

**MUHAMMAD AL-XORAZMIY NOMIDAGI
TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI**

**NAZIROVA ELMIRA SHODMONOVNA
PARDAYEVA GULMIRA ABDUNAZAROVNA**

MOBIL ILOVALARINI ISHLAB CHIQISH

**fanidan
O‘QUV QO‘LLANMA**

**5330500 - Kompyuter injiniringi (“Kompyuter injiniring”, “AT-Servis”,
“Multimedia texnologiyalari”)**

Toshkent – 2022

Mualliflar:

Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti KIF “MT” kafedrasini mudiri t.f.d., dos. E.Sh.Nazirova

Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti Qarshi filiali TTKTF “ATT” kafedrasini katta o’qituvchisi G.A.Pardayeva

“Mobil ilovalarini ishlab chiqish” fanidan oquv qo‘llanma. /TATU Toshkent 2022.

Ushbu o‘quv qo‘llanma Mobil ilovalarini ishlab chiqishga bag‘ishlangan. Har bir mavzu zarur nazariy, ko‘plab misollar va rasmlarni o‘z ichiga oladi. Mobil ilovalarini ishlab chiqish bilan tajribani shakllantirish uchun nazariy material amaliy vazifalar bilan birga keladi.

E.Sh.Nazirova va G.A.Pardayeva tomonidan tayyorlangan “Mobil ilovalarini ishlab chiqish” qo‘llanmasi qo‘lyozmasi universitetning 5330500 - Kompyuter injiniringi (“Kompyuter injiniring”, “AT-Servis”, “Multimedia texnologiyalari”) bakalavriat talabalari uchun mo‘ljallangan.

Taqrizchilar:

Zaynidinov H. – Muhammad al-Xorazmiy nomidagi TATU t.f.d., professor, “Axborot texnologiyalari” kafedrasini mudiri

Sevinov J.U. – Islom Karimov nomidagi Toshkent Davlat texnika universiteti, t.f.d., dotsent, “Axborotlarga ishlov berish va boshqarish tizimlari” kafedrasini mudiri

TATU ilmiy-uslubiy Kengashida ko‘rib chiqildi va chop etishga tavsiya etildi.

KIRISH

Tez rivojlanib borayotgan kompyuter axborot texnologiyalari bizning kundalik hayotimizning barcha jabhalarida sezilarli o'zgarishlarni olib kirmoqda. Barchamizga sir emaski, hozirgi vaqtda jamiyatni axborot kommunikasiya texnologiyalarsiz tasavvur qilib bo'lmaydi. Umuman olganda, har bir mamlakatni rivojlanishini zamonaviy axborot-kommunikasiya texnologiyalarini sohalarga joriy qilinishi bilan baholash mumkin. Mamlakatimizda xam buning uchun bir qator ishlar amalga oshirilib kelinmoqda. Xususan "O'zbekiston Respublikasi Axborot Texnologiyalari va Kommunikasiyalarini Rivojlantirish Vazirligini tashkil etish to'g'risida"gi (04.02.2015 yil PF-4702) O'zbekiston Respublikasining Prezidentining Farmonida yangi tashkil etilgan vazirlikning asosiy vazifalaridan biri qilib "raqobatdosh dasturiy mahsulotlarning mamlakatimizda ishlab chiqarilishini va ichki bozorini hamda ularga ko'rsatiladigan xizmatlarni rivojlantirishga ko'maklashish va uning muvofiqlashtirilishini ta'minlash, iqtisodiyotning real sektori tarmoqlarida va iste'molchilarda zamonaviy dasturiy mahsulotlar, axborot tizimlari va axborot resurslarini joriy etish" etib belgilangan.

Ma'lumki, hozirgi kunda dasturiy ta'minot yoki dasturiy mahsulotni yaratish dasturiy ta'minot ishlab chiquvchi mutaxassislar jamoasi orqali amalga oshiriladi. Umuman olganda, hozirda, har bir dasturiy ta'minot yoki dasturiy mahsulotni ishlab chiqishga amaliy loyiha sifatida qarash mumkin. Chunki, har bir ishlab chiqilgan dasturiy ta'minot biror bir sohadagi (ijtimoiy, iqtisodiy, sanoat, xalq xo'jaligi va boshqalar) ma'lum bir muammolarni hal qilishga yo'naltirilgan bo'ladi.

Mobil ilova (inglizcha: Mobile app) — ma'lum bir platforma (iOS, Android, Windows Phone va boshqalar) uchun ishlab chiqilgan smartfonlar, planshetlar va boshqa mobil qurilmalarda ishlashga mo'ljallangan dastur. Ko'pgina mobil ilovalar qurilmaning o'zida oldindan o'rnatiladi yoki ularni App

Store, Google Play va boshqalar kabi onlayn dastur do'konlaridan bepul yoki pullik bilan yuklab olish mumkin.

Dastlab, mobil ilovalar elektron pochtni tezda tekshirish uchun ishlatilgan, ammo ularning yuqori talablari boshqa sohalarda ham kengayishiga olib keldi, masalan, mobil telefon va GPS o'yinlari, suhbatlashish, video tomosha qilish va Internetdan foydalanish.

Har bir abonent (uyali telefon apparati) muayyan bir uyali telefon kompaniyasining telefon tarmog'idagi baza stansiyaga bog'lanadi. har qaysi baza stansiyada "uyalar" bo'ladi (nomi shundan). Har bir uyaga bir necha kanal biriktirib qo'yiladi; tarmoqning kattakichikligiga qarab uyalar soni har xil bo'ladi. Mobil telefon apparati qaysi kompaniya tarmog'ining baza stansiyasiga ulanganligiga qarab uning aloqa bog'lash doirasi chegarasi har xil bo'ladi. Ba'zi abonentlar faqat bir shahar doirasida aloqa bog'lay olsa, ba'zilari boshqa shaharlar va hatto boshqa mamlakatlar bilan ham aloqa bog'lay oladi.

Dastlab XX-asr 80-yillarida qo'l telefon aloqa, asosan, analog texnologiyalarga asoslangan, keyinroq Mobil telefonda aloqa tizimi sof raqamli tizimga o'tgan. Yevropa mamlakatlarining deyarli har birida o'z milliy tarmog'i ishlab chiqilgan (masalan, Fransiyada Radiocom 2000). Mobil telefonga doir xalqaro hamkorlik dastlab ko'chma radioaloqa tizimi masalalari bo'yicha nemisfransuz tadqiqotlari natijasi sifatida pochta va kommunikatsiyalar ma'muriy xodimlarining Yevropa konfederatsiyasi SERT (Conference des administrations Europcennes des Postes et Telecommunications) rahbarligida tashkil etildi. 1982 yil bu internet 26 Yevropa mamlakatini, aniqrog'i pochta, telefon va telegraf aloqa RTT (Post, Telephone and Telegraph) ma'muriy muassasalarini birlashtirdi. Shu tariqa Yevropa konfederatsiyasi "Mobil aloqani ishlab chiqish bo'yicha maxsus guruh" (GSM) ni tuzdi. Keyinchalik u "Mobil aloqa global tizimi" deb atala boshladi. Yevropa hududida bu tizim 1993 yil arafasida to'liq faoliyat ko'rsata boshladi. GSM tizimi boshidanoq xalqaro ko'lamdagi tizim sifatida ishlab

chiqilgan. Barcha qit'alardagi ko'p mamlakatlar unga qo'shilgan. Mobil telefon tizimi inson hayotining barcha sohalariga kirib keldi.

I BOB. MOBIL ILOVALARNING ASOSIY TUSHUNCHALARI

O'quv qo'llanmaning ushbu bobi mobil ilovalarning rivojlanish tarixi. Dastlabki mobil qurilmalar va mobil ilovalar. Zamonaviy mobil OT lar. Mobil ilovalarni yaratish tillari va texnologiyalari: iOS va Android uchun dasturlash, emulyatorlar. Mobil ilovalarni joylash uchun platformalar: Play Market, Apple Store internet do'konlari. Mobil ilova ishlab chiqish tamoyillari, talablarni o'rganish va texnik topshiriqni shakllantirish, platformani tanlash, Mobil ilova dizaynini ishlab chiqish, UI/UX dizaynga doir umumiy tushunchalar batafsil bayon etiladi.

1.1. Mobil ilovalarning rivojlanish tarixi. Dastlabki mobil qurilmalar va mobil ilovalar. Zamonaviy mobil OT lar.

Reja:

1. Mobil qurilmalarning rivojlanish tarixi
2. Dastlabki mobil qurilmalar va mobil ilovalar
3. Mobil qurilmalarning operatsion tizimlar platformasi

Birinchi telefon ixtirosi. Telefon tarixi

Tarix g'ildiraklari o'zi bilan kecha, bugun, ertaga tushunchalari va ularning mazmun-mohiyatini jamlab, ilgari aylanib boraveradi. Ba'zan bizga vaqt juda tez odimlab ketayotgandek, rejalashtirgan ishlarimizni amalga oshirishga ulgurish imkonini bermayotgandek tuyuladi. Aslida, ana shu kecha, bugun, ertaga deb nomlagan kunlarimizning har biri tariximiz sahifasini boyitadi, inson hayotini umid bilan ilgari odimlashga undaydi.

Ayrim voqea va hodisalar esa tarix zarvaraqlarida o'chmas iz qoldiradi. Ular haqidagi ma'lumotlar tarixiy yodgorlik va obidalar, kitoblar hamda muzeylarda kelajak avlodlar uchun saqlanadi. Aloqa sohasining rivojlanish tarixida bo'lib o'tgan shunday yorqin xotiralar Toshkent shahridagi Aloqa tarixi muzeyi

zallaridan o‘rin olgan. Demak, avvalgi sonlardagi mavzuimizni davomi sifatida muzey eksponatlarini o‘rganish uchun muzey zallarini tomosha qilamiz.

XIX asrning so‘nggi yillarida simli telegrafning takomillashib, rivojlanishi bilan telefon ham ixtiro etildi. 60-yillarda I. F. Reys telefon apparatini loyihalashtirdi va u tezda amaliyotda qo‘llanila boshladi. Keyinchalik telefon rivoji amerikalik ixtirochilar I. Grey (1835–1901) va A. G. Bell (1847–1922)lar nomi bilan bog‘liq bo‘ldi. Bell uzatuvchi va qabul qiluvchi elektromagnit telefonini taqdim etganda unga katta qiziqish bilan qarab «Bell telefon kom-paniyasi»ning tashkil etilishiga ko‘maklashdilar. Natijada u yirik konsernga aylanib ketdi.

1878-yilda D. E. Yuz o‘zi a‘zo bo‘lgan London qirollik jamiyatiga mikrofon ixtiro etganligini ma‘lum qildi. Yuz sifatsiz elektr ulanishlarni tadqiq eta turib, bunday ulanishda, telefonda shovqin eshutilishini aniqladi. Turli materiallardan tayyorlangan ulanishlarni tekshirib, presslangan ko‘mirdan foydalanilganda katta samara berilishini aniqladi. Yuz mazkur natijalarga tayanib, 1877-yilda mikrofon deb nomlangan telefon uzatkichni taqdim etdi. «Bell kompaniyasi» Yuzning yangi ixtirosini o‘zining dastlabki apparatida mavjud bo‘lmagan detal sifatida ishlab chiqara boshladi. Telefonning takomillashuvi ustida (V. Simens, Ader, Gover, Shteker, Dolbir va boshqa) ko‘plab ixtirochilar mehnat qildilar. Edison boshqa turdagi telefon apparatini loyihalashtirdi (1878). U ilk bor telefon apparati sxemasiga induksion katushka va presslangan ko‘mir qurumili mikrofonni kiritdi va uzoq masofalarga ovozni uzatishni ta‘minladi.

Bellning ixtirosi telefoniya asrini boshlab berdi. «Telefoniya» atamasi ancha keng ma‘noga ega bo‘lib, telefon aloqasining barcha ilmiy-texnik jihatlarini qamrab oldi. 1876-yildan keyin telefon aloqasi abonentlar sonining va tarmoq bo‘yicha uzatiladigan axborot hajmining o‘sib borishi bilan eng ommabop aloqa vositasiga aylandi. Telefon aloqasining bunday ahamiyati uning boshqa texnika vositalariga nisbatan shaxsiy muloqot o‘rnatishni samarali ta‘minlashi bo‘lib hisoblanadi. Mazkur aloqa vositasi orqali: telefon xabari bir vaqtda ma‘no-

mazmunli axborot (matn)ni uzatadi, xabar bilan birga, gapirayotgan odamning individual belgilari va emotsional bo‘yoqlari bilan yetkaza oladi.

O‘tgan 125 yil mobaynida telefoniya Bell taklif etgan qo‘shni binodagi shaxslar bilan muloqot qilishga asoslangan oddiy elektromagnit telefondan videotelefoniga va hozirgi global telefon aloqasi tarmog‘iga aylangunga qadar, katta yo‘lni bosib o‘tdi. Keyinchalik Bellning ixtirosi keng qo‘llanila olmasligi ma’lum bo‘ldi. Shu sababli ham, 1878-yilda Nyu-Xeyven (AQSH) shahrida abonentlarga xizmat qiluvchi dastlabki telefon stansiyasidan foydalanila boshladi. Ke-yinchalik telefon aloqasida muloqot zonasini kengaytirib borish zarurati tug‘ila boshladi. Shu tariqa, telefon tarmog‘ida ko‘p bosqichli struktura konsepsiyasi yuzaga keldi. Telefon tarmog‘i masshtabi kengligi va murakkabligidan qat’iy nazar, uch guruhga birlashtirish mumkin bo‘lgan elementlardan tashkil etiladi:

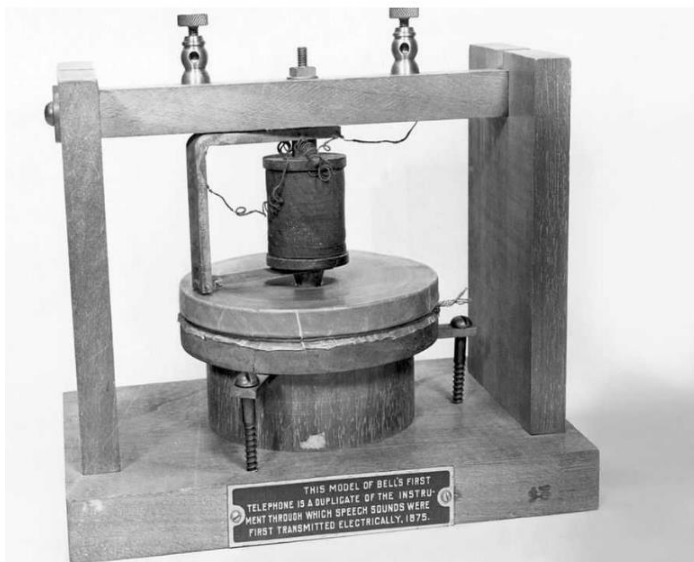
- abonent terminallari (odatda — telefon apparati);
- aloqa liniyasi (abonentlik va birlashtiruvchi liniyalar);
- kommutatsiya markazi va telefon stansiyalari.

Birinchi telefonni yaratuvchisi Bell A.G.

1876-yil 14-fevralda, shvetsiyalik amerikalik Aleksandr Graham Bell AQSh Patent idorasi tomonidan o'zi ishlab chiqargan qurilma ya'ni, telefonni chiqardi. Bell o'z qurilmasini 1876 yil 7 martda patentladi va shu bilan birga qurilma tezda tarqala boshladi. 1877 yilga kelib, Bostondan Massachusets shtatidagi Somervilgacha birinchi oddiy telefon liniyasining qurilishi yakunlandi. 1880 yil oxiriga kelib, Qo'shma Shtatlarda 47,900 telefon mavjud edi. Keyingi yili Rost-Aylenddagi Boston va Providens o'rtasida telefon aloqasi o'rnatildi. Nyu-York va Chikago o'rtasidagi xizmat 1892 yilda, Nyu-York va Boston o'rtasida esa 1894 yilda boshlangan. Transkontinental xizmat 1915 yilda boshlangan.

Bell o'zining Bell telefon kompaniyasini 1877 yilda tashkil qildi. Sanoat jadal kengayib borar ekan, Bell tezda raqiblarini sotib oldi. Bir qator qo'shilishlardan so'ng, bugungi AT&T kompaniyasining asoschisi bo'lgan Amerika Telefon va Telegraf kompaniyasi 1880 yilda birlashtirildi. Bell telefon tizimi orqasidagi

intellektual mulk va patentlarni nazorat qilganligi sababli, AT&T yosh sohada amalda monopoliyaga ega edi. 1984 yilgacha AQSh telefon bozori ustidan AQSh Adliya vazirligi bilan kelishuv AT&T kompaniyasini davlat bozorlarini nazorat qilishni tugatishga majbur qilgan paytgacha o'z nazoratini saqlab qoladi.



Telefon stansiyalarining rivojlanishi Dastlabki telefon stansiyasi 1877-yilda AQSH'da, venger muhandisi T. Pushkash (1845–1893) loyihasi asosida qurilgan. 1879-yilda telefon stansiyasi Parijda, 1881-yilda Berlinda, keyinroq Peterburg, Moskva, Odessa, Riga va Varshavada qurilgan. Telefon aloqasining keyingi rivoji uchun P. M. Golubitskiyning 1885-yildagi markaziy batareyadan elektr quvvat oluvchi telefon stansiyasi sxemasi muhim o'rin tutdi. P. M. Golubitskiy yuqori darajada sezgir telefon ixtiro etdi va telefon apparatni yaratdi. Uning ishlash tamoyili hozirgi zamonaviy apparatlarda ham saqlanib qoldi. 1883-yilda olim ko'mir kukuni bilan mikrofon yaratadi.

1887-yilda rus olimlari tomonidan «o'zi ishlaydigan markaziy kommutator» — avtomatik telefon stansiyasi (ATS) ga asos solindi. U zamonaviy mazmundagi ATS'larga o'xshamasa ham, stansiyada kommutatsiyalarni birlashtirish vazifasini

telefonist shaxs yordamisiz amalga oshirardi va abonentlarning o'zlari tomonidan boshqarilardi.

Shu tariqa, jahon bo'ylab rivojlana boshlagan telefon aloqasi mamlakatimizga ham kirib keldi. XIX asrda iqtisodning sezilarli darajada o'sishi, shaharlarda sanoat, bank-moliyaviy tashkilotlari, tijorat korxonalari faoliyat ko'rsata boshlashi o'lkada shahar va shaharlararo telefon aloqasining asta-sekin rivojlanishiga asos bo'ldi.

1887-yilda Toshkent pochta-telegraf idorasi o'rtasida telefon aloqasi tashkil etildi.

1891-yil iyulda Toshkent telefon tarmog'ini tashkil etish bo'yicha muzokaralar bo'lib o'tdi va 1891-yilda Toshkent shahrida umumiy foydalanish imkoniyatlarini beruvchi telefon tarmog'i tashkil etildi.

Dastlabki mobil qurilmalar va mobil ilovalar

Mobil telefon, uyali telefon yoki qo'l telefoni uyali aloqa tizimi bo'ylab ovoz va boshqa axborot uzatuvchi hamda qabul qiluvchi elektron qurilmadir. Zamonaviy mobil telefonlar internet mijozlari, o'yinlar, infraqizil va bluetooth portlar, video/fotokamera, MP3 player, radio va hokazo ta'minot bilan qurollangan.

Birinchi odatiy telefon almashinuvi 1878 yilda Konnektikut shtatining Nyu-Xeyven shahrida tashkil etilgan. Dastlabki telefonlar abonentlarga juft bo'lib ijaraga berilgan edi. Boshqa abonent bilan bog'lanish uchun abonent o'z liniyasini qo'yishi kerak edi. 1889 yilda Kanzas-Siti korxonasi rahbari Almon B. Stroger o'z o'rnini va slayderlaridan foydalanib, 100 ta chiziqdan biriga ulanadigan kalit ixtiro qildi. Ma'lum bo'lishicha, Stroger tugmachasi 100 yildan keyin ham ba'zi telefon idoralarida ishlatilgan.

Strogerga birinchi avtomat telefon stantsiyasi uchun 1891 yil 11 martda patent berildi. Stroger kalitidan foydalangan holda birinchi almashinuv 1892 yilda La-Porte shahrida (Indiana) ochilgan. Dastlab, abonentlar o'zlarining telefonlarida urish orqali kerakli miqdordagi impulslarni ishlab chiqarish uchun tugmachaga ega

bo'lishgan. Strovergerning sherigi 1896 yilda tugmachani almashtirib, aylantiruvchi qo'ng'iroqni ixtiro qildi. 1943 yilda Filadelfiya ikki tomonlama xizmatdan (aylanma va tugma) voz kechgan so'nggi yirik hudud edi.

To'lov telefonlari

1889 yilda tanga bilan ishlaydigan telefon Konnektikut shtatidagi Xartfordlik Uilyam Grey tomonidan patentlangan. Greyning to'lov telefoni birinchi bo'lib Xartford bankida o'rnatildi va ishlatildi. Bugungi kunda to'lov telefonlaridan farqli o'laroq, Grey telefonidan foydalanuvchilar qo'ng'iroqni tugatgandan so'ng to'lashdi. To'lov telefonlari Bell tizimi bilan bir qatorda ko'paymoqda. Birinchi telefon kabinalari 1905 yilda o'rnatilgan vaqtga kelib, AQShda 100 mingga yaqin taks telefonlari mavjud edi. 21-asrning oxiriga kelib, mamlakatda 2 milliondan ortiq taksilar mavjud edi. Ammo mobil texnologiyalar paydo bo'lishi bilan, pullik telefonlarga bo'lgan talab tezda pasayib ketdi va bugungi kunda Qo'shma Shtatlarda 300 000 dan kam odam ishlaydi.

Sensorli telefonlar

Western Electric, AT&T kompaniyasining ishlab chiqarish sho'ba korxonasi tadqiqotchilari 1940 yillarning boshlaridan beri telefon ulanishini boshlash uchun impulslardan emas, balki ohanglardan foydalanish bo'yicha tajriba o'tkazdilar. Biroq, 1963 yilgacha, nutq bilan bir xil chastotani ishlatadigan ikki tonli ko'p chastotali signalizatsiya tijorat uchun foydalidir. AT&T uni Touch-Tone terish sifatida taqdim etdi va bu tezda telefon texnologiyalaridagi navbatdagi standartga aylandi. 1990 yilga kelib, tugmachali tugmali telefonlar Amerika uylarida rotatsion terish modellariga qaraganda ko'proq tarqalgan edi.

Simsiz telefonlar

70-yillarda birinchi simsiz telefonlar paydo bo'ldi. 1986 yilda Federal aloqa komissiyasi simsiz telefonlar uchun 47 dan 49 MGts gacha bo'lgan chastota diapazonini berdi. Ko'proq chastota diapazonini berish simsiz telefonlarga kamroq shovqin va ishlash uchun kam quvvat talab qildi. 1990 yilda FCC simsiz telefonlar uchun 900 MGts chastota diapazonini berdi.

1994 yilda raqamli simsiz telefonlar va 1995 yilda raqamli tarqalish spektri (DSS) mos ravishda joriy etildi. Ikkala ishlab chiqish ham simsiz telefonlarning xavfsizligini oshirish va telefon suhbatini raqamli ravishda tarqalishini ta'minlab, kiruvchi eshitishlarni kamaytirishga qaratilgan edi. 1998 yilda FCC simsiz telefonlar uchun 2,4 GGts chastota diapazonini berdi; bugungi kunda yuqoriga ko'tarilish diapazoni 5,8 gigagertsni tashkil qiladi.

Uyali telefonlar

Eng qadimgi mobil telefonlar transport vositalariga mo'ljallangan radio tomonidan boshqariladigan qurilmalar edi. Ular qimmat va noqulay edi va juda cheklangan edi. 1946 yilda birinchi bo'lib AT&T tomonidan ishga tushirilgan ushbu tarmoq asta-sekin kengayib, yanada rivojlanib boradi, ammo u hech qachon keng qabul qilinmagan. 1980 yilga kelib, u birinchi uyali tarmoqlarga almashtirildi.

Bugungi kunda ishlatiladigan uyali aloqa tarmog'iga aylanishi bo'yicha tadqiqotlar 1947 yilda AT&T tadqiqot qanoti Bell Labs-da boshlandi. Kerakli radio chastotalar hali tijoratda mavjud bo'lmagan bo'lsa-da, telefonlarni "uyalar" yoki uzatgichlar tarmog'i orqali simsiz ulash tushunchasi hayotiy edi. Motorola 1973 yilda birinchi qo'lda ishlaydigan uyali telefonni taqdim etdi.

Telefon kitoblari

Birinchi telefon kitobi Konnektikut shtatining Nyu-Xeyven shahrida, Nyu-Xeyven okrugi telefon kompaniyasi tomonidan 1878 yil fevralda nashr etilgan. Bir sahifali bo'lib, 50 nomga ega edi; hech qanday raqamlar ko'rsatilmagan, chunki operator sizni bog'laydi. Sahifa to'rt qismga bo'lingan: turar joy, professional, asosiy xizmat va boshqa.

1886 yilda Reuben H. Donnelly Yellow Pages brendli birinchi katalogni yaratdi, unda firma nomlari va taqdim etilayotgan mahsulotlar va xizmatlar turlari bo'yicha tasniflangan firma nomlari va telefon raqamlari mavjud. 80-yillarga kelib, Bell tizimi yoki xususiy nashrlar tomonidan chiqarilgan telefon kitoblari deyarli har bir uy va biznesda bo'lgan. Ammo Internet va uyali telefonlar paydo bo'lishi bilan telefon kitoblari deyarli eskirgan edi.

Oddiy uyali telefonlardan tashqari, quyidagi turlar mavjud:

1. Kamerofon - kamera va / yoki videokamera vazifasini bajaradigan mobil telefonlar. Hozirgi vaqtda ushbu nom amalda asl ma'nosidan chiqib ketdi, chunki aksariyat zamonaviy qurilmalar o'rnatilgan foto / videokameralar bilan jihozlangan. Biroq, bu ko'pincha rivojlangan foto imkoniyatlariga ega multimedia telefonlari (xususan, yuqori sifatli kameralarga ega telefonlar) deb nomlanadi



2.2-rasm. Motorola DynaTAC - FCC tomonidan sertifikatlangan birinchi mobil telefon

2. Multimediya telefoni (pleer telefoni, musiqa telefoni, musiqiy telefon) - audio va videofayllarni ijro etish imkoniyatlari yaxshilangan ixtisoslashgan mobil telefonlar va boshqalar raqamli fonogrammalarni ijro etishi mumkin va xotira kartasi uyasi bilan jihozlangan. Shunga qaramay, bir qator multimedia telefonlarida audio imkoniyatlarga (masalan, o'rnatilgan pleyerga ega Sony Ericsson W seriyali (Walkman), Motorola ROKR seriyali; W, Nokia X seriyali, asosan byudjet telefonlari) ba'zi istisnalar).



2.3-rasm. Nokia 6680, 2005 yilgi smartfon namunasi

3. Smartfon - to'liq operatsion tizimga ega mobil telefon (Symbian OS, Windows Mobile, Palm OS, GNU / Linux, Android, Apple iOS, MeeGo va boshqalar). Bunday telefonlar ushbu operatsion tizim tomonidan qo'llab-quvvatlanadigan har qanday yangi dasturlarni o'rnatishga va ularning imkoniyatlarini kengaytirishga imkon beradi: IM mijozlari, ofis to'plamlari, tashkilotchilar, audio va video pleerlar, qo'ng'iroqlarni boshqarish dasturlari, brauzerlar va boshqalar. Shuningdek, smartfonlar uchun viruslar mavjud (imkoniyati mavjud bo'lsa ham) yopiq operatsion tizim tufayli halokatli kodni oddiy telefonlarga kiritish juda qiyin). Rus tilida bunday telefonlarning paydo bo'lishi bilan "dialer" jargonli atamasi bunday funktsiyalarga ega bo'lmagan telefonlarga yopishib qoldi.
4. Kommunikator - mobil telefon funktsiyalariga ega bo'lgan shaxsiy kompyuter (PDA). Smartfonlarga nisbatan mobil qurilmalarning boshqa tarmog'i, ammo ular o'rtasida yaqinlashishga intilish mavjud (hozirda ikkalasining ham funktsionalligi yaqinlashmoqda). Smartfonlar singari ular ham uchinchi tomon dasturlarini ishlab chiqish uchun ochiq operatsion tizimlarni boshqaradilar. Smartfonlar va kommunikatorlarning imkoniyatlari, har qanday "eski" kompyuterlar singari, o'rnatilgan dasturlar va jihozlarga bog'liq.

5. Ish telefoni - korporativ foydalanuvchilar uchun maxsus funktsiyalarga ega telefon. Bunday telefonlar sizga matnli hujjatlar va elektron jadvallarni ko'rishga, elektron pochta bilan ishlashga, tashkilotchi ma'lumotlarini korporativ server bilan sinxronlashtirishga va boshqalarga imkon beradi. Ish telefonlarining katta qismi smartfonlar yoki kommunikatorlar bo'lib, ko'pincha QWERTY-klaviaturali qurilmalar mavjud. Ba'zan katta funktsional imkoniyatlarga ega bo'lgan bunday telefonlarda kamera yo'q (xavfsizlik talablari yuqori bo'lgan tashkilotlar uchun).
6. Tasviriy telefon - telefonlari, ularning asosiy xususiyati zamonaviy ko'rinish va yoqimli funktsionallik (xususan, avtomatlashtirilgan ochilish). Moda telefonlari ko'pincha g'ayrioddiy form-faktorlar va zamonaviy (va qimmat) materiallardan foydalanadilar - bunday telefonlarni ishlab chiqarishda qimmatbaho metallar va qimmatbaho toshlar korpusni bezash uchun tez-tez ishlatiladi. Bunday qurilmalarning funktsionalligi odatda past, ammo istisno tariqasida ular orasida smartfonlar va kommunikatorlar mavjud. Vertu modellari moda telefonlariga misol bo'la oladi, garchi ular odatda ko'plab ishlab chiqaruvchilarning modellarida mavjud.
7. Bir martali ishlatiladigan telefon - asosiy funktsional imkoniyatlarga ega telefon (ko'pincha displeysiz va hattoki SIM-kartasiz, ba'zilari kredit karta formatida), balans tugaguniga qadar yoki batareyalar tugamaguncha qo'ng'iroq qilish uchun mo'ljallangan, keyin tashlanadi yoki yo'q qilinadi.
8. Keksalar uchun telefon (buvi, bobo) - asosiy funktsional imkoniyatlarga ega telefon, SOS tugmasi, katta tugmalar, displeyda katta bosma (ko'pincha monoxrom).

Mobil telefon, qo'l telefon — simsiz telefon turi. Har bir abonent (uyali telefon apparati) muayyan bir uyali telefon kompaniyasining telefon tarmog'idagi baza stansiyaga bog'lanadi. har qaysi baza stansiyada "uyalar" bo'ladi (nomi shundan). Har bir uyaga bir necha kanal birlashtirib qo'yiladi; tarmoqning kattakichikligiga qarab uyalar soni har xil bo'ladi. Mobil telefon apparati qaysi

kompaniya tarmog'ining baza st-yasiga ulanganligiga qarab uning aloqa bog'lash doirasi chegarasi har xil bo'ladi. Ba'zi abonentlar faqat bir shahar doirasida aloqa bog'lay olsa, ba'zilari boshqa shaharlar va hatto boshqa mamlakatlar bilan ham aloqa bog'lay oladi.

Dastlab 20-asr 80-yillarida qo'l telefon aloqa, asosan, analog texnologiyalarga asoslangan, keyinroq Mobil telefonda aloqa tizimi sof raqamli tizimga o'tgan. Yevropa mamlakatlarining deyarli har birida o'z milliy tarmog'i ishlab chiqilgan (masalan, Fransiyada Radiocom 2000). Mobil telefonga doir xalqaro hamkorlik dastlab ko'chma radioaloqa tizimi masalalari bo'yicha nemisfransuz tadqiqotlari natijasi sifatida pochta va kommunikatsiyalar ma'muriy xodimlarining Yevropa konfederatsiyasi SERT (Conference des administrations Europcennes des Postes et Telecommunications) rahbarligida tashkil etildi. 1982 yil bu internet 26 Yevropa mamlakatini, aniqrog'i pochta, telefon va telegraf aloqa RTT (Post, Telephone and Telegraph) ma'muriy muassasalarini birlashtirdi. Shu tariqa Yevropa konfederatsiyasi "Mobil aloqani ishlab chiqish bo'yicha maxsus guruh" (GSM) ni tuzdi. Keyinchalik u "Mobil aloqa global tizimi" deb atala boshladi. Yevropa hududida bu tizim 1993 yil arafasida to'liq faoliyat ko'rsata boshladi. GSM tizimi boshidanoq xalqaro ko'lamdagi tizim sifatida ishlab chiqilgan. Barcha qit'alardagi ko'p mamlakatlar unga qo'shilgan. Mobil telefon tizimi inson hayotining barcha sohalariga kirib keldi. GSM tarmoqlari abonentlarining soni dunyoda 200 mln. kishidan oshib ketdi (2005).

Dunyodagi birinchi sensorli telefon IBM kompyuter uskunalari ishlab chiqaruvchi Simon modeli edi. Gadget 1993 yilda chiqarilgan, garchi birinchi namunali telefon 1992 yilda e'lon qilingan bo'lsa ham. Qurilmaning og'irligi 500 grammni tashkil etdi, bu esa undan foydalanish va uzatishni qiyinlashtirdi. Formada ushbu qurilma g'ishtga o'xshardi. Qanday bo'lmasin, ushbu uyali telefon tarixdagi birinchi smartfon edi, chunki u operatsion tizimda ishlaydi.

Qurilma tizimiga o'rnatilgan asosiy ilovalar taqvim, kalkulyator, bloknot, manzillar kitobi, elektron pochta va o'yinlar edi. Telefonda to'plam bilan kelgan

stylus yordamida qo'lda yozilgan yozuvlarni yaratish mumkin edi. Matn QWERTY aqlli klaviaturasi yordamida kiritildi. Ma'lumotni saqlash uchun smartfon PCMCIA xotira kartasidan foydalanishi mumkin.

PCMCIA kartalari noutbuklarda keng qo'llanilib, zamonaviy flesh-disklarning prototipiga aylandi.

Shuni ta'kidlash kerakki, ushbu qurilma barmoqlaringiz bilan ishlashga mo'ljallanmagan va siz funktsiyalarni boshqarish uchun stylusdan foydalanishingiz mumkin. Ushbu qurilma insoniyat tomonidan kashf qilingan 50 eng muhim qurilmalar ro'yxatiga kiritilgan. Ushbu smartfonning narxi telekommunikatsiya operatoriga ulanish bilan 900 dollardan boshlanadi va o'z SIM-kartangizni o'rnatish imkoniyatisiz telefonni sotib olish uchun taxminan 1100 dollarni tashkil etadi.

Haqiqat shundaki, allaqachon iPhone-ni birinchi sensorli telefon deb hisoblaydigan odamlar avlodi ulg'aygan. U haqiqatan ham ko'p jihatdan birinchi bo'lib mobil aloqa bozorini butunlay o'zgartirdi. Bu, albatta, sensorli ekranli birinchi qulay smartfon edi. Va faqat barmoqlaringiz bilan boshqarish uchun juda moslashdi. Ammo 10 yil oldin birinchi iPhone texnologiyaning rivojlanishi, birinchi sinovlar va xatolar, birinchi "xitlar" bo'lmaganda edi bo'lmaydi. Sensorli telefonlar iPhone-ga qanday kelib chiqqanini eslaylik.

IBM Simon (1994)

iPhone-dan oldin Simon deb nomlanadigan qurilma ishlab chiqilgan. Aslida, bu dunyodagi birinchi birinchi sensorli telefon (va undan keyin iPhone'dan oldin ular juda ko'p edi).

"Simon" ning prototipi birinchi marta 1992 yilda namoyish etilgani. 1994 yilda ushbu qurilma sotila boshlandi va uning narxi 1090 dollarni tashkil etdi (inflyatsiyani hisobga olsak, endi u yanada qimmatga tushadi).

Simon mobil kompyuter va telefonning birinchi "simbiozlari" dan biri bo'lgan, keyinchalik bunday qurilmalar smartfonlar deb nomlangan. Bundan tashqari, u peyjer va faks mashinasi sifatida ishlatilishi mumkin. Unda manzillar

daftarchasi, daftar, vazifalar ro'yxati, elektron pochta mijosi, taqvim, kalkulyator bor edi, shuningdek bir nechta o'yinlarni o'z ichiga olgan.

Qurilma 4x7 dyuymli monoxrom displey bilan jihozlangan bo'lib, o'lchamlari 160x293 pikselga teng. Hech qanday jismoniy tugmachalar yo'q edi - terish ham, matn ham virtual klaviatura va stilus yordamida amalga oshirildi.

Simonda xotira kartalari uchun uyasi, 16 MGts chastotali protsessor, 1 MB tezkor xotira mavjud edi. Telefon maxsus mahkamlash stantsiyasida zaryadlangan. Batareya faqat 8-12 soat kutish vaqti va bir soat gaplashish / ma'lumot uzatish uchun xizmat qiladi. Lekin qo'shimcha haq evaziga batareyani sotib olish mumkin edi.

IBM Simon bozorda muvaffaqiyatsizlikka aylandi deb ayta olmaymiz. Yuqori narx ko'pchilikni qo'rqitdi. Qurilma 5 ming nusxadan oshiq nusxada chiqarilgan va bir vaqtning o'zida bu haqda tezda unutilib ketgan. Ammo bu mobil texnologiyalarning rivojlanish tarixida muhim qadam bo'ldi.

O'tkir shaxsiy aloqa kommunikatori MC-G1 (1997)

Simon bilan bo'lgan voqeadan 3 yil o'tgach, Sharp bozor pishgan deb qaror qildi va sensorli ekranli kommunikatorning versiyasini taqdim etdi. Afsuski, u deyarli e'tiborni jalb qilmadi. Endi uning rasmiy fotosuratini yuqori aniqlikda, shuningdek, tavsiflarning to'liq ro'yxatini topib bo'lmaydi.

Keyinchalik zamonaviy sensorli telefonlar

Sensorli ekran bilan chiqarilgan keyingi telefon, Sharp PMC-1 Smartfoni bo'lib, u yapon uskunalari ishlab chiqaruvchisi rivojlanishining mahsuli edi. U o'zidan oldingilariga qaraganda ancha kichikroq edi va o'sha paytda barcha funktsiyalar mavjud edi.

Smartfolar keyinchalik Nokia 9000-ning raqobatchisi sifatida ishlab chiqarildi, u tobora keng tarqalib, Finlyandiya mobil qurilmalari ishlab chiqaruvchi kompaniyasining birinchi ommaviy ishlab chiqargan smartfonlar seriyasiga aylandi.

2007 yilda barmoq yo'naltirilgan va Multi-touch texnologiyasiga ega sig'imli ekranga ega bo'lgan birinchi qurilmalar ommaviy ishlab chiqarishga kirdi. Ushbu funktsiya sizga qurilma interfeysini bir necha barmoq bilan boshqarishga imkon beradi, masalan, rasmni kattalashtirish yoki dasturlar bilan ishlash. Bunday qurilmalar birinchi iPhone va LG KE850 Prada edi. Sensorli displeyi tufayli u ko'plab mukofotlarga sazovor bo'ldi.

Kundan kunga mobil internet ulushi va mobil qurilmalar foydalanuvchilari soni ko'payib bormoqda. Bunday o'zgarishlar butunlay asosli – hozir hamma hayotning barcha jabhalarida qulayliklarga va mobillikga intilishmoqda. Aloqa vositalari va planshetlarning paydo bo'lishi bilan yo'lga, ish safariga, uchrashuvlarga (ish yuzasidan) va boshqa maqsadlar bo'yicha “noqulay” noutbuklarni yonida olib yurishga hojat qolmadi. Biroq mobil qurilmalar ba'zi maxsus qo'shimchalarsiz – mobil ilovalarsiz bunchalik foydasi katta bo'lmas edi. Ta'lim sohasida mobil ilovalarning qadri o'sib bormoqda bunga sabab esa u beradigan quyidagi qulayliklar:

- Dars paytida yoki darsdan tashqari paytda topshiriq ustida ishlayotgan o'quvchilarning hamkorlikdagi ishi;
- Fayllar almashish;
- Masofaviy ta'lim tashkiloti va ota-onalarning o'zaro harakati. Bu mobil ilovalarni qo'llashning bir necha aspekti xolos! Keling “mobil ta'lim” tushunchasiga bafurcha to'xtalamiz.

- **Mobil ta'lim (M-Learning)** – bu vaqt va makondan qat'iy nazar mobil qurilmalar orqali o'qish imkoniyatidir. Mobil ta'lim – bu joylashgan o'rindan yoki o'quvchilarning joylashuvining o'zgarishidan cheklanmay mobil qurilmalar orqali elektron o'qishdir.



Mobil ta'lim odatda “tarkib bo'yicha yengil” deb hisoblanadi, va ko'proq o'quvchilarga audiomaterillar yetkazib berishda, matnli xabarlar almashinuvida, so'rovlarda qatnashishda, matnli chatlarda, konspekt qilish va ko'rishda ishlatiladi. Shuningdek, mobil ta'lim tarkibiga quyidagi talablar qo'yiladi:

- **Ixchamlilik** – mobil ta'lim tarixiy qismlari aloqadagi ehtimolli uzilishlarni inobatga olgan holda davomiylik bo'yicha qisqa bo'lishi lozim
- **Mikroergonomikaning yuqori darajadali** – kichik ekranda ovoz va rasmlarning kichik hajmda va yuqori sifatli bo'lishi. Fayllarning kichik hajmdaligi yuklab olishning tezligini ta'minlaydi.
- **Har joydali** – mobil o'qitish tarkibidagi materillarni joylashgan joyidan qat'iy nazar hohlagan joyda olish mumkin. Mobil aloqa va mobil qurilmalar ta'minotchilarining ulkan qamrovi mobil ta'lim xizmatlaridan har joyda va siz uchun qulay paytda foydalanish imkoniyatini yaratib beradi.
- **Talab bo'yicha foydalanishga ruxsat** – tabiatan mobil qurilmalar o'quvchilar uchun talab bo'yicha zarur paytda o'zining maksimal yetkazib berish imkoniyatini ishga solib zaruriy tarkibdan foydalanishni ta'minlaydi.

Mobil qurilmalar	
* Mobile computers	• Mobil kompyuterlar

<ul style="list-style-type: none"> * Mobile Web * Smartphone, Tablet computer * Wearable computer * Calculator watch * Smartwatch * Head-mounted display * Personal digital assistant/enterprise digital assistant * Calculator * Handheld game console * Portable media player * Ultra-mobile PC * Digital still camera (DSC) * Digital video camera (DVC or digital camcorder) * Mobile phone * Smartphone, Feature phone * Pager * Personal navigation device (PND) 	<ul style="list-style-type: none"> • Mobil Internet • Smartfon, planshet kompyuter • Kiyiladigan kompyuter • Kalkulyator soati • Aqlli soat • Boshga o'rnatilgan displey • Shaxsiy raqamli yordamchi / korporativ raqamli yordamchi • Kalkulyator • Qo'lda ishlaydigan konsol • Portativ media pleer • Ultra mobil kompyuter • Raqamli fotoapparat (DSC) • Raqamli videokamera (DVC yoki raqamli videokamera) • Mobil telefon • Smartfon, Badiiy telefon • Peyjer • Shaxsiy navigatsiya qurilmasi (PND)
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Mobil qurilmalarning operatsion tizimlar platformasi

So'nggi bir necha yillardan beri, mobil divayslar sanoati shunchalik ulkanlashib kettiki, internetdan umumiy foydalanish va faollik bo'yicha hattoki kompyuterlar bozorini ham egallab bormoqda. Bu albatta mobil vositalardagi ustunlik, hamda ma'lumotlar tarmog'i bo'yicha muntazam aloqa tezligi sababidandir.

Xuddi kompyuterlar o'tmishidagi singari, mobil operatsion tizimlari (O.T.) ishlab chiqaruvchilari orasida ham yakka liderlik uchun kurash boshlanib ketdi. Eng talabgorlar ro'yhatidan iOS, Android va Windowslar turli sanoat ishtirokchilari tomonidan kiritilgan. Ushbu holat nafaqat kim eng kuchli ekanligi to'g'risidagi savol balki SIZ uchun qaysi biri to'g'ri kelishini bilish uchun ham muhimdir. Har bir O.T ba'zi foydalanuvchilarga mos keluvchi va ba'zilarini yuzini burishtiruvchi o'z xususiyatlariga ega.

Mobil operatsion tizim degan atamaning o'zi Nokia-ning sa'y-harakatlari bilan foydalanuvchilarning so'z birikmalariga kirdi. O'sha davr uchun Symbian-ga asoslangan va 2001 yil bo'lgan asbob chinakam rivojlangan va ilg'or deb topildi. Bir oz vaqt o'tgach, o'z operatsion tizimiga ega bo'lgan barcha telefonlar smartfonlar deb nomlana boshladi. Masalan, Android bugungi kunda eng mashhur tizimlardan biri hisoblanadi va uning orqasida milliarddan ortiq qurilmalar ishlaydi.

Endi Microsoft tomonidan chiqarilgan Windows Phone OS ham mashhur emas. U o'tmishdagi kabi talabga ega emas, ammo sodiq muxlislariga ega. Tez ishlash, barqarorlik va oddiy interfeys Windows Phone operatsion tizimi joylashgan uchta ustundir. Va u boshqa kompyuterlar uchun taniqli tizim asosida yig'ilganligini unutmang. Boshqa mashhur operatsion tizim - bu Apple-dan iOS. Asosiy diqqat uning yaqinligidadir. "Apple" dasturiy ta'minoti o'rtacha foydalanuvchi uchun mavjud emas [va viruslarga duch kelmaydi](#), bu esa ishlash va foydalanish uchun eng qulay sharoitlarni ta'minlaydi.

Shubhasiz, eng mashhur operatsion tizimlar haqida hech bo'lmaganda umumiy tasavvurga ega bo'lish, ularning xususiyatlarini bilish juda muhimdir. Shunday qilib, iOS eng qimmat va istiqbolli deb hisoblanadi. Bugungi kunda, hatto bola ham xuddi shu iPhone-ning ne'mati va imkoniyatlari haqida bilib olishi mumkin. Operatsion tizimning ishlashini keraksiz so'zlarsiz tushunishingiz mumkin, hamma narsa intuitiv ravishda aniq. Qulaylik to'lovi - dunyoning qolgan qismidan iOS-ning haddan tashqari chiqarib tashlanishi. Bu, ayniqsa, boshqa

shunga o'xshash gadjetlar bilan mumkin bo'lgan tarkibni ishlatish yoki uzatish paytida seziladi. Yuqori narxlar to'g'risida ham bir sabab bilan aytilgan: hamma ham Apple gadjetlariga qimmat narxlari tufayli pul bera olmaydi. [Internetdan emas](#), balki rasmiy do'kondan olingan qo'shimcha dasturlar ham pul talab qiladi. Barcha xarajatlar evaziga iOS shveytsariyalik soat darajasi, tezlik, zarba, inoyat va uslub sifatini taqdim etishga tayyor.

Android operatsion tizimida ishlaydigan qurilmalar ancha arzon deb hisoblanadi. Oldingi versiyadan farqli o'laroq, bu erda egasi hamma narsani o'zi uchun va eng kichik tafsilotlarda to'liq sozlashi mumkin. Ilovalarning aksariyati bepul, kutubxonada har qanday did uchun son-sanoqsiz dasturlar mavjud va smartfonlarning o'zi ham keng narx oralig'ida taqdim etilgan. Apple'dan kelgan smartfonlar uchun operatsion tizim singari, Android ham juda yaxshi va amaliy jihatdan ko'p jihatdan axborot oqimlari bilan ishlashga moslashtirilgan. Biroq, bularning barchasida virusni o'z shaxsiy sozlamalaringizda ushlab qolish yoki chalkashlik ehtimoli yuqori. Ba'zida hatto oddiy piktogrammalar ham, chalkashtirib yuborishi mumkin.

ANDROID

Ushbu tizimning eng o'ziga xos xislatlaridan biri - unda o'zlashtirish (customization) imkoniyatlari miqdoridir. Bu qulaylik telefonning [butun telefonning barcha qismi](#), o'rnatilgan fon, vidjetlar va ikonkalarni o'z ichiga oladi. Android Google Play yoki nashr qiluvchining o'zidan bevosita yuklab olinadigan, bir milliondan oshiq turli ilovalarga ega. Agar siz aynan o'zingizga moslashtirilgan mobil xizmatini istasangiz, Android eng yuqori tanlovdir.

Linux yadrosiga asoslanib, ochiq manbali operatsion tizim. Ochiqlikning oqibati - bu bepul yoki Internet orqali yuklab olinadigan juda ko'p sonli ilovalar. Bundan tashqari, ilovalar bilan yuqori tezlikni ta'minlaydi .



Bu hozirgi kunga qadar eng keng tarqalgan va mashhur operatsion tizim. Android-ning tarixi yaqinda boshlangan bo'lsa ham, 80% dan ko'prog'ida aynan u ishlaydi. Faqat 2008 yilda Kaliforniyada asos solingan Android Inc. Biroz vaqt o'tgach, uni Google qidiruv giganti sotib oldi.

Android-ning afzalliklari:

- Oddiy va intuitiv interfeys. Zamonaviy texnologiyalar olamiga yangi kelgan kishi shunchaki bunday OSga ega bo'lgan smartfonlarning asosiy funksiyalarini aniqlab beradi.
- Ochiq manba. Har qanday malakali dasturchi [ushbu tizim uchun ariza yozib](#), uni rasmiy Android do'koni - Play Market-ga joylashtirishi mumkin.
- Ma'lumotlarni uzatish qulayligi. Agar biron-bir ma'lumotni smartfondan kompyuterga, masalan, fotosuratlar, audio yoki video fayllarni uzatish zarurati bo'lsa, u holda USB ulanishi orqali hech qanday muammosiz amalga oshirilishi mumkin. Qo'shimcha dasturlarni o'rnatish yoki ular bilan shug'ullanishning hojati yo'q.
- Ishlash. Operatsion tizim juda tez. Har bir telefon modeli ishlab chiqaruvchi tomonidan muntazam ravishda yangilanib turadi, shuning uchun qurilma har doim zamonaviy talablarga javob beradi.
- Ko'p o'lchovli. OT barcha kerakli funksiyalarni bajarishga qodir, bu esa gadgetni qulay va amaliy sotib olishga imkon beradi.

- Narxi Android OS bilan yangi smartfonning narxi juda xilma-xil bo'lishi mumkin. Siz o'zingizning talablaringizga javob beradigan arzon smartfonni osongina topishingiz mumkin.

Androidning kamchiliklari:

- Ochiq manba. Siz ushbu elementni operatsion tizimning afzalliklari ro'yxatida ko'rgansiz, [ammo bu nafaqat fazilat](#), balki uning kamchiliklari hamdir. Shu sababli, Android-da ishlaydigan qurilmalar zararli dastur va xakerlarning hujumlariga juda himoyasiz deb hisoblanadi. Shuning uchun, sotib olgandan so'ng darhol gadgetingizni himoya qilish va unga yuqori sifatli antivirusni o'rnatish juda muhimdir.
- Katta trafik sarfi. Ko'pgina foydalanuvchilar Android smartfonlaridagi trafik bizning ko'zimiz oldida erib ketayotganidan shikoyat qilmoqdalar, chunki ko'plab dasturlar doimiy ravishda yangilanishni talab qiladi. Albatta, bu muammoni Internet-trafik iste'molini boshqarish orqali hal qilish mumkin.
- Yangilanishlarni kutish vaqti. Ko'pgina smartfonlar uchun OSni yangilash masalasi butunlay muammoga aylanadi, chunki eski modellar uchun umuman yangilanishlar bo'lmasligi mumkin, boshqalari uchun yangi versiya juda kech chiqishi mumkin.

IOS

Apple'ning iOS, iPhone'da 2007 yilda ishga tushirilgan bo'lib, mavjud eng birinchi mobil operatsion tizimlaridan biridir. Ushbu tizimdan foydalanish ortidagi asosiy g'oya -boshqa barcha Apple mobil mahsulotlaridan foydalana olish imkoni bo'ladi. Ma'lumki Apple mavjud eng zo'r telefonlarni ishlab chiqadi, shu bilan o'z operatsion tizimini ahamiyatini oshiradi. Undan tashqari iOS uchun mobil ilovalar

soni ulkandir, hayolingizga kelishi mumkin bo'lgan barcha dastur va xizmatlar muhayyo.



OS Apple tomonidan maxsus iPhone va iPad qurilmalari uchun ishlab chiqilgan. Ushbu operatsion tizimning bozor ulushi taxminan 14% ni tashkil qiladi.

IOS-ning afzalliklari:

- Xavfsizlik Ushbu OS uchun manba kodi yopiq, ya'ni "olma" korporatsiyasining muhandislari va dasturchilaridan tashqari hech kim unga kira olmaydi. Shunday qilib, kompaniya o'z OS-ni o'z imkoniyatlaridan ruxsatsiz foydalanish va virus hujumlaridan himoya qildi.
- Do'konlarning boy assortimenti. IOS - AppStore-ga ega smartfonlar uchun ilovalar do'konida 1 milliondan ortiq turli xil o'yin va dasturlar mavjud. Albatta, bu Android OS do'koniga qaraganda bir oz kamroq, ammo bu erda "olma" ning yopiq manbasini ko'rib chiqish muhimdir. Shuning uchun barcha ilovalar faqat professional dasturchilar tomonidan yozilgan va ishonchliligi va yuqori sifati bilan ajralib turadi.
- Siri modulining mavjudligi. iPhone egasida foydalanuvchi buyruqlarini samarali bajarish va hatto u bilan konstruktiv muloqotni amalga oshirishga qodir bo'lgan shaxsiy virtual yordamchi mavjud.
- Ishlash. Ushbu smartfonlar tezkor javob berish va barcha funktsiyalarning bajarilishi bilan ajralib turadi. [iPhone telefonlar](#), qoida tariqasida, osib qo'yishga qodir emas.

IOS kamchiliklari:

- Norasmiy dasturlarni o'rnatish Jailbreak operatsiyasini talab qiladi, uning ishlatilishi Apple tomonidan mutlaqo qo'llab-quvvatlanmaydi va iPhone-ni texnik qo'llab-quvvatlash va kafolat huquqidan mahrum qilish bilan tahdid qiladi.
- O'zingiz yoqtirgan qo'shiqni o'z gadjetingizga yuklash uchun siz maxsus iTunes dasturidan foydalanishingiz kerak, bu noqulay va biroz muammoli.

WINDOWS

Kompyuterlar orasida Windows eng hurmatga sazovor O.T. bo'lishiga qaramay, Window mobil muqobillari "to'yga so'nggi kelgan mehmon"ga o'xshab ketadi. Ijro bobida Windows Phone o'z raqiblari iOS va Androidga yaqin darajadagi ko'rsatkichga erishdi; biroq uning ilovalari miqdori sezilarli darajada kamchilikni tashkil etadi. Umuman olganda Windows O.T.ning ashaddiy muhlisi bo'lmasangiz, Android va iOS yaxshi tanlov bo'lib ko'rinishi mumkin.

Windows operatsion tizimi mobil telefonlarga 2010 yilda o'rnatilgan bo'lishiga qaramay, Nokia Lumia liniyasi (710, 800) ishga tushgandan keyingina haqiqiy mashhurlikka erisha boshladi. Vidjetlar o'rniga "jonli plitkalar" ishlatilgan, ular dasturlarni ochmasdan katta hajmdagi ma'lumotlarni namoyish etishgan. Bunday yangilik foydalanuvchilarga zerikarli va standart Android menyusini almashtirish uchun juda qiziq tuyuldi. Biroq, bir muncha vaqt o'tgach, foydalanuvchilarning ishtiyoqi biroz zaiflashdi. 2016 yilda Windows operatsion tizimi bilan sotilgan gadjetlarning ulushi atigi 2,5% edi.



Windows Phone-ning afzalliklari:

- Intuitiv va sodda interfeys.
- Marketplace orqali o'rnatiladigan ilovalarning ozgina "og'irligi".
- Xbox o'yin xizmatidan foydalanish imkoniyati.
- Microsoft Office o'rnatilgan dasturiy to'plam. Ko'pincha ushbu smartfonlar biznes maqsadlarida sotib olinadi. Smartfon orqali siz yangi hujjatlarni tahrirlashingiz va tuzishingiz mumkin. Hatto korporativ Outlook pochta ham mavjud.
- Ma'lumotni sinxronizatsiya qilish uchun qulay va tushunarli dastur mavjudligi.

Windows Phone-ning kamchiliklari:

- Ilovalarning kichik tanlovi. Ularning soni 300 **mingga yaqin**, bu Android va iOS-ga qaraganda ancha kam. Bundan tashqari, ularning aksariyati to'liq emas.
- Windows 7 bilan ishlaydigan smartfonlarning noqulayligi multimediya fayllarini gadgetning xotirasiga o'tkazish qiyinligi. Buning uchun iTunes dasturidan foydalaning - Zune.

Nazorat savollari

1. Mobil qurilmalarning rivojlanish tarixi haqida ma'lumot bering?
2. Dastlabki mobil qurilmalar va mobil ilovalar haqida ma'lumot bering?
3. Mobil qurilmalarning operatsion tizimlar platformasi deganda nimani tushunasiz?
4. Androidning afzallik va kamchiliklari haqida ma'lumot bering?
5. iOS ning afzallik va kamchiliklari haqida ma'lumot bering?
6. Windowsning afzallik va kamchiliklari haqida ma'lumot bering?

1.2. Mavzu: Mobil ilovalarni yaratish tillari va texnologiyalari: iOS va Android uchun dasturlash, emulyatorlar.

Reja:

1. Mobil ilovalarni yaratish tillari va texnologiyalari
2. Android uchun Java dasturlash tilida ilovalar yaratish
3. Emulyatorlar
4. Infrared blaster haqida ma'lumot.

Mobil operatsion tizimlar – bu mobil qurilmalar (smartfonlar, tabletlar, PDA (Personal digital assistant) va boshqa mobil qurilmalar) ni boshqarish uchun mo'ljallangan tizimlar hisoblanadi.

Mobil operatsion tizimlar ikkita asosiy platforma ya'ni **“user-facing”** dasturiy ta'minot platformasi va qurilmalarni boshqarish uchun mo'ljallangan **quyi sathdagi real-vaqt operatsion tizimidan** iborat bo'ladi.

Mobil qurilmalarga dasturiy vositalar yaratish uchun platformalar.

Nomi	Dasturlash tili
Android	Java, qisman C/C++
BlackBerry	Java
iOS SDK	Objective – C, Objective Pascal

Java ME	Java
Windows Phone (WP)	Visual C++, Visual C#, Visual Basic
Marmalade	C/C++
Python	Python
FireFox OS	HTML5, CSS, JavaScript
...	...

Java yuqori darajada himoyalangan va obyektga yo'naltirilgan dashturlash tilidir. Javaning o'zini maxsus bajarilish muhiti – platformasi mavjud (JRE – Java Runtime Environment).

Javadan qayerda foydalaniladi?

Sun firmasining ma'lumotlariga qaraganda 3 mlrd. atrofidagi qurilmada **Javadan** foydalaniladi.

Masalan:

- Shaxsiy kompyuter dasturlari (Desktop Applications) – acrobat reader, media-player, antiviruslar va h.k.;
- Veb dasturlar;
- Korxonatashkilotlar dasturlari (Enterprise Applications) – bank yoki ishlab chiqarishga oid dasturlar;
- Mobil dasturlar;
- Smart kartalar;
- Robotlar;
- O'yinlar.

Kotlin-Yana bir rasmiy va kamroq mashhur til. Bundan tashqari, Android Studio tomonidan qo'llab-quvvatlanadi, sintaksisi bilan ajralib turadi .

Java sintaksisi kotlinga qaraganda ancha murakkab kodni nazarda tutadi. Misol uchun, Android-ning ishlab chiqaruvchisi ushbu ikki dasturlash tilida maydon matnini tayinlaydi:

Java

```
final TextView helloTextView = (TextView) findViewById(R.id.text_view_id);
helloTextView.setText("Hello world");
```

Kotlin

```
helloTextView.text = "Hello world"
```

Unity-kompyuter o'yinlarini ishlab chiqishning o'zaro platformali muhitidir. Birlik shaxsiy kompyuterlar, o'yin konsollari, mobil qurilmalar, Internet ilovalari va boshqalarni o'z ichiga olgan 20 dan ortiq turli xil operatsion tizimlar ostida ishlaydigan ilovalarni yaratishga imkon beradi. Unity ning chiqarilishi 2005 yilda bo'lib o'tdi va o'sha paytdan boshlab doimiy rivojlanish davom etmoqda.

Pythonni tarixi-Python Guido van Rossum tomonidan sakson va sakkizinchi yillarda Niderlandiyadagi Matematika va informatika ilmiy tadqiqot institutida ishlab chiqildi.

Python ABC, Modula-3, C, C ++, Algol-68, SmallTalk va Unix shell kabi boshqa ko'plab tillardan va boshqa skript tillaridan olingan.

Python xususiyatlari:Pythonning xususiyati quyidagilarni o'z ichiga oladi:

O'qish oson: Python nisbatan kam kalit so'zlar, oddiy tuzilish va aniq belgilangan sintaksisga ega. Bu o'rganuvchini qisqa vaqt ichida yodlab olish imkonini beradi.

O'qish oson: Python kodi juda aniq va ko'zga ko'rinadigan bo'ladi.

Oson ishlash: Pythonning muvaffaqiyati – manba kodi juda oson.

Keng standart kutubxona: Pythonning eng qudratli jihatlaridan biri kutubxonaning asosiy qismi juda portativ va UNIX, Windows va Macintosh-da o'zaro faoliyat platformalar bilan mos keladi.

Interaktiv usul: Pythonda ishlashda terminalda ishlash uchun juda qulay terminalda test qilib ko'rsa bo'ladi.

Portativ: Python keng apparat platformalarida ishlaydi va barcha platformalarda bir xil interfeysga ega.

Kengaytirilgan: Python tarjimoniga past darajadagi modullarni qo'shishingiz mumkin. Ushbu modullar dasturchilarni o'zlarining vositalarini samaraliroq bo'lishiga qo'shish yoki sozlash imkonini beradi.

Ma'lumotlar bazasi: Python barcha ma'lumotlar bazasini qo'llab quvvatlaydi.

GUI dasturlash: Python Windows MFC, Unix, X Window kabi platformalarga GUI dasturlar tuzishni qo'llab quvvatlaydi.

Moslashuvchan: Python, qobiq buyruq fayliga qaraganda katta dasturlarga yanada yaxshi tuzilish va qo'llab-quvvatlash imkonini beradi.

C/C++ C tillari yuqori ko'rsatkichga ega bo'lib, ayniqsa, mobil 3D o'yinlari kabi og'ir dasturlash imkonini beruvchi til. Android NDK sizga C/C++ da faqat dasturning bir qismini, masalan, Java orqali dasturga ulanishi mumkin bo'lgan kutubxonani yozish imkonini beradi.

Dart - veb ilovalar va mobil ilovalar kabi mijozlarni rivojlantirish uchun mo'ljallangan dasturlash tilidir. Google tomonidan ishlab chiqilgan va server va ish stoli ilovalarini yaratish uchun ham foydalanish mumkin bo'lgan dasturlash tili.

C# yordamida Androidda dasturlash *Xamarin* platformasini nazarda tutadi. Shu bilan siz Android va iOS uchun C# da bitta dastur imkonini yaratishingiz mumkin.

Java dasturlash tili va platforma

Javada asosan 4 turdagi dasturlar yoziladi:

1) Standalone applications – Linux, Mac yoki Windowsga o'rnatib, kundalik ishlatadigan dasturlarimiz: mp3 player, ofis, antivirus kabilar. Ular AWT, Swing yoki JavaFX texnologiyalari orqali tuziladi.

2) Web Applications – tarmoq orqali ishlovchi ixtiyoriy dasturlar. Eslatma: veb dasturlar ikki qismdan, server tomon hamda klient tomon (brauzer) dan iborat. Javada faqat server tomoni uchun yoziladi. Bunda servlet, jsp, jsf kabi fundamental texnologiyalardan boshlab Spring, Play kabi freymworklar qo'l keladi. Umuman

olganda brauzer uchun HTML, CSS hamda Java Scriptdan boshqa tilda yozib bo`lmaydi.

3) Enterprise Applications – bu dasturlar yirik salmoqqa ega bo`lib, odatda katta jabhalarda ishlatiladi. Banklar, tashkilotlar yoki astronomiya kabi sohalarda. Ular yuqori xavfsizlik, yuklamani (nagruzka) serverlarga teng taqsimlash (load balancing) yoki klasterlash (clustering – katta tizimdan huddi yagona obyekt sifatida foydalanish) kabi sifatlarni talab qiladi. Javada bular bor.

4) Mobile Applications – Mobil qurilmalarga mos dasturlarni ham Javada yozish mumkin. Androiddan boshlab, Java ME (JME – Java Micro Edition) gacha. JME ga misol qilib, Nokia telefonlarimiz uchun ishlangan JAR dastur va o`yinlarni misol keltirish mumkin.

Java bir necha qismlarni o'z ichiga oladi va ular :

1. Simple (oddiy)
2. Object-Oriented (obyektga yo'naltirilgan)
3. Platform independent (mustaqil platformalar)
4. Secured (himoyalangan)
5. Robust (kuchli)
6. Architecture neutral (neytral arxitektura)
7. Portable (ixcham)
8. Dynamic (dinamik)
9. Interpreted (interpretatsiyalangan)
10. High Performance (yuqori tezlik)
11. Multithreaded (ko'p potokli)
12. Distributed (model bo'yicha taqsimlanish)

iOS uchun Swift dasturlash tilida ilovalar yaratish

Apple qurilmalari axborot texnologiyalari sohasida tan olingan etakchi kompaniyalardan biridir. So'nggi ma'lumotlarga ko'ra, boshqa mobil operatsion tizimlar orasida iOS ulushi 15-20% atrofida o'zgarib turadi va ish stoli tizimlar

orasida Mac OSX ulushi, 15-20% ni tashkil qiladi. Apple qurilmalarining keng qo'llanilishi ushbu qurilmalar uchun dasturiy ta'minotga ehtiyoj tug'diradi

An'anaga ko'ra ob'ektiv-C iOS va MacOS uchun asosiy dasturlash tili bo'lib kelgan, ammo 2014 yil 2 iyunda Apple WWDC 2014 dasturchilar konferentsiyasida yangi va qulayroq dasturlash tili **Swift** taqdim etildi. Objective-C bilan taqqoslaganda Swift quyidagi xususiyatlarga ega:

Swift sof ob'ektga yo'naltirilgan dasturlash tili

Oddiylilik, aniq va aniq sintaksis

Har bir o'zgaruvchining ma'lum bir turi mavjud

Xotirani avtomatik boshqarish

Swift - bu kompilyatsiya qilingan dasturlash tili. Ya'ni, ishlab chiquvchi manba kodini yozadi va keyin kompilyator yordamida ushbu kodni boshqaruv dasturiga kompilyatsiya qiladi. Ushbu dastur fayli keyinchalik AppStore-dan yuklab olinishi va boshqa foydalanuvchilarga tarqatilishi mumkin bo'ladi. Swift Apple dasturchi muhandislari tomonidan yaratilgan. Uning yaratilishi Apple qurilmalariga dastur yaratilishini juda keng ma'noda o'zgarish yaratdi deyishimiz mumkin.

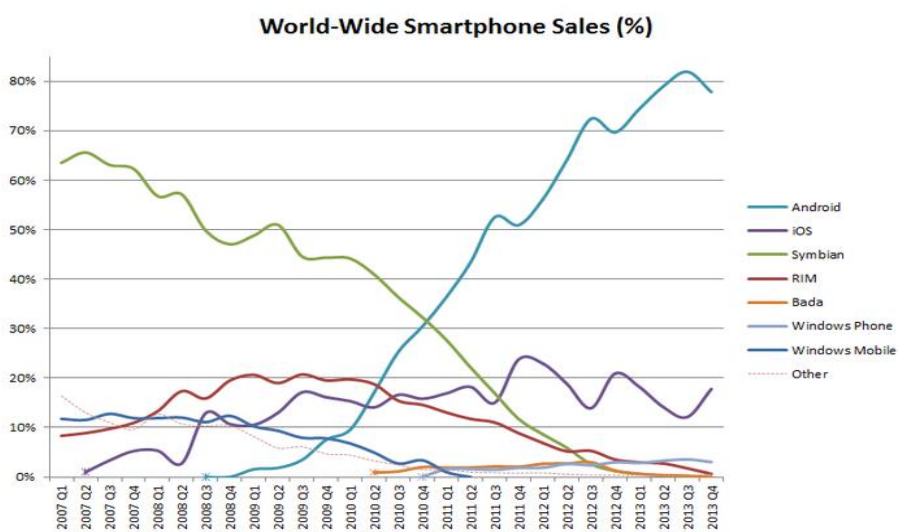
Swift tili yutuqlari:

- Soddalik;
- Tezkorlik;
- Barcha qurilmalar uchun yagona qo'llash uslubi;
- Tiplash majburiy emas (ammo tip degan tushuncha mavjud va uni qo'llay olish ham);
- Sodda integratsiya qilish (Objective C dasturchilari bimalol Swiftda qiynalmasdan code yoza oladi);
- Cocoa frameworki Swift uchun mavjud;
- Playground mavjudligi (ushbu tilni o'rganish uchun muhit);
- Yagona IDE Xcode.

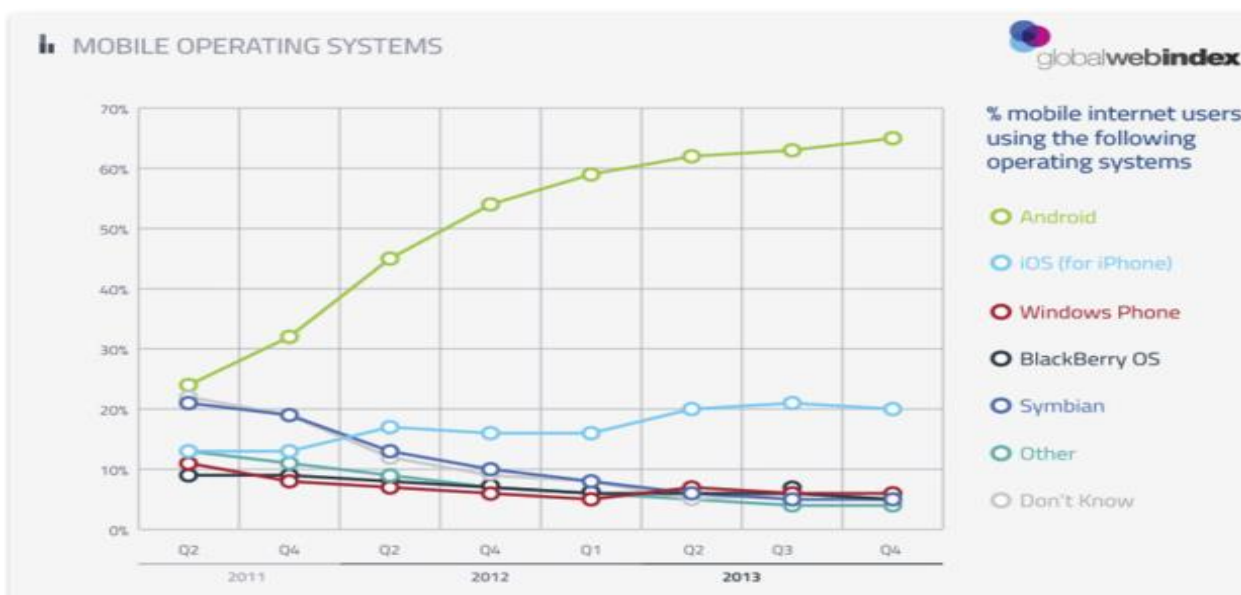


Ilovalar do'konlari

Store	2010 (millions U.S.) ^[74]
Apple App Store	\$1782
BlackBerry App World	\$165
Nokia Ovi Store	\$105
Google Play	\$102
Total	\$2155



Mobil qurilmalar uchun OT larning qo‘llanilish ko‘rsatkichlari



Zamonaviy mobil qurilmalar uchun operatsion tizimlarning asosiy funksiyalari

- * Touchscreen
- * Cellular
- * Bluetooth
- * Wi-Fi
- * GPS mobile navigation
- * camera, video camera
- * speech recognition, voice recorder
- * music player
- * near field communication (NFC)
- * infrared blaster (IR)
- * Sensorli ekran
- * Uyali
- * Bluetooth
- * Wi-fi
- * GPS mobil navigatsiyasi
- * kamera, videokamera
- * nutqni aniqlash, ovoz yozish
- * musiqa pleyeri
- * yaqin aloqa (NFC)
- * infraqizil portlovchi (IR portlovchi).

Emulyator - bu boshqa kompyuter yoki dasturga taqlid qiladigan dastur. Masalan, emulyatorlar Windows operatsion tizimini Mac-da va aksincha ishlashga imkon beradi.

Kompyuterni taqlid qilish tushunchasi IBM tomonidan eski modellari uchun mo'ljallangan dasturlarni yangi modellarda ishga tushirish usuli sifatida yaratilgan. Ular qo'llagan usul taqlid qilish uchun mo'ljallangan dasturiy ta'minot va apparat vositalarining kombinatsiyasiga asoslangan edi. Yangi kompyuterlari uchun mutlaqo yangi dasturlarni ishlab chiqish o'rniga, o'rnatilgan eski dasturlarga moslashishga imkon beradi.

Bugungi kunda emulyator atamasi odatda video o'yinlar kontekstida qo'llaniladi. Video o'yinlar emulyatorlari 1990-yillarda ommalashgan, chunki ular odamlarga Nintendo Entertainment System va Atari 2600 singari konsollar uchun zamonaviy ish stollarida o'yin o'ynashga imkon berishgan. Smartfon va planshetlarning ko'payishi bilan kompyuterda iOS yoki Android qo'llab-quvvatlaydigan emulyatorlarga bo'lgan talab tobora ko'proq talabga aylanib bormoqda.

Emulyatorlar qanday ishlaydi

Turli xil emulyatsiya usullarini qo'llaydigan har xil turdagi emulyatorlar mavjud, ammo yakuniy maqsad har doim bir xil: asl apparat yoki dasturiy ta'minotdan foydalanish tajribasini ko'paytirish. Ba'zi emulyatorlar hatto asl mahsulotning ishlash ko'rsatkichlaridan yuqori bo'lib, qo'shimcha funktsiyalarni o'z ichiga oladi.

Emulyatsiya virtualizatsiya tushunchasi bilan chambarchas bog'liq. Virtual mashinalar - bu xost tizimining asosiy apparatida ishlashi mumkin bo'lgan emulyatorning bir turi.

Nima uchun emulyatorlardan foydalanish kerak?

Dasturiy ta'minot platformaga bog'liq bo'lib qoladi, shuning uchun ishlab chiquvchilar Android, iOS, Windows va Mac uchun alohida dasturlar yaratishi kerak. Agar siz Mac foydalanuvchisiz va faqat Windows uchun mavjud bo'lgan dasturdan foydalanmoqchi bo'lsangiz, u holda bitta variant (Windows kompyuterini sotib olishdan tashqari) emulyatordan foydalanishdir.

Raqamli ma'lumotlarni saqlashda emulyatorlar ham muhim rol o'ynaydi. Eski o'yin kartridjlari kabi eski formatlarda saqlangan dasturlarni maxsus qurilma yordamida faqat o'qish uchun ROM fayllari sifatida yuklash mumkin. Keyin ROM o'zi ishlab chiqilgan original o'yin tizimi uchun emulyator yordamida ijro etilishi mumkin.

Emulyatorlarga misollar

Har bir yirik operatsion tizim uchun ko'plab tijorat va ochiq manbali emulyatorlar mavjud. Masalan:

- BlueStacks kabi emulyatorlar sizga Windows va Mac-da Android dasturlaridan foydalanishga imkon beradi.
- Xcode kabi dasturlar Mac va Windows-da iOS-ni boshqarishi mumkin.
- Appetize.io - bu har qanday kompyuterda iOS dasturlaridan foydalanishga imkon beruvchi brauzer emulyatori.
- WINE Linux operatsion tizimida Windows dasturlarini ishlaydi.
- Nestopia kabi emulyatorlar Linuxda Nintendo o'yinlarini o'ynashlari mumkin.
- SNES Classic singari konsol emulyatorlari geymerlarga zamonaviy yuqori aniqlikdagi televizorlarda eski video o'yinlarni o'ynashga imkon beradigan mustaqil uskuna.
- PlayStation Portable uchun foydalanuvchilarning Sony mobil tizimidagi boshqa konsollar uchun o'yin o'ynashlariga imkon beradigan ko'plab emulyatorlar mavjud.

Android uchun Windows emulyatorlari

Android platformasining imkoniyatlari cheklanganligi sababli, Windows OS-ning to'liq versiyasini alohida dastur sifatida ishga tushirish zarur bo'lishi mumkin. Bu zamonaviy Android qurilmalarining faol rivojlanishi tufayli to'liq bajarilishi mumkin bo'lgan vazifadir, ularning aksariyati quvvat jihatidan shaxsiy kompyuterlardan kam emas. Bugungi maqola davomida biz sizga ushbu turdagi bir nechta emulyatorlar haqida gapirib beramiz, ularning faoliyati to'g'ridan-to'g'ri smartfonning dasturiy ta'minot talablari va xususiyatlariga bog'liq.

Android uchun Windows emulyatorlari

Android-da Windows-ning to'liq versiyasini joylashtirishga emas, balki ma'lum bir vazifani bajarishga imkon beradigan juda ko'p sonli emulyatorlar

mavjud. Masalan, ko'rib chiqilayotgan dasturlarning ba'zilari faqat o'yinlarni boshqarish uchun mo'ljallangan, muqobil variantlar esa Windows-ning aksariyat funksiyalarini bajaradi, bu sizga faqat kichik dasturlardan foydalanishga imkon beradi. Ilovani tanlashda ushbu nuanslarni hisobga olish kerak, chunki hatto kuchli telefon yoki planshetda ham resurslar cheklangan.

O'QING: Android-da EXE fayllarini ochish

Bochs

Android uchun barcha Windows emulyatorlari orasida Bochs - bu operatsion tizimning to'liq versiyasini joylashtirish uchun eng yaxshi variant. Uning afzalliklari qatorida smartfon xususiyatlariga past talablarni, faqat tanlangan OS versiyasi bilan cheklangan ko'plab dasturlarni ishga tushirish imkoniyatlarini yozish mumkin.



Har doim ham sezilmaydigan yagona kamchilik - bu boshqarishdir - vaqti-vaqti bilan sichqoncha kursori imo-ishoralarga javob bermaydi yoki juda sekin harakat qiladi.



Umuman olganda, Windows-ning to'liq ishlaydigan versiyasini XP-ga qadar boshqarishingiz kerak bo'lsa, Bochs-dan yaxshiroq narsa yo'q. Bundan tashqari, har bir emulyator grafik interfeysga ega emas va dasturning imkoniyatlari va talablari oxir-oqibat bog'liq bo'ladigan komponentlarni tanlashni boshqarish imkoniyatini beradi.

Bochs-ni Google Play do'konidan bepul yuklab oling

QEMU

QEMU dasturi juda taniqli variant, chunki u nafaqat Android-da Windows-ni, balki ko'plab muqobil platformalarda boshqa operatsion tizimlarni ham taqlid qilish uchun mo'ljallangan. Ehtiyojga qarab, deyarli har qanday Windows-ni IMG yoki ISO formatidagi rasm shaklida yagona talab bilan joylashtirishingiz mumkin.



Ishlash nuqtai nazaridan dastur Bochs-dan unchalik farq qilmaydi, xuddi shu tarzda u emulyatsiyani boshqarish uchun qulay interfeysni taqdim etadi. Bunday holda, nafaqat Windows, balki boshqa operatsion tizimlar, masalan, Linux ham qo'llab-quvvatlanadi. Kamchiliklari orasida faqatgina Google Play do'konida dasturiy ta'minotning etishmasligi va rivojlanish jarayonida qiyinchiliklar mavjud.

W3bsit3-dns.com forumidagi QEMU sahifasi

Limbo

Limbo PC Emulator Windows-ni ham o'z ichiga olgan 32-bitli operatsion tizimlarning katta qismini boshqarishi mumkin. Mavjud cheklovlar Android qurilmasining texnik xususiyatlari va OS talablariga bog'liq. Xususan, dastur faqat ARM protsessorida ishga tushirilishi mumkin.



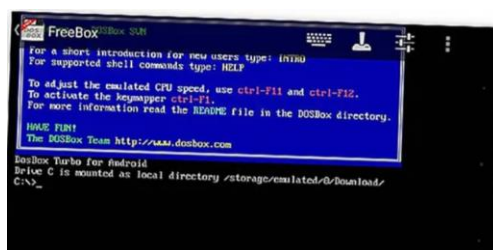
Ushbu emulyatorga katta umid bog'lamaslik kerak, chunki u QEMU-ga asoslangan bo'lsa ham, uning imkoniyatlari juda cheklangan. Limbodan

foydalanish faqat biron sababga ko'ra boshqa har qanday o'xshash analog boshlamaganida tegishli bo'ladi.

Limbo-ni w3bsit3-dns.com saytidan yuklab oling

DosBox

Emulyatorning ushbu versiyasi, nomidan ko'rinib turibdiki, Windows-ni joylashtirish uchun emas, balki telefonda DOS dasturlarini ishga tushirish uchun mo'ljallangan. Ushbu dastur fayllarni qayta ishlashda juda zo'r ish olib boradi, hatto nisbatan zaif smartfonlarda ham deyarli ishlashga oid muammolarga duch kelmaydi va nisbatan engilroq.

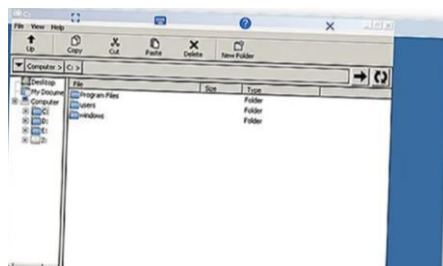


Ushbu turdagi dasturlarni tanlashda DosBox-ni hisobga olishning asosiy sababi Windows-ning 95-versiyasiga qadar ishlash qobiliyatidir. Albatta, bu ma'lum ko'nikmalarni talab qiladi, ammo natijada siz to'liq Windows-ning boshqaruvini olishingiz mumkin. Ushbu parametрни eng yaxshisi deb atash mumkin emas, lekin uni ham unutmaslik kerak. Ayniqsa, eski OSni yanada noqulay Bochs va QEMUlarni o'rnatmasdan tarqatishda.

DosBox Turbo-ni Google Play do'konida sotib oling
DosBox Free-ni Google Play do'konidan bepul yuklab oling

ExaGear

Taqdim etilgan barcha variantlardan eng ko'p ExaGear ajralib turadi, chunki bu dastur klassik Windows emulyatori emas. Bundan tashqari, uning asosiy maqsadi kompyuter uchun yaratilgan dasturlarni ishga tushirishdir. Shunday qilib, dastur mustaqil ravishda kerakli komponentlarning ishini taqlid qiladi, bu sizga ko'plab o'yinlar va ba'zi dasturlarni ishga tushirishga imkon beradi.



Ushbu dastur ma'lum bir Windows dasturini ishlatish uchun mo'ljallangan bir nechta farqlarga ega. Bunga quyidagilar kiradi:

- ExaGear - strategiyalar;
- ExaGear - RPG;
- ExaGear - Windows emulyatori.

Afzalliklar va kamchiliklar nuqtai nazaridan, barcha holatlarda dastur avval ko'rib chiqilgan analoglardan ancha barqaror ishlaydi, ammo Android qurilmasi ta'sirchan kuchga ega bo'lsa. Aks holda, ExaGear o'rnatadi va muammosiz ishga tushiradi, ammo xatolar bilan yakunlanib, hech qanday o'yin va dasturlarga taqlid qila olmaydi.



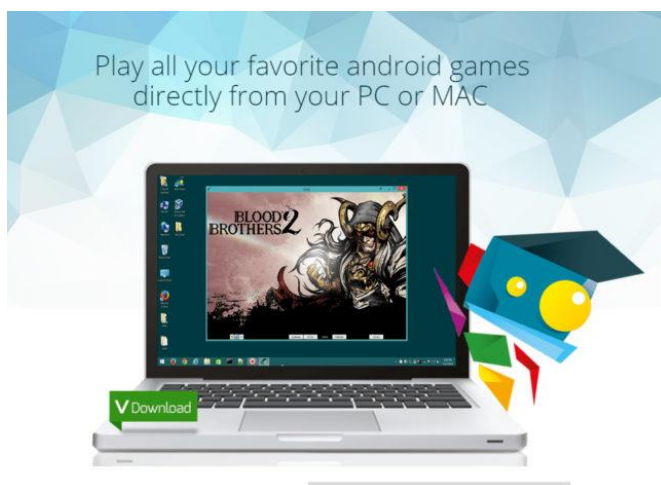
Ilovaning alohida e'tiborga loyiq afzalligi, shuningdek, ba'zi bir qo'llab-quvvatlanadigan o'yinlarning birlashtirilishi va alohida bo'limga joylashtirilganligi. Bu nafaqat Windows dasturlarini taqlid qilish, balki barcha kerakli fayllarni muammosiz yuklash imkonini beradi.

Page ExaGear - w3bsit3-dns.com forumidagi Windows Emulator
ExaGear - w3bsit3-dns.com forumidagi strategiyalar / RPG sahifasi

Bluestacks-katta fan bazasiga ega bo'lgan eng samarali Android emulyatorlaridan biri.

Andy

Bu ko'p funksiyali va kuchli Android emulyatori. Bu oson va tez ishlashi bilan ajralib turadigan oddiy va tizimlashtirilgan foydalanuvchi interfeysi bilan birga keladi.

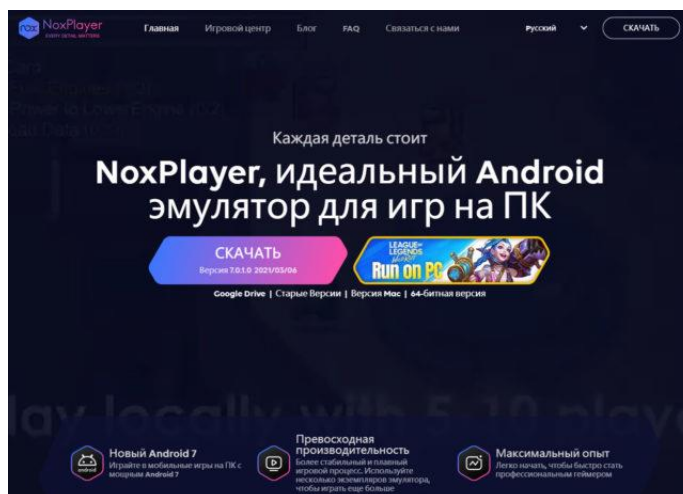


Xususiyatlari

1. Playstation va Xbox uchun to'liq qo'llab – quvvatlash, kamera integratsiyasi, mobil ilovalarni oson sinxronlash va bulutda saqlash va mikrofonni qo'llab-quvvatlash-bu uning mashhur xususiyatlarining bir nechtasi.
2. Ilova sizning kompyuteringiz bilan osongina sinxronlashtirilishi mumkin va sensorli ekranli mobil qurilmada bo'lgani kabi, u ustida ishlash imkonini beradi.
3. Bu rivojlangan xususiyatlarga ega Pro versiyasiga ega bo'lgan bepul dastur.
4. AWD Launcher texnologiyasiga ega bo'lish sizning smartfoningizga yuklaydigan har qanday dastur avtomatik ravishda ushbu emulyatorning ish stoli ekraniga joylashtirilganligini anglatadi.

Nox

Nox ham yaxshi tanlov bo'lishi mumkin. Emulyatorida ko'plab xususiyatlar va noyob takliflar mavjud va MS ish stolida Android smartfon ish stolini ishga tushirishga qodir.

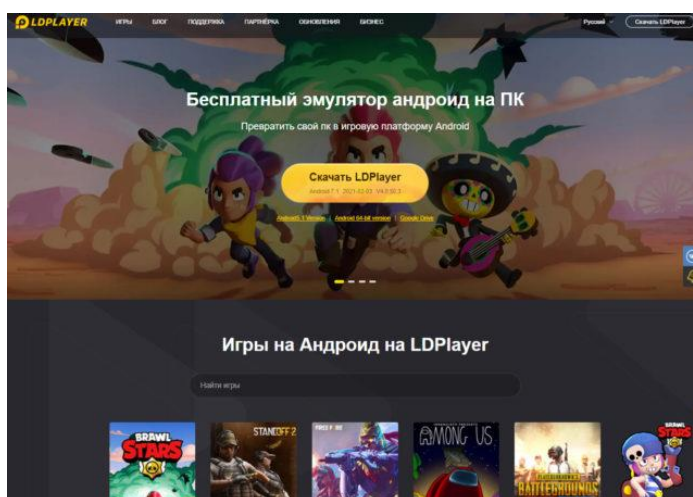


Хусусияtlari

1. Ilova oson ishlaydigan oddiy interfeysga ega.
2. Oddiy boshqaruvlari bilan jihozlangan va foydalanuvchi xohishiga ko'ra o'yin va ilovalarda ko'rsatilishi mumkin bo'lgan intuitiv klaviatura mavjud.
3. Sichqoncha kabi oddiy boshqaruvlari bilan MS ish stolida Android ilovalarini ishga tushirishning nisbatan oddiy jarayonini ta'minlaydi.
4. Material UI dizayniga ega va 5.1.1 Android Lollipop tizimiga asoslangan.

Nox eng yaxshi Android emulatorlaridan biri hisoblanadi.

LDPlayer

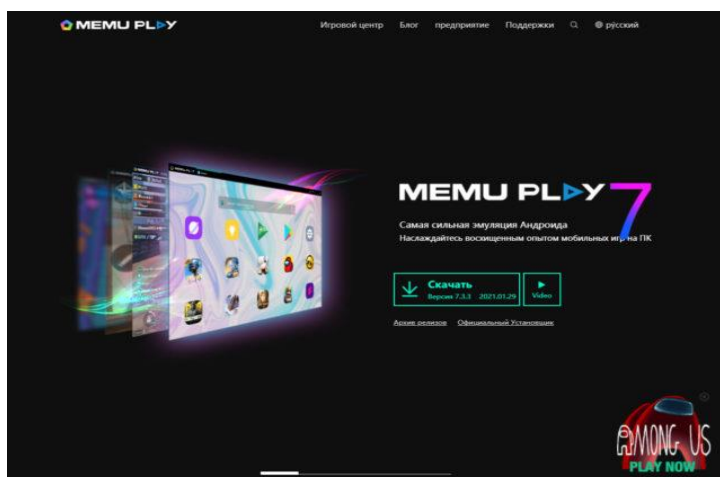


Optimallashtirilgan o'yinni va'da qilgan nisbatan yangi Android emulyatori. Bu, albatta, mukammal va barqaror ishlashi bilan Bluestacks uchun

3. Tugma ekrani, qo'lda joylashuvni sozlash, batareya, tarmoq turi va Google Play-ning to'liq qo'llab – quvvatlashi uning eng asosiy xususiyatlaridan biridir.

Memu Play

Sizning kompyuteringizda Android ilovalarini ishga tushirishni tanlashingiz mumkin bo'lgan Bluestacks uchun yana bir muqobil. Ilova foydalanuvchi bilan do'st interfeysga ega va uning engil dizayni sizning kompyuteringizning muhim resurslaridan foydalanishni talab qilmaydi.



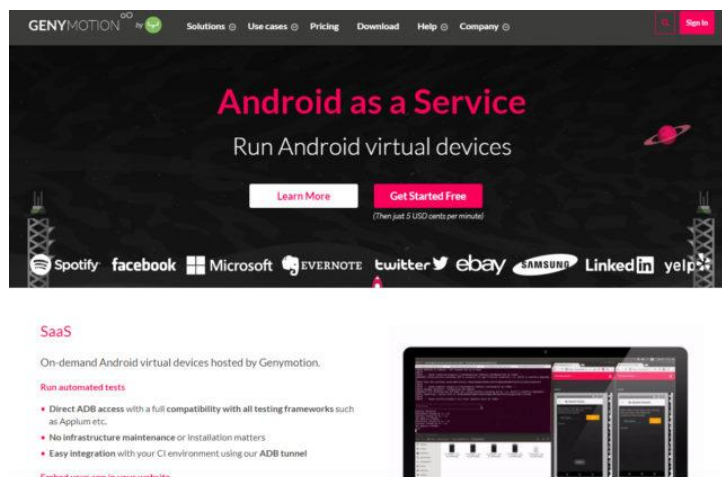
Xususiyatlari

1. Android ilovalarini cho'kishlardan xavfsiz rejimda ishga tushirishi mumkin. Bunday en barcha emulyatorlar o'zlariga ruxsat beradi.
2. AMD va Intel protsessorlarini qo'llab-quvvatlaydi.
3. Android Kit Kat 4.4, JollyBean 4.2 va Lollipop 5.0 versiyalari bilan mos keladi.
4. Barqaror va optimallashtirilgan ish tufayli mashhurlikka erishildi.

Umuman olganda, ajoyib grafikalar va takomillashtirilgan boshqaruv bilan MS ish stoli ekranida sevimli mobil o'yinlarni o'ynash imkonini beradi

GenyMotion

Bluestacks-ga muqobil variantlar ro'yxatida bu ajoyib dastur. Bu sizning MS mobil ilovalar va mobil o'yinlar, ham optimal va silliq ishlashini bahramand mos keladi.

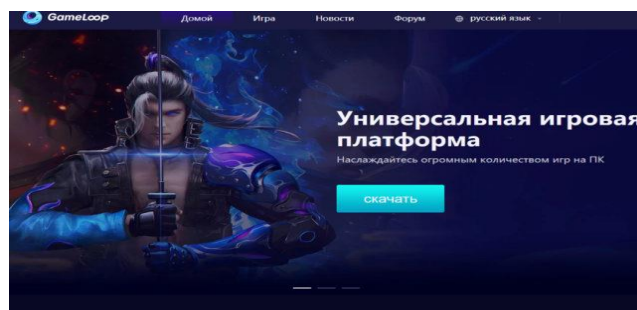


Xususiyatlari

1. Ko'pgina mobil o'yinlarning grafik tarkibiy qismining haqiqiy kuchini ochib beradi, chunki ular kuchli kompyuterlarda ishlaydi, chunki smartfonlar texnik imkoniyatlarning etishmasligi tufayli hamma narsani to'liq ishlata olmaydi.
2. Yuqori oqim tezligi bilan yangi avlod Android uchun barcha o'yinlar va ilovalar qo'llab-quvvatlanadi.
3. Soddashtirilgan va oson ish muhiti.
4. Ilova bepul va zerikarli reklama yo'q!
5. Klaviatura va sichqoncha bilan qulay boshqaruvdan bahramand bo'lish imkonini beradi.

Game Loop

Ushbu Android emulyatori o'yinchilarga kuchli kompyuter ekranida yuqori talabga ega va grafik jihatdan og'ir mobil o'yinlardan bahramand bo'lish imkoniyatini berish uchun mo'ljallangan.



Xususiyatlari

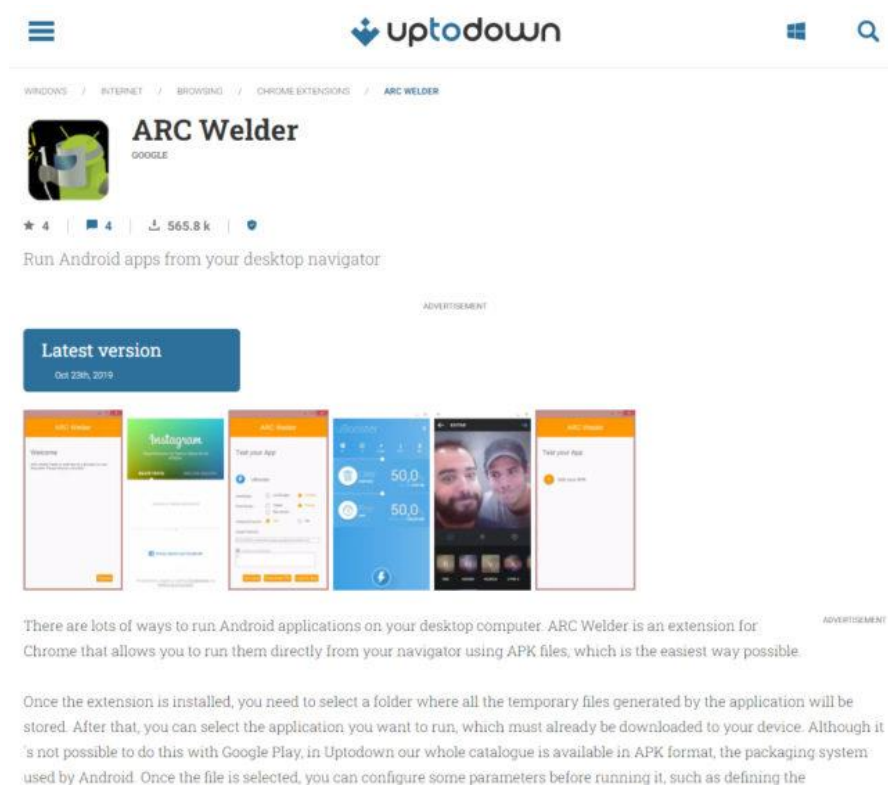
1. PUBG, Call of Duty va Free Fire kabi o'yinlarni qo'llab-quvvatlaydi.

2. Play do'koni funksiyasini qo'llab-quvvatlaydi.
3. Emulyatorida boshqaruv klaviatura va sichqoncha yordamida amalga oshiriladi.

Faqat mobil o'yinlar uchun maxsus emulyator dasturi uchun GameLoop ajoyib tanlovdur.

ARC Welder

Ushbu avtomobil brauzer kengaytmasi sifatida ishlaydi va butunlay bepul. Funktsional jihatdan ham o'ynash va ishlash imkonini beradi.



There are lots of ways to run Android applications on your desktop computer. ARC Welder is an extension for Chrome that allows you to run them directly from your navigator using APK files, which is the easiest way possible.

Once the extension is installed, you need to select a folder where all the temporary files generated by the application will be stored. After that, you can select the application you want to run, which must already be downloaded to your device. Although it is not possible to do this with Google Play, in Uptodown our whole catalogue is available in APK format, the packaging system used by Android. Once the file is selected, you can configure some parameters before running it, such as defining the

Xususiyatlari

1. Ishning soddaligi va qulayligi. Faqat brauzeringizga kengaytmani qo'shishingiz kerak va u tayyor.
2. Tizimdagi tegishli APK fayllarini qidirib Android ilovangizga kirishingiz mumkin.
3. Smartfonda mavjud bo'lgan barcha Android ilovalarining tezroq va barqaror ishlashini ta'minlaydi.
4. Kompyuter va Android qurilma uchun yagona OS.

AMIDuOS

Birinchi darajali xususiyatlar bilan jihozlangan AMIDuOS Bluestacks muqobilining eng yaxshi variantidir. Keling, nima borligini va foydalanuvchilar uni sevishini ko'rib chiqaylik...



Xususiyatlari

1. MS ish stolida mobil o'yinlar va ilovalarni oson va tezkor ishga tushirishni ta'minlaydi.
2. Bepul versiyada va pulli versiyada mavjud.
3. Androidning asosiy xususiyatlaridan tashqari, qurilma karnaylari, mikrofonlar, apparat tezlashuvi va OpenGL kabi bir nechta periferik boshqaruvni ham qo'llab-quvvatlaydi. Ya'ni, agar smartfon ularni qo'llab-quvvatlamasa yoki uning kuchi etarli bo'lmasa ham, kompyuterda Android ilovalari ishga tushiriladi va ishlaydi.
4. Android ilovalarida hech qanday cheklovlarsiz tez, aniq va aniq ishlashni va'da qiladi.

YouWave

Windows XP – Windows 10 tizimi foydalanuvchilari, jumladan, 32 va 64-bit versiyalari ushbu ilovadan xavfsiz foydalanishlari mumkin. YouWave bepul va pulli versiyalarga ega. Farqi shundaki, bepul versiya android 4.0.4 foydalanuvchisini taklif qiladi va pulli versiya 5.1.1. Ammo pulli versiyada 10 kunlik sinov muddati ham mavjud. Biroq,...

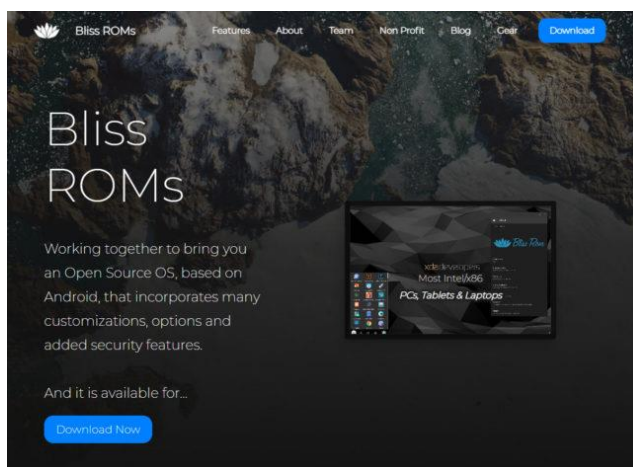


Xususiyatlari

1. Windows operatsion tizimining aksariyat versiyalari bilan mos keladi.
2. Mobil MMO-o'yinlar kechikishlar, kechikishlar va h.k. "hayvon"ishi qo'llab-quvvatlanadi.
3. Dasturda ushbu emulyatorga moslashtirilgan ilovalarni va o'yinlarni yuklab olish uchun qo'shimcha alohida ilovalar do'koni mavjud.

Bliss

Dastur foydalanuvchi uchun qulaydir, chunki ko'plab analoglardan farqli o'laroq, sozlamalarda erkinlik beradi. Turli ilovalar va o'yinlarni yuklab olishingiz va ishlatishingiz mumkin. Dastur Windows va Linux tizimlarida ishlashi mumkin. So'nggi to'rt yil ichida chiqarilgan Android-ning har bir smartfoni ushbu emulyatorga kirish imkoniyatiga ega.



Xususiyatlari

1. Foydalanuvchi batareya iste'molini nazorat qilishi va sozlashi mumkin.
2. Emulyator mobil ilovalar va o'yinlarda yuqori ko'rsatkichlarga ega bo'lishi mumkin.
3. Oddiy va foydalanuvchilar bilan do'st interfeysga ega, bu alohida-alohida sozlanishi mumkin, bu esa raqobatchilardan deyarli yo'q.

Deyarli barcha holatlarda emulyator Windows operatsion tizimining fayllarini ma'lum formatda talab qiladi. Siz mos tarqatmalarni w3bsit3-dns.com forumidagi maxsus sahifada topishingiz mumkin. Yuqori sifatli tasvirlardan foydalangan holda va ilovalar bilan ishlashda ehtiyotkorlik bilan, kamchiliklarga qaramay, Windows-ning ko'plab versiyalarini osongina taqlid qilish mumkin bo'ladi.

Nazorat savollari

1. Mobil ilovalarni yaratish tillari va texnologiyalari haqida ma'lumot bering?
2. Android uchun Java dasturlash tilida ilovalar yaratishga misollar keltiring?
3. Emulyatorlar haqida ma'lumot bering?
4. Infrared blaster haqida ma'lumot bering?
5. Swift haqida ma'lumot bering?

1.3. Mobil ilovalarni joylash uchun platformalar: Play Market, Apple Store internet do'konlari

Reja:

1. Mobil ilovalarni joylash uchun platformalar
2. Play Market va Apple Store internet do'konlari

2020 yil to'rtinchi choragidan boshlab Android foydalanuvchilari 3,14 million dasturni tanlash imkoniyatiga ega bo'lib, Google Play ilovalar do'konini

birinchi o'rinni egalladi. [Apple App Store](#) iOS uchun deyarli 2,09 million dastur mavjud bo'lgan ikkinchi eng yirik dasturlar do'konidir. Apple va Google o'zlarining do'konlaridan past sifatli tarkibni muntazam ravishda olib tashlaganligi sababli, aniq ilovalar soni o'zgarishi mumkin bo'lsada, yillar davomida ilovalar soni tobora ko'payib bormoqda.

Play Market juda foydali xizmatdir. U Googlega tegishli ("Android" operatsion tizimiga egalik qiladi) va "markali" onlayn-do'kon. Unda taqdim etilgan barcha dasturlar bir nechta toifalarga bo'linadi ("ilovalar", "o'yinlar", "musiqa", "kitoblar" va boshqalar), ularning har biri o'z sinfiga ega (janrlar, uslublar, yo'nalishlar va boshqalar). Play Market mahsulotlari ko'p jihatdan bepul, pulli dasturlarning narxi ancha past. Android smartfonining har qanday foydalanuvchisi, masalan, Highskrin markasi, Play Market ilovasidan foydalanishi mumkin.

Play Marketdan foydalanish uchun esa google tizimida ro'yxatdan o'tgan bo'lishingiz kerak.

Play Market Xususiyatlari

Androidda arzon narxlardagi smartfonlar sotib olish, xuddi shu Highscreen markasi, foydalanuvchilar o'zlarining barcha funktsiyalaridan to'liq foydalanishni xohlashadi. "Play Store" dan foydalanish smartfonlar uchun qo'shimcha imkoniyatlarni ochib beradi, bu bir qarashda aql bovar qilmaydigan ko'rinadi (masalan, smartfonni veb-kamera yoki shtrix brauzer sifatida ishlatish).

PlayMarket hisob uchun kerak:

- ✓ Android uchun maxsus yozilgan 1,5 milliondan ortiq ilovalarga kirish.
- ✓ Barcha Google xizmatlariga kirish (virtual flesh-disk va onlayn hujjatlardan pochta va xaritalarga).
- ✓ Barcha Android qurilmalarida (smartfon, planshet va boshqalar) xatcho'plar va eslatmalarni sinxronlashtirish.
- ✓ Aloqalarni sinxronlashtirish va barcha Android qurilmalar ularni kirish.

Shubhasiz, ilovaning imkoniyatlari juda keng, ammo ular faqat xizmatda ro'yxatdan o'tgandan keyin foydalanishlari mumkin.



Play Marketda ro'yxatdan o'tish

Shunday qilib, "Play Market" ni egallashdan oldin, ro'yxatdan o'tishingiz kerak. Agar foydalanuvchi allaqachon Google xizmatlaridan birida (masalan, Gmail) hisobiga ega bo'lsa, unda alohida ro'yxatdan o'tish shart emas, gmail dagi akkountimiz orqali biz Play Marketdan ro'yxatdan o'tishimiz mumkin.

Smartfonda yangi hisob bir necha daqiqada yaratiladi. Play Market yorlig'ini bosish kifoya, "yangi yaratish yoki mavjud profildan foydalanish" xabari paydo bo'ladi.» Uning ostida ikkita tugma, "yangi" va "mavjud". Agar hisob allaqachon mavjud bo'lsa, "mavjud" tugmasini bosgandan so'ng uning Login/parolini kiritishingiz kerak. "Yangi" tugmasini bosgandan so'ng siz Login/parol bilan chiqishingiz kerak (Login avtomatik ravishda yangi Gmail-qutining manzili bo'ladi). Shundan so'ng siz foydalanuvchi shartnomasini qabul qilishingiz va tasdiqlovchi captcha kiritishingiz kerak. Bu "Play Store" da asosiy hisob yaratish tugallanadi.

Play Marketni Sozlash

Ro'yxatga olish jarayonida siz telefon raqamini yoki qo'shimcha pochta qutisini ko'rsatadigan "sozlamalarni sozlash" ni tanlashingiz mumkin. Bu joriy hisobni tiklash uchun foydalidir (masalan, parolni tiklash uchun). Agar siz pul uchun ilovalarni sotib olishni rejalashtirmoqchi bo'lsangiz, "kredit kartasini qo'shish" variantini tanlashingiz kerak.

Hisobdan mablag' yechib olinadigan karta raqamini kiritish kerak bo'ladi. Agar pulli yuklab olishni rejalashtirmasangