)	Tug‘ilgan yili

14.2-jadval. Semestri jadvalining tuzilishi

Maydonnomi	tipi	Izoh
ID_Session	Butunson (int)	Birlamchikalitmaydon
ID_Book	Butunson (int)	reyingdaftariraqami
Mathematics	Butunson (int)	Matematikabahosi
Informatics	Butunson (int)	Informatikabahosi
Philosophy	Butunson (int)	Falsafabahosi

Ushbujadvallarningo‘zaromunosabatiquyidagibog‘lanishorqaliifod alanganbo‘lsin.



14.1-rasm. Jadvallarningrelyatsionbog‘lanishi.

Berilganma‘lumotlarasosidahisobotlardanfoydalanishnitashkilqilib beruvchidasturniishlabchiqishniko‘ramiz.

Quyidagima‘lumotlarnio‘zichigaolganhisobotyaratishlozim:

- Reyingdaftarini
- Talabangismi – sharifi
- Matematikabahosi
- Informatikabahosi
- Falsafabahosi
- Talabangio‘rtachabahosi

Jorihisobotniyaratishdama‘lumotlar Microsoft Access DBMS tomonidanhosilqilinganso‘rovorqaliolinadi. So‘rov Query1 debataladi. Hisobotalohidashakldako‘rsatilishikerak.

Buninguchunquyidagiamallar bajariladi:

1. Microsoft Visual Studio tizimiorqali Windows Forms dasturshabloniyordamidalojihayaratiladi. Microsoft Visual Studio ish gatushirilgandanso‘ng, siz yangiloyihayaratishkerak. Windows Forms dasturshabloniyordamidayangiloyihaniyaratishvasaqlashningbatafsilolro qytibo‘tilgan.

2. Ma‘lumotlar bazasini yaratish yoki tayyorm a‘lumotlar bazasifaylini yuklash lozim. Ma‘lumotlar bazasi uchun Microsoft Access ma‘lumotlar bazasini boshqarishtizimiyordamidayaratiladi.

Ma‘lumotlar bazasida ikki tategishli jadval talabavasesmestr, shuningdek, hisobotni yaratish uchun ishlatiladigan Query1 so‘rovi yaratib olish kerak.

### 3. Ma'lumotlar bazasifaylining loyihaga ulanishi.

Ma'lumotlar bazasifaylini loyihaga ulashdan oldin ushbu faylni loyihafayllar isaqqlanadigan papkaga saqlashni tavsiya etamiz. Foydalanish uchun "Education.mdb" ma'lumotlar bazasi, uni standart usulda loyihaga ulash kerak.

Loyihaga ma'lumotlar bazasifaylini ulash uchun "Add Connection" dan foydalaniladi.

Bu amalkontekst meniyudan buyruq ro'yxatida chaqiriladi.

### 4. Natijada "ma'lumotlar manbasini tanlash"

oynasidama'lumotlar manbai (Data Source) tanlash uchun - Microsoft Access ma'lumotlar bazasifaylini ulash kerak. Keyingi oynada "ma'lumotlar bazasifaylnomi" dagi "ulanish qo'shish" ma'lumotlar bazasifayligayo'lni ko'rsatiladi ("Browse" tugmasiorqali).

### 5. Asosiy loyiha oynasini ishlab chiqish.

Topshiriq sharti gako'rahisobot alohida shakldako'rsatilishi lozim.

Shuning uchun loyiha ning asosiy shakli 14.2-rasmdako'rsatilganidek shaklga egabolaradi.



14.2-rasm.

Agar rasmdako'rib turganingizdek, shakl button1 nomidagi faqat bitta tugma turini o'z ichiga oladi. "Show Report" tugmasini bosganda so'ngi hisobot boshqachashakldako'rsatiladi. Quyidagi xususiyatlarini sozlash kerak:

Boshqarish elementi button1 ning Text = «Show Button»

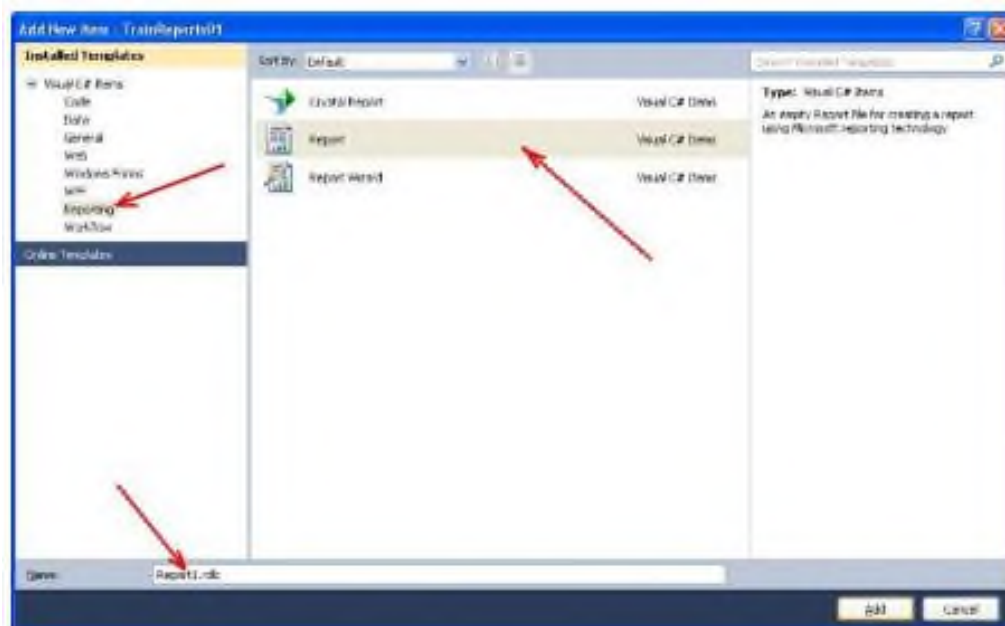
Boshqarish elementi Form1 ning Text = «Report Viewer»

6. Hisobot faylining loyihaga ulanishi. Microsoft Visual Studio dasturida har bir hisobotda «\*.rdlc» kengaytmasiga egafaylmavjud.

Bu fayl hisobotda hosil qilingan ma'lumotlarni o'z ichiga oladi. Microsoft Visual Studio dasturida fayl yaratish uchun loyiha meniyusidan Add New Item buyruq'ini chaqiriladi (14.3-rasm).



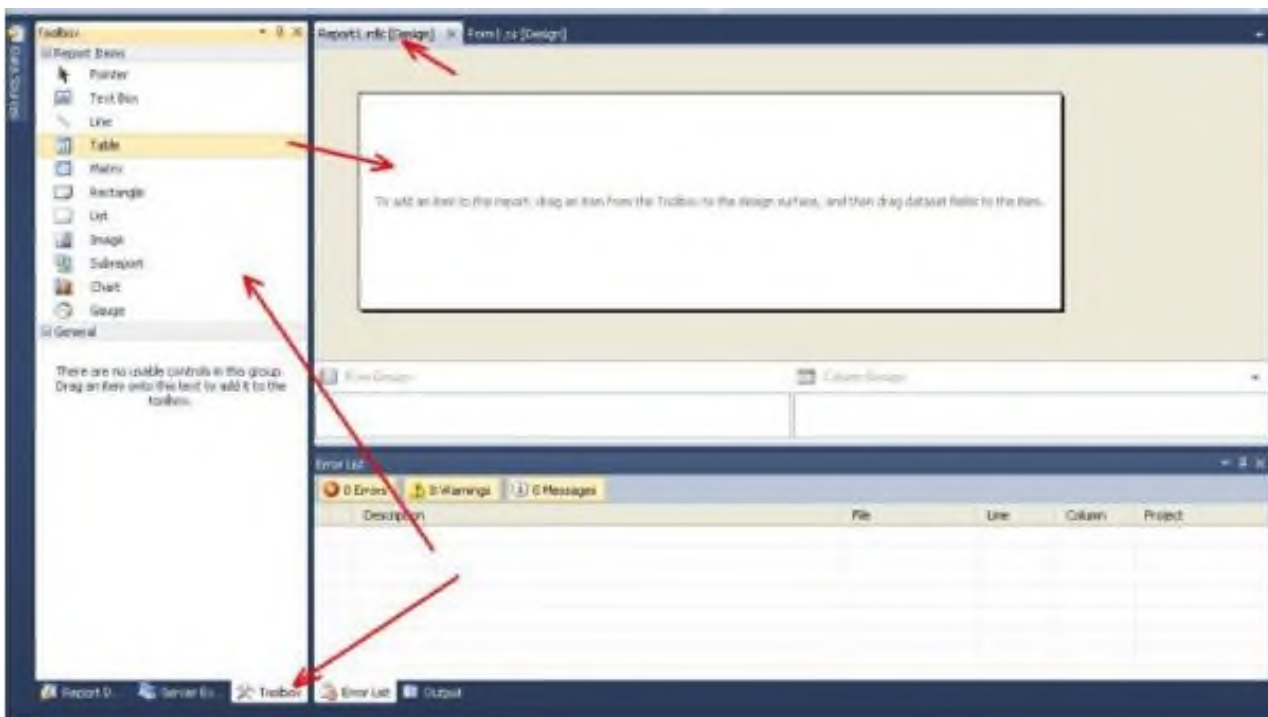
14.3-rasm. Loyihamenyusidan Add New Item amali Natijada, «Add New Item...» hisobotshabloninitanlashkerakbo‘lganoynaochiladi (14.4-rasm) vahisobotfayliuchunnombelgilanadi.



14.4-Rasm. Hisobotshabloninitanlashvahisobotnio‘rnatish Tanlovnitasdiqlagandanso‘ng Microsoft Visual Studio oynasidahisobotningasosiyoyinasiko‘rsatiladi.

Hisobotoynasidahisobotni loyihalashtirish lozim. Buning uchun quyidagilozim.

1. ToolBox asboblari paneli. ToolBox paneli - hisobot elementlarini loyihalash uchun ishlatiladi.



14.5-rasm. ToolBox panel vahisobotloyihalashtirishvositalari.

2. Hisobot yaratish uchun - Microsoft Visual Studio hisobotini ishlab chiqish uchun bir qator taklif etiladigan standart shablonlar bor. Boshqaruvlardan turli hisobotlarning ko'rinishini (taqdimotini) loyihalashtirish uchun foydalanishingiz mumkin.

Quyidagi elementlar taklif etiladi:

TextBox - Hisoblash natijasida olingan teglar, maydonlari yoki qiymatlarni ko'rsatadi;

Line - nuqtayoki undanko'p qalinlikka ega bo'lgan chiziqchizadi;

Table (jadval) - nazorat ustunlar va qatorlar o'zgaraydigan ma'lumotlarni ko'rsatadi;

Matrix - o'zgaruvchan raqamli, ustunlar va qatorlar o'zgaruvchan raqamli borma'lumotlarni jamlash ko'rsatiladi;

Rectangle - hisobot elementlari uchun shablonsifatida to'rtburchak chizadi;

List - har bir guruh yoki ma'lumotlar qatori uchun takrorlanadigan hisobot bandlarini o'plamini ko'rsatadi;

Image - tasvirni bitmap sifatida ko'rsatadi (masalan, logotipi yoki fotosurat);

Subreport - hisobot chegaralaridagi bog'liq sub-hisobot ko'rsatadi;

Chart - Ma'lumotlarni har xil turdagi diagrammalari ko'rsatadi;

Gauge - Maydonyokiifodaningqiymatinichiziqliiyoki radial qarama-qarshilikko‘rinishidako‘rsatadi.

Harbinnazoratdanfoydalanishningo‘zigaxosxususiyatlarigakirmasda n, muammonihalqilishuchunjadvalnazoratidanfoydalanamiz. Bunazoratustunlarsonvaqatorlarbiro‘zgarmaydiganqatorborma’lumotlarn iko‘rsatadi.

Buelementlarningfoydalanishxususiyatlaritizimdabirxilamalgaoshiri ladi.

## NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.

### 1. Visual

C++

muhitidaturlisohagaoidmasalalarniyechishuchunkichikl oyihalarniloyihalashusullariasosanqandaytizimlarniloyi halashtirishusullarinazardatutiladi.

2. Axborottizimlariningloyihalashtirishuchunaxborottiziminingq andaydavriniaxshibilishvatushunishkerak?

3. Hayotdavrinigtuzilishinibelgilovchiasosiystandar tnominiayting.

4. Hayotdavristrukturasi nechta jarayon guruhiga asoslangan.

5. Mustaqil (loyihaganisbatan)

ekspertlarningkorxonafaoliyatiningqabulqilingantalablar, rejalarvashartnomashartlarigamuvofiqliginianiqlashishlarinimadebyur itiladi.

6. Strukturalitahlilvaloyihalashtexnikasigaasoslanganmetodolog iyanominiayting.

7. Kichikloyihalaralgorimlariniyozishusullarinechta.

8. BPMNdiagrammalarningvazifasinimadaniborat.

9. AlgoritmlarningblokdiagrammalariniqurishqandayGOSTdast uriyhujjatlarningyagonatizimibilantartibgasolinadi.


10. Ma’lumotlarxemalarinimauchunkerak.

11. Hisobotshaklinitayyorlashvachopetishuchunnimanibilashkera kvanimauchun.



12. Jadvallarning relyatsion bog‘lanishidegandan imani tushunasiz.
13. Yangi hisobot loyihasini yaratish uchun qanday amallar bajariladi.
14. Hisobot konstruktorining elementlarini sanab bering.
15. Hisobotdagi dalga asoslangan ma’lumotlarni chiqarish uchun qaysi elementdagi foydalanish maqsadga muvofiq.

### 3.7. Foydalanuvchi interfeysini loyihalash.

 Foydalanuvchi interfeysini loyihalash usullari, talablari, ma’lumotlarni eksport qilish vositalari va komponentalari, .NET texnologiyasining hisobotni yaratish konstruktori, testlash va instruksiya yozish usullari, sinovchixodim xususiyatlarini bo‘yicha nazariy bilimlarni keltirilgan.. Bilimlarni mustahkamlash uchun 15 tanazariysavol berilgan.

*✍* *Kalitso‘zlar.* Interfeys, UI, WUI, amaliy dastur, ma’lumotlarni kiritish, PUI, integratsiya, o‘zaro bog‘lanish, xisobot, testlash.

*📄* *Bilish shart bo‘lgan tushunchalar.* Operatsion tizim va amaliy dasturlar daishlash, sinf va sinf ob’ekti, xususiyat, hodisa, forma, komponent dasturlashga oid dastlabki tushunchalar hamda C++ tilini qo‘llab quvvatlovchi muhit daishlashni bilish lozim.

*👤* *Bilibolasiz.* Visual C++ muhitidagi tuzilgan dasturlarning foydalanuvchi interfeysini loyihalash usullari, talablari, ma’lumotlarni eksport qilish vositalari va komponentalari, .NET texnologiyasining hisobotni yaratish konstruktori, testlash va instruksiya yozish usullari, sinovchixodimga qo‘yiladigan talablarni o‘rganishingiz mumkin.

#### REJA

1. Foydalanuvchi interfeysini loyihalash usullari.
2. Ma’lumotlarni eksport qilish vositalari va komponentalari.
3. Testlash va instruksiya yozish usullari.

Inson-mashina interfeysi (IMI) keng tushunchadir. Inson-operator va ular boshqaradigan mashinalar bilan o‘zaro aloqasita‘minlashning muhandislikechilarni o‘z ichiga oladi. Inson-mashina interfeysitizimlarini yaratishda yaqindan ergonomika tushunchalari bilan bog‘liq. IMI loyihalashtirish quyidagilarni o‘z ichiga oladi:

- ish joyini yaratish: ishchilarni maydon va boshqaruvi paneli, qurilmalar va boshqaruvi larni joylashtirish, ish joyi va ehtimol ranglar uyg‘unligi.

- operatorning barcha boshqaruv bilan o'zaro aloqasini ta'minlash hisoblanadi:

ularning mavjudligi vazarur harakatlari, kirish samaradorligi va tezligi, mustahkamligi, nazorat harakatlarining (izchilligi), joylashgan o'rin , displeylar va ulardagi teglar hajmi (bularning barchasi mavjudligi).

Inson -

kompyutero'zaro hamkorligi ilmiy ko'pyo'nalishlik kontekstida kompyuter grafika, muhandislik psixologiyasi, ergonomika, tashkilot nazariyasi, kognitiv fan, informatika va boshqalarni o'z ichiga oladi.

Foydalanuvchi interfeysi -

bu foydalanuvchi kompyuter bilan o'zaro aloqa qilish imkonini beruvchi dasturiy va apparat vositalar majmuasidir.

Bu o'zaro muloqotlarda dialoglarga asoslanadi. Bu holda dialog deganda inson va kompyutero'rtasida Real vaqt da amalga oshiriladigan va muayyan vazifa birgalikda haletishga qaratilgan axborot almashuvitushuniladi. Axborot almashuvish va harakatlarni muvofiqlashtirish har bir dialog foydalanuvchi va kompyuterni fizik ulaydigan alohida i/o jarayonlardan iborat. Xabarlarni uzatish orqali axborot almashiladi va nazorat signallari. Xabar-dialog almashuvida ishtirok etuvchi axborot qismidir. Ular quyidagilardir:

- odam tomonidan kiritish vositalari yordamida hosil qilinadigan kiritish xabarlari: klaviatura, manipulyatorlar, sichqonchak va boshqalar;

- kompyuter tomonidan matnlar, audio signallar va/yoki tasvirlar ko'rishida hosil qilinadigan va foydalanuvchiga monitor

ekranida yoki boshqa chiqish qurilmalarida ko'rsatiladigan chiqish xabarlari.

Asosan foydalanuvchi quyidagi turdagi xabarlarini hosil qiladi:

axborot so'rovi, yordam chiso'rovi, operatsiya yoki funksiya so'rovi, ma'lumot kiritish yoki o'zgartirish, shablon maydonini tanlash va boshqalar. Bunga javoban, u qabul qiladi: maslahatlari yoki yordam, javob talab qilmaydigan axborot xabarlari, harakatni talab qiladigan so'rovlar, javob kerak bo'lgan xato xabarlarini va shablon formatidagi o'zgarishlar.

Dasturlashga protsessual va ob'ektga yo'naltirilgan yondashuvlar bilan taqqoslaganda,

interfeysni ishlab chiqishga protsessual yo'naltirilgan va ob'ektga yo'naltirilgan yondashuvlar mavjud. Protsedura ga yo'naltirilgan interfeyslar

"protsedura" va "operatsiya" tushunchalariga asoslangan foydalanuvchi o'zaro aloqasining anaviy mo

delidanfoydalanadi. Bu model doirasidadasturiyta 'minotfoydalanuvchigategishlima' lumotlarnivauning natijasini foydalanuvchianiqlaydiganma' lumamallarnibajarishimkoniyatini beradi.

Ob'ektyo 'naltirilgan interfeyslarni foydalanuvchi o'zarobiroz boshqa chamodelini foydalanib, manipulyatsiyadomenob'ektlariniqaratishim mumkin.

Zamonaviy realliklardan farqli o'laroq, birinchi kompyuterlarga grafik foydalanuvchi interfeyslari juda zaif bo'lgan. Shuning uchun, eng boshida, odamlar faqat buyruq qatoridan foydalanishgan (CLI yoki buyruq qatorni interfeysi), unda buyruqlar so'rovlari yordamida belgilangan edi. Keyinchalik, bu TU interfeyslarni ishlab, endi operatsion tizimlari o'rnatish jarayonida ishlatiladi. Kompyuterlarning mavjudligi foydalanuvchilarga qulay interfeysni ishlab chiqish zarurligiga olib keladi.

Grafik foydalanuvchi interfeysi - bu kompyuterning tobora ortib borayotgan ishlab chiqish bilan birga mustahkam o'rnatilgan interfeysturi. Yaqin kelajakda, odamlar nutq yordamida kompyuter bilan o'zaro imkonini beradi foydalanuvchi audio interfeyslarni (VUI yoki ovozli foydalanuvchi interfeysi) bo'lishi mumkin.

Turli kompyuter o'yinlar tabiiy foydalanuvchi interfeysi (NUI yoki tabiiy foydalanuvchi interfeysi) yaratib beradi. Uning tizimi insonning harakatlarini tahlil qiladi va ularni o'yindagi harakatlarga aylantiradi. Perseptual foydalanuvchi interfeysi (PUI) va aqlli-kompyuter interfeysi (BCI yoki brain-kompyuter interfeysi) hozirda ishlab chiqilmoqda. eng so'nggi rivojlanish odamlarga kompyuterlarni fikrikuch bilan boshqaris hqobiliyatini ta'minlashga qaratilgan.

Web-ilovalarning foydalanuvchi interfeysi ham katta auditoriyada kelib chiqqan holda o'zigacha xususiyatlarga ega. Notiqlik haqida, notiqlik-ritorika haqida shunday fan mavjud. Fikrlash insoniyat bora bora, bu mavzu bo'yicha birinchi ishlar Empedokllarga (taxminan miloddan avvalgi v asr) taalluqlidir. Ritorika dan bilamizki, hozirgi odamlar sonining ortishi bilan tomoshabinlarning intellektual darajasi nika maytirish qonuni mavjud. Auditoriya qanchalik katta bo'lsa,

nutqshunchaliksoddabo‘lishi,  
iboralarshunchalikqisqabo‘lishivanutqdagipauzalarshunchalikmuhimaha  
miyatkasbetadi. Boshqatomondan,

ijtimoiytarmoqlarningriwojlanishibizgamuloqotmutlaqoog‘zakiemasligin  
iko‘rsatadivafoydalanuvchiinterfeysiorqalihammuloqotqilishingiz mumki  
n. Yuqoridagilardankelibchiqqanholda, veb-

dasturinterfeysisoddabo‘lishikerakligimantiqankelibchiqadi.

Ilojiborichaoddiy.

Aytaylik,

ma’lumbirkorxonaichidaishlashuchunmo‘ljallangandasturuchuninterfeys  
ishlabchiqyapsiz. Shubilanbirga,

sizhardoimtizimningkelajakdagifoydalanuvchilariuchuno‘quvkurslarinit  
ashkilqilishingiz,

ekrandanimaborligini vaishlabchiquvchisifatidanimademoqchiekanliging  
izniaytishingiz mumkin.

FI dizayniko‘plabyondashuvlarnio‘zichigaoladi: kaskadli, ichki,  
tashqi, iterativvaboshqalar.

Yondashuvlarningharbiriichidabirqatorusullarhammavjud.

Yuqoridarajadagiyondashuvdanqat‘iynazar,

yuqoridarajadagivapastdarajadagio‘zaroaloqalarniloyihalasho‘zigaxosus  
ullarnitalabqiladi. Ob’ektgayo‘naltirilganyondashuvtasvirlangan,

garchiko‘plabdizaynbosqichlariuchunan‘anaviyriwojlanishusullaribilanal  
mashtirilishim mumkin.

UshbuyondashuvyordamidaolingandizaynnatijalariFIgayo‘naltirilganilo  
valarnihujjatlashtirish, modellashtirish, prototiplash,  
baholashvaamalgaoishirishuchunfoydalidir.

Nostandartyondashuv.

YuqoridarajadagiloyihatizimibilanFIvafoydalanuvchio‘zaro standart  
xattihaqidao‘ylashmumkin, lekintafsilotlarqo‘shiladi, deb,

ustandartierishishuchunyanadaqiyinbo‘ladi —

vabirvaqtningo‘zidaizchil-qo‘llashturliqismlarigaajratiladi.

Tafsilotlarodatdakompromisslarvakutilmaganhodisalarbilanbohliqdir .

Foydaliqoida. Butasodifanyokiloyihagamuvofiqsodirbo‘lsa,  
nostandartxattiharakatlariqo‘llashkerak.

Yangilangangrafik.

Dastlabkigrafikimkoniyatlarvavizuallashtirishmasalalaridastlabkiloyihal  
ashbosqichlaridaishlabchiqiladi,

shuninguchunbatafsilloiyihalashbugrafikimkoniyatlarnianiqlashnitalabqil  
adi

(ba‘zangrafikadanfoydalanishdastlabkiloyihalashtushunchalaridantashqa

rigachiqadi).

Bundantashqari,

bubosqichdaekranlarvaxabarlarningyakuniybatafsiltartibitalabqilinadi.

Javobvaqti. Og'irtahlilyokisimulyatsiyaorqali, muhimvaengtez-tezishlatiladigandasturiyta'minotjoylariuchunjavobvaqtsmetasiniolishkerak.

Oxirgifoydalanuvchitomonidanamalgaoshirilganharbifunksiyauchunehtimoliyjavobvaqtigaasoslanib,

dasturiyta'minotuchunmosteskarialoqamexanizmiishlabchiqiladi.

Interfeysyaratishningasosiytamoyillari

1. Tabiiylik (intuitiv).

Tizim bilan ishlash vazifasini hal qilish jarayonini boshqarish uchun kerakli dasturlar (interfeys elementlarini)

topishda foydalanuvchiga qiyinchilik tug'dirmasligi kerak.

2. Mustahkamlik.

Agar foydalanuvchi tizim bilan ishlayotgan paytda tizimning biror qismi bilan ishlash uchun ayrim metodlardan foydalangan bo'lsa,

u holda metodlar tizimning boshqa qismidabir xil bo'lishi kerak. Shuningdek, interfeys orqali tizim bilan ishlash o'rnatilgan, doimiy, qulay me'yorlarga (masalan, Enter klavishi dan foydalanish) mos kelishi kerak.

3. Ortiqchalilik.

Bu foydalanuvchi faoliyat yoki tizimni boshqarish uchun faqat minimal ma'lumot kiritish kerak degan ma'noni anglatadi.

Masalan, foydalanuvchi ahamiyatsiz sonlarni kiritmasligi kerak (00010 o'rniga 10).

Xuddishunday,

foydalanuvchi oldindan kiritilgan yoki avtomatik ravishda tizimdan olinishi mumkin bo'lgan ma'lumotlarni kiritishni talab qila olmaydi.

Axborot kiritish jarayonini kamaytirish uchun iloji boricha standart qiymatlardan foydalanish tavsiya etiladi.

4. Yordam tizimi ga to'g'ridan-to'g'ri kiritish.

Ish vaqtida tizim foydalanuvchiga kerakli ko'rsatmalar berishi kerak bo'ladi.

Yordam tizimi uchta asosiy jihatta javob beradi:

taqdim etilgan buyruqlar sifat va miqdori;

xato xabarlarining tabiati va tizimning nima qilayotganini tasdiqlash.

Xato xabarlarini foydalanuvchi uchun foydalivatushunarlibo'lishi kerak.

Shubilanbirga,

busavollarga foydalanuvchilar uchun imkon qadarmuloyim vatushunarlibo'lgan tilda javob berishi kerak.

5. Moslashuvchanlik.

Tizim interfeysi turli darajadagi tayyorgarlikka ega foydalanuvchilarga xizmat

atqilish kerak.

Tajribasiz foydalanuvchilar uchun interfeysi yerarxik menyu strukturasi vataj ribalif foydalanuvchilar uchun buyruqlar, kalit kombinatsiyalar va parametrlar sifatida tashkilotilishi mumkin.

Sayt moslashuvchanligi

foydalanuvchilar osonlik bilan saytidamanipulyatsiyadan ortiq chara qamibil anbezovta holda zarar ma'lumotlarni topishimkonini beradi va sayt materiallari tashkilotilish hisoblanadi.

Maxsus tadqiqotlarga ko'ra,

o'rtacha bir foydalanuvchi saytda o'ttiz etti soniyasarf laydiva bu vaqt ichida ul arka mdan-kamsahifaning oxiriga o'tadi.

Shu munosabat bilan,

sayt samimiyma'lumotlarni chi qishitashkilotilish sohasidazamonaviy standartl argajavob bermasa,

agar,

bunday sayt targ'ib qilish,

barcha sa'y harakatlari kerakli maqsadga erishish mumkin emas,

chunki foydalanuvchi saytidazarur ma'lumotlarni topib bo'lmasa,

ular shuncha ko'proq samimiy boshqaresur ga o'tadi.

Saytning eng samarali moslashuvchanligi ta'minlash Internetda axborot taqdim etish xususiyatlarini bilishni talab qiladi.

Foydalanuvchilar,

birinchi navbatda, ularning ko'zoldidamoddiy ko'rish muhimdir.

Shuning uchun interfeysning maksimal soddaligi va qulayligi Internetdama'lumotlarni tashkil qilishning asosiy tamoyilidir.

Saytda oddiy va samimiy interfeysni yaratish uchun mavjudlikning asosiy tamoyillarini amalga oshirish gaimkon beruvchi texnikalardan foydalanish kerak.

Butun sayt uchun navigatsiya umumiy bo'lishi kerak,

sayt logoyu qorichap burchagidajoylashtirilgan bo'lishi kerak va logobosish boshaifa olib kelishi kerak va veb-saytida header va alt bilgide aloqama'lumotlarni joylashtirilgan kerak.

Saytda malakali qidiruvni tashkilotilish saytning mavjudligini ta'minlashning muhim element hisoblanadi.

Qidiruv tugmasi, logotip dantashqari,

sayt uchun ikkinchi eng muhim element

hisoblanadi va u har qandaysaytda foydalanuvchi uchun birinchi yordamchidir.

Foydalanuvchi birinchi 2 soniya ichida butugmanitopsakamida,

ular xavfsiz hisvasayt tomon keskinlik ularning darajasikamayadi.

Moslashuvchanlik sinov o'tkazish paytida ta'qib qilinishi kerak va uning ma'lum bosqichlaribor. Sinov o'tkazishning mohiyati quyidagicha:

1. Muammolarni aniqlash;
2. Gipotezani hosil qilish;
3. Test uchun o'lchov aniqlash;
4. Belgilar va test senariysini aniqlash;
5. Tanlang respondents;

6. Anketalarni to'ldirish;
7. Kirish brifingini o'tkazish;
8. Mavjudligini to'ldirish;
9. Intervyu respondentlari;
10. Natijalarni tahlil qilish;
11. Saytdizayni uchun talablarni aniqlash.

Umuman olganda, fani ishlab chiqish sohasida foydalanuvchilarning imkoniyatlarini yaxshilashlari inobatga olinishi kerak.

**Ma'lumotlarni eksport qilish vositalari va komponentalari.** Zamonaviy dasturlar foydalanuvchini qiziq tiradigan ma'lumotlarga kirishning ko'p usullarini ta'minlaydi.

Ma'lumotlar bilan ishlash uchun ixtisoslashgan dastur lovalari ishlab chiqilgan. Microsoft Office kabi umumiy ofis vositalari bilan integrallashgan ma'lumotlarni chiqarish imkoniyatlarini o'z ichiga oladi.

Foydalanuvchi hozirda ma'lumotlar bazasidan kerak ma'lumotlarni chiqarib eksport qilish imkonin.

Biroq, turli xil ma'lumotlar bilan ishlashning eng keng tarqalgan vatan ish usulian'anaviy hisobot shaklidir. Birtomondan, ko'p foydalanuvchilar, ayniqsa, halihisobot standart qog'oz shaklidan voz kechishga kuch topolmaydi. Boshqa tomondan, hisobot, asosan, turli xil xususiyatlarga ega bo'lgan ma'lumotlarni birlashtiruvchi vositadir.

Hisobot berish funksiyalarini ta'minlash talab ko'pincha zamonaviy dasturiy mahsulotlarni ishlab chiqishda paydobo'ladi. Data beacon Toolkit sifatida dastur plugin-smaylik va hisobot qo'llab-quvvatlash vositalari va komponentalari mavjud.

Shunga o'xshash vositalar Microsoft Visual Studio .net (VS.NET)dir. Microsoft Visual Studio .net (CRVS) uchun Crystal Reports deb nomlangan, biznes ob'ektlarini ishlab chiqish va litsenziyalash sxemasini bo'yicha etkazib berishga yo'naltirilgan. Yanabir vositasi, Microsoft hisobot xizmatlari (MSRS), Microsoft SQL Server DBMS asoslangan platforma hisoblanadi.

Crystal Reports for Microsoft Visual Studio .NET (Hisobotlarni qurish instrumenti). Crystal Decisions mahsulotlarini Microsoft development tools bilan bog'lashtirishi 1993-yil gaborib taqaladi. O'shanda Crystal Decisions hisobotlarning tarkibiy qismlari avval Visual Basic 3.0 data qidimetilgan edi. 1996 Crystal hisobotlari dasturi Visual Studio

development tools paketining bir qismigakiritildi va 2002 dauning bir qismibo'ldi VS.NET va. net Framework, rasmiynomi CRV Sqabul qilindi. Crystal Reports komponentlarining paydobo'lishi va ularnio'zarori vujudga kelingan va rinalarima vujud. Masalan, VS.NET 2003 VS.NET 2005.

CRV Skomponentlarining asosiy maqsadidasturning hisobot funksiyalarini ishlab chiqishni qo'llab-quvvatlashdir.

Bu maxsus hisobot shablon talablariga muvofiq ishlab chiqilgan bo'ladi.

Hisobot sahifasidama'lumotlarni joylashtirishni belgilovchi ript kengaytmasidan foydalanib, shablonda qaysima'lumotlartuzilmalari, hisobot ma'lumotlari uchun manbabo'lib xizmat qilishiko'rsatilgan bo'lishi kerak.

Keyinchalik shablon manbada ma'lumotlarni chiqarib, yakuniy hisobotning tuzilishini shakllantirish uchun foydalanadigan CRV Skomponentlarini kiritish gao'tiladi.

Hisobotni ko'rsatish uchun ishlab chiqilayotgan dasturga ko'shilgan maxsus funksiyalardan foydalanib web va Windows ilovalarini ham qo'llab-quvvatlanadi.

Aslida,

VS.NET da ekspert jarayoni kodini bir nechta satrlarni chaqirib kamayadi, lekin " hisobot dizayner " yordamida amalga oshiriladi. Shunday qilib, hisobot shablonini yaratish uchun Microsoft.

net platformasida dasturlashingiz shart emas (MS.NET). xohlasangiz, uni o'zgartirishimkonini beruvchi,

dastur hisobot shablonini o'zichiga olishim mumkin. Bujuda qulay, chunki bu holda, hardim ishlab chiqilgan dasturni qo'llab-quvvatlash uchun amalga oshirishim mumkin.

Yuqorida aytib o'tilganidek,

CRV Skomponentlarini vebgayo'naltirilgan va muntazam Windows ilovalarini ishlab chiqish uchun ishlatilishim mumkin.

Ma'lumotlar bilan ishlash nuqtainazaridan Crystal hisobotlari komponentlari ikki rejimda ishlash - surish modeli (Push model) va tortish modelida (Pull model) ishlatilishim mumkin.

Surish modeli - oraliq ma'lumotlartuzilmasi bilan ishlash modeli.

Shu bilan birga hisobotlar uchun ma'lumotlar odatiy takrorlanish ko'rinishida hosil qilinadi .NET Framework - ma'lumotlar bazasida bataladigan ma'lumotlartuzilmasidir.

Dastur hisobotlarni qurish uchun kiritish ma'lumotlarni tayyorlash barchama ntiqamallarga bog'liq. Bu qulay,

chunki ma'lumotlar faqat ma'lumotlar manbaiga bevosita kirish yo'libilanim



as, balkimutlaqobiron-birtarzdahosilqilinishimumkin. Biroq, agarhisobotyetarlichakattabo'lsa, judanoqulaytuzilmanito'ldirishingizkerak.

Tortishmodeli - talabbo'yichama'lumotlarniso'rash, ya'nima'lumotlarbazasidanolish. Buholda, Crystal Reports komponentlarigao'tganhisobotshablonmanbadanma'lumotlarniolishbilan bog'liqishlarniamalgaoshiradi.

Ushbuyondashuvko'psahifalihisobotlarninamoyishqilishuchunjudasamar ali, chunkiko'rsatilgansahifalaruchunfaqatma'lumotlarmanbadanolinadi.

Hisobotniishlabchiqarishvaaksettirishdantashqari, CRVSkomponentlariolinganma'lumotlarniPDF, HTML, XLS, RTFvaDOC formatlarigaeksportqilishniqo'llab-quvvatlaydi. Buholda, dasturfayltizimikatalogdabevositazarurhisobotfayllarniyaratishmumkin. Buhisobotfayllarniyaratishningommaviyrejimiuchunjudaqulay, foydalanuvchikeyinchaliknatijalarnimanfaatdortomonlargauzatishuchun ularnioldindanko'rmasdanhisobotlarguruhiniyaratishkerak.

CRVSnin gafzalliklarivakamchiliklariniqisqachaeslatibo'tamiz. Shubhasizafzalligi. net Framework vahamyaqinintegratsiyabo'ladiVS.NET, qaysiilovatuzuvchirivojlanishvositalaribilanhisobotyaratishimkoninibera di. Crvskomponentlarigaasoslanganechimlar, shuningdek, buxususiyathisobotlarva Microsoft .NET. arxitekturasi dajoylashganligisababli, murakkabvainteraktivhisobotlarniamalgaoshirishqiyinemas, chunkiularniinteraktivnazoratelementlarikabi "kublar" CRVSDamavjud.

Ushbumahsulot Microsoft texnologiyalarioldingiavlod (u Win32 va C++yordamidaishlabchiqilgan) bo'yichabutlovchiyadrodayashirinamalgaoshirishgamo'ljallangan, uningishlabchiqaruvchihisoblanadi.

Dinamikkutubxonalarako'rsatkichlarnio'rnatishvadasturiyjudaoddiyeksport hisobotlarniyordamidaamalgaoshiriladi. Shubilanbirga, yadroishlaydiganbundayplatforma

.netvaan'anaviyxotiraalgoritmdinamikyig'ish, raqobatkabinostandartinfratuzilmamuammolariolibkelishimumkin.

Bumuammonihalqilishyo'lidaharakatqilib, ishlabchiqaruvchi 1.1 versiyasiqismisifatidaerkinfoydalanishgaberilganva

.netplatformabosqichma-bosqicho'tishniboshlaganVS.NET 2003). Buo'zgartirish Crystal

hisobotlarkomponentlarinifoydalanadivadasturbilanfaqat.

netma'lumotlarni ekspert qilib beruvchi tomonidan tarqatish jarayonini soddalashtirish kerak. Crvsning platforma ga to'liq o'tish kutilmoqda.

CRVSkomponentlarining arxitekturasini yakuniy hisobotlashlabchi qarish jarayonlarini ko'pdarajali taqsimlashning ssenariysini tashkil etishga imkon berishi ga qarabmasdan, ushbu mahsulot butun hisobotni bosh qarish siklini tashkil qilish uchun mos emas.

MSRS platformasini umumiy hal qilish uchun mo'ljallangan vatayoreks pertfayllarini hosil qilishga qaratilgan.

Testlash va instruktsiya yozish usullari. Dasturiy ta'minot testi - nuqsonlarni aniqlash va mahsulot sifatini yaxshilash maqsadida dasturiy ta'minotni vategishli hujjatlarni tahlil qilish jarayonidir.

Dasturiy ta'minotlashlabchi qisho'nyillarda davomida, turli xil yo'llar bilan sinov sifatiga folatiorqalijudayaqinlashib qilindi. Uning bir necha asosiy "sinov davrlari" bor.

O'tgan asrning 50-60-yillarida sinov jarayoni keng rasmiylashtirilgan, bevositadasturiy ta'minotlashlabchi qish jarayonidan ajralgan va "matematika lashtirilgan" edi. Aslida, test ko'proq disk raskad rovkadasturlarika bibo'lgan. "Mukammal testing 4" - deb atalmish birtushuncha bo'rdiki, barcha mumkin bo'lgan kiritish ma'lumotlar bilan kodni ijrosi barcha mumkin bo'lgan yo'llarini tekshirish amalga oshirilardi.

Kiritish ma'lumotlar soni juda katta, chunki, mukammal sinov mumkin emas, va buyondashuv bilan hujjatlardagi muammolarni topish qiyin bo'lgan.

70 yillari sinov kiasosiy g'oyalarasosidabo'lgan: birinchi muayyan belgilangan sharoitlarda (ijobiy testing) dastur faoliyatini isbotlash jarayoni sifatida qabul qilindi va keyin - qat'iy qarabmas - qarshi: dastur muayyan belgilangan sharoitlarda (salbiy testing) inoperablekanligini isbotlash jarayoni sifatida qaraladi. Nafaqat bu ichki ziddiyat vaqt davomida g'oyib bo'lmaydi, lekin bugungi kunda ko'pmualliflar juda to'g'risinov kito'ldiruvchimaqsad bo'lgan deb hisoblaydi.

U aniqlangan muammolarga ko'zyumish garux sat bermaydi.

80-yillarda, dasturiy ta'minotlashlabchi qish sinov joyda asosiy o'zgarish yuz berdi: loyihayaratish yakuniy bosqichlaridabiri, test butun rivojlanish sikli davomida tatbiq etildi (dasturiy lifecycle).

Bundantashqari,  
oldindanvaularningyuzagakeladiganvaziyatlarnioldiniolishuchunemas,  
balkifaqatko'phollardaruxsatberilgan  
"dasturiyta'minotishlabchiqishmodellari"  
iterativortibdasturiyta'minotishlabchiqishmodelitavsifiniko'rishmaqсадg  
amuvofiq.

90-yillarda,  
testlashjarayonibutundasturiyta'minotishlabchiqishsiklio'zichigaoladiva  
mavjudsinovhollardavasinovmuhitlarqo'llab-quvvatlash, rejalashtirish,  
loyihalash, yaratishvasinovishlarniamalgaoshirishjarayonlarinita'sir  
"sifatkafolati"  
debnomlanganyanadakengqamrovlijarayonikabisinovo'tishboredi.  
Sinovsifatjihatifidanyangidarajagaetdi,  
butabiiyravishdauslubiyotningyanadarivojlanishiga,  
sinovjarayoniniboshqarishuchunjudakuchlivositalarvahozirgiavlodlarigaj  
udao'xshashbo'lgansinovlarniavtomatlashtirishvositalarinipaydobo'lishi  
gaolibkeldi.

Testlashningboshidaharqandaymutaxassis (tester  
hambundanmustasnoemas) ijrochivatalaba.  
Sinovholatlarivanuqsonhisobotlariqandayekanliginiyaxshitushunish,  
talablarnio'qiyolish,  
birnechtavositalaridanfoydalanishvajamoadayaxshiiishtiroketishkifoya.  
Asta-  
sekinsinaluvchiloyihaniishlabchiqishningbarchabosqichlarigakirishboshl  
anadi, buesao'znavbatidaularniyanadato'laroqanglatadi,  
ulardannafaqatfaolfoydalanaboshlaydi,  
balkiloyihahujjatlariniishlabchiqib,  
yanadamas'uliyatliqarorlarqabulqilaboshlaydi.

Testlashningasosiymaqsadinimajoziyma'nodaifodallasangiz,  
shundaybo'ladi: loyihahozirgivaqtdanimagamuhtojligini,  
loyihabuzarurnio'zo'lchovidaqabulqiladimi, yo'qmi,  
qandayqilibvaziyatniyaxshiroqo'zgartirishkerakliginitushuntiribberish.  
Loyihamenejeriningmaqсадigao'xshaydi, to'g'rimi? Sogi-hidori.  
huMuayyanrivojlanishdarajasidanboshlab, ITmutaxassislari,  
faqattexnikko'nikmalarto'plamlaridavabuko'nikmalarniqo'llashningasos  
iysohasidafarqqiladi.

Sinovchisifatidaishlashnimuvaffaqiyatliboshlashuchunqandaytexni  
kko'nikmalarkerak?

1. Chettillarko‘nikmalari. Ha, butexnikbo‘lmaganmahorat. Buniaksiomadebhisoblashingiz mumkin.

2. .Haqiqiyriyojlanganfoydalanuvchidarajasidaishonchlikkompyuter qobiliyatlarivabusohadadoimorivojlanishkuzatibborish.

3. Dasturlashtirish.

ButartibdabirinchinavbatdaharqandayITshaxsvadasturiyta‘minot tester hayotini osonlashtiradi. Dasturlash bilim siz test qilish mumkinmi? Ha, mumkin. Bu, albatta, yaxshiamalga oshirilishi mumkin.

4. Ma‘lumotlar bazalarivaSQLtili.

Buyer date testlovchi ham torma taxassislar darajasidamalakaga egabo‘lishital abetilmaydi, lekin eng ko‘ptarqalgan DBMLar bilan ishlashning minimal ko‘nikmalariva oddiy so‘rovlarni yozish qobiliyatimajburiy deb hisoblanishi mumkin.

5. Tarmoqlar va operatsion tizimlar qanday ishlashi haqidatushuncha. Hech bo‘lmasa, muammonitashxislashvailojibo‘lsa, unio‘zingiz hal qilishimkoniniberuvchi minimal darajada.

6. Veb-ilovalarva mobil ilovalar tamoyillari haqidatushuncha. Shukunlardadeyarli hammanarsabunday ilovalar shakli dayoziladivategishl itexnologiyalarnitushunishsamaralisinovuchunmajburiybo‘ladi.

Xulosasifatidatestlovchigatezdaajoyibmutaxassisbo‘lishimkoniniberuvch ishaxsiysifatlarni hamqaydetamiz:

1) mas‘uliyat va ishlashning ortishi;

2) yaxshimuloqotqobiliyatlari, o‘z fikrlarini aniq, tez, aniqifodalashqobiliyati;

3) sabr-toqat, qat‘iyat, tafsilotga e‘tibor, kuzatish;

4) yaxshimavhumva analitik fikrlash;

5) nostandart tajribalar o‘rnatishqobiliyati, tadqiqot uchun imkoniyat.

Bu fazilatlar gategalik qiladigan odamni topish qiyin, ammo o‘z-o‘zini rivojlantirish uchun ma‘lum birmosyozuvlarnuqtasiga egabo‘lish hard oimfoydalidir.

**NAZARIY BILIMLARNI TEKSHIRISH UCHUN SAVOLLAR.**

1.



- Insonvashinaningo‘zaromuloqotinita‘minlashnimadebaytiladi.
2. Inson kompyutero‘zarohamkorligiilmiyko‘pyo‘nalishlarinisanabbering.
  3. Foydalanuvchiinterfeysinima?
  4. Realvaqtdaamalgaoshiriladiganvamuayyanvazifanibirgalikdahaleti shgaqaratilganaxborotalmashuvinimadebaytiladi.
  5. Kiritishvositalariyordamidahosilqilinadigankirishxabarlariningtexnikvositalarinisanibbering.
  6. Interfeysyaratishningasosiytamoyillarisanabbering.
  7. Ma’lumotlarnieksportqilishvositalarivakomponentalaridegandanimanazardatutiladi.
  8. Turlixilxususiyatlargaegabo‘lganma’lumotlarnibirlashtiruvchivosita – bunima.
  9. Crystal Reports for Microsoft Visual Studio .NET tizimihaqidanimalarnibilasiz.
  10. Surishmodeliningvazifasi
  11. Tortishmodeliningvazifasinimadaniborat.
  12. Testlashdegandanimatushuniladi.
  13. Testlashningrivojlanishbosqichlarinisanabbering.
  14. Ikkiyo‘nalishlitestlashjarayoniqaysibosqichgategishli.
  15. Zamonaviytestlashningqulayliklarivakamchiliklari.
  16. Sinovchixodimbo‘lishningxususiyatlarisanabbering?

## Foydalanilgan adabiyotlar

1. Nazirov Sh.A., Qobulov R.V., Bobojanov M.R., Raxmanov Q.S. S va C++ tili. “Voris- nashriyot” MCHJ, Toshkent 2013, 488 b.
2. Horstmann, Cay S. C++ for everyone/Cay S. Horstmann. Printed in the United States of America - 2nd ed. 2010. – P. 562.
3. Horton I.-Beginning Visual C++ 2012/ I.Horton. Published simultaneously in Canada. –2012. –P. 988.
4. Алгоритмы. Справочник с примерами на C, C++, Java и Python. Джордж Хайнемай, Гэри Поллис, Стэнли Селков. Москва «Альфа-Книга», 2017. –С254.
5. ASP.NET MVC Framework с примерами на C# для профессионалов. Стивен Сандерсон. Москва 2010. –С. 562.
6. Самоучитель Microsoft Visual Studio C++ и MFC. Сидорина Т.Л. Санкт-Петербург. «БХВ-Петербург» 2009. –С. 600.
7. Си Шарп: Создание приложения для Windows. В.В. Лабор. Минск, Харвест, 2003.
8. Полный справочник по C++. Герберт Шилдт. Москва, 2006.
9. Современные технологии разработки программ, взаимодействующих с базами данных. С.А. Минеев, Ю.Е. Чуманкин. Нежный Новгород, 2018.
10. C/C++. Программирование на языке высокого уровня. Т.А.Павловская. Санкт-Петербург, 2003.
11. Самоучитель C++. Герберт Шилдт. Санкт-Петербург. 2003.
12. C/C++ и MS Visual C++ 2010 для начинающих. Санкт-Петербург. 2011.
13. Программирование под Windows в среде Visual C++ 2005. М.В. Свиркин, А.С. Чуркин. Санкт-Петербург. 2008.
14. Простая графическая задача на Microsoft Visual C++ с использованием библиотеки MFC. Е.П. Дербакова. Москва 2015.
15. Программирование в Visual C++ с использованием библиотеки MFC. В.В. Васильчиков. Ярославль 2006.
16. MS Visual C++ 2010 в среде .NET. В.В. Зиборов. Санкт-Петербург, 2012
17. Herbert Schildt, Java the complete reference ninth edition, oracle press, 2014.
18. Алексеев А.П. Информатика. 2001. М., СОЛОН-Р, 2001, 364 с.
19. Арипов М.М. Internet ва электрон почта асослари. –Т.: Университет, 2000. -132 б.

20. Страуструп. Б. Язык программирования С++. Специальное издание.-М.:ООО «Бином-Пресс», 2006.-1104 с.
21. Вирт Н. Алгоритмы + структуры данных = программа.- М.:Мир,1985.-405с.
22. Говорухин В., Цибулин В. Компьютер в математическом исследовании. Maple, Matlab, Latex и др. Учебный курс. Питер. 2001. – 624 с.
23. Информатика. Базовой курс. Учебник для Вузов., Санк-Петербург, 2001. под редакцией С.В.Симоновича.
24. Павловская Т.С. Щупак Ю.С. С/С++. Структурное программирование. Практикум.-СПб.: Питер,2002-240с
25. Павловская Т.С. Щупак Ю.С. С++. Объектно-ориентированное программирование. Практикум.- СПб.: Питер,2005-265с
26. Подбельский В.В. Язык СИ++.- М.; Финансы и статистика-2003 562с.
27. Романчик В.С., Люлькин А.Е. Программирование в С++ BUILDER. Учебное пособие. Мн.: БГУ, 2007. –126 с.
28. Юлдашев У.Ю., Боқиев Р.Р., Зоқирова Ф.М. Информатика. – Т.: F.Фулом номидаги нашриёт-матбаа ижодий уйи, 2002. - 237 б.
29. Abduqodirov A.A., Nayitov A.G', Shodiyev R.R. Axborot texnologiyalari // Akademik litsey va kasb hunar kollejlari uchun darslik. -Т.: O'qituvchi, 2003. –152 b.
30. Axmedov A.B., Tayloqov N.I. Informatika. Akademik litsey va kasb hunar kollejlari uchun darslik.-Т.: O'zbekiston, 2001. 2 – nashri. – 272 b.
31. Mo'minov B.B. Microsoft Excel bo'yicha amaliy mashg'ulotlar va ularni bajarish tartibi. Uslubiy qo'llanma. – Buxoro: Ziyo Rizograf, 2008. – 72 b.
32. Mo'minov B.B. Microsoft Excel dasturi bo'yicha laboratoriya ishlar to'plami. Uslubiy qo'llanma. – Buxoro: Ziyo Rizograf, 2008. – 92 b.
33. Mo'minov B.B. Pedagogik dasturiy ta'minot yaratish texnologiyasi. Monografiya. –Buxoro, Buxoro nariyoti, 2010. -168 b.
34. Xaldjigitov A.A., Madraximov Sh.F., Adamboev U.E. Informatika va programmash. O'quv qo'llanma. O'zMU, 2005 yil, 145 bet.
35. <http://aut.researchgateway.ac.nz/index.jsp> Auckland University of Technology digital library

36. <http://dastur.uz> Kompyuter Dasturlari va Kompyuterda Dasturlashga oid Forum, Habar va Yangiliklar
37. <http://google.com> Google qidiruv tizimi
38. <http://informatika.sch880.ru> Основы логики
39. <http://lex.uz/> O‘zbekiston Respublikasi qonun hujjatlari ma’lumotlari milliy bazasi
40. <http://ru.wikipedia.org/wiki> Википедия — свободная энциклопедия
41. <http://uza.uz> O‘zbekiston milliy axborot agentligi
42. <http://winedt.com> winedt rasmiy portali
43. <http://www.aci.uz> O‘zbekiston aloqa va axborotlashtirish agentligi
44. <http://www.cplusplus.com> cplusplus.com
45. <http://www.edu.uz> O‘zbekiston Respublikasi Oliy va O‘rta maxsus ta’lim vazirligi
46. <http://www.exponenta.ru> Образовательный математический сайт Exponenta.ru
47. <http://www.intuit.ru> Интернет-Университет Информационных Технологий
48. <http://ziyonet.uz> Oliy va O‘rta maxsus ta’lim vazirligi huzuridagi axborot ta’lim portali



## MUNDARIJA

<b>KIRISH.....</b>	<b>3</b>
<b>1-BOB. KONTEYNERLAR .....</b>	<b>4</b>
1.1. Noma'lum tiplar va noma'lum nomlar fazosi. ....	4
1.2. Konteynerlar (Kollektsiyalar).....	67
1.3. Assosiativ va tartiblanmagan assosiativ konteynerlar .....	98
1.4.Konteynerlarning adapterlari. ....	160
<b>2-BOB. DASTURLASH TILINING TAKOMILLASHTIRILGAN IMKONIYATLARI .....</b>	<b>219</b>
2.1. Standart algoritmlar va iteratorlar. ....	219
2.2. Sonli sinflar bilan ishlash .....	266
2.3. Sintaktik tahlil .....	313
2.4. Murakkabsaralash algoritmlari. Amaliy dasturlash .....	355
<b>3-BOB. VISUAL C++ MUHITIDA DASTURLASH. ....</b>	<b>434</b>
3.1. Visual C++ muhitida dasturlash. ....	434
3.2. Komponentalar bilan ishlash. ....	476
3.3. Muloqot oynalari bilan ishlash. ....	506
3.4. Visual C++ning grafik imkoniyatlari.....	538
3.5. OLE, MFC texnologiyalari. Ko'p oynali muhitlar bilan ishlash.....	603
3.6.Kichik loyihalarni yaratish .....	616
3.7.Foydalanuvchi interfeysini loyihalash. ....	632
<b>FOYDALANILGANADABIYOTLAR.....</b>	<b>645</b>

**MO'MINOV Bahodir Boltaeyich**

e-mail: [Mbbahodir@gmail.com](mailto:Mbbahodir@gmail.com)

# **DASTURLASH II**

**DARSLIK**

O'zbek tilida

**Toshkent – «NIHOL PRINT» OK – 2021**

Muharrir: A.Tog'ayev  
Tex. muharrir: F.Tog'ayeva  
Musavvir: B.Esanov  
Musahhiha: O.Muxammadiyeva  
Kompyuterda  
sahifalovchi: G.Tog'ayeva

9323



№ 7439-765f-47f1-7ea1-a683-4648-1314.  
Bosishga ruxsat etildi: . Bichimi 60x841 /16.  
Shartli bosma tabog'i 38,00. Nashr bosma tabog'i 37,75.  
Adadi 100. Buyurtma № 54.

«Nihol print» Ok da chop etildi.  
Toshkent sh., M. Ashrafiy ko'chasi, 99/101.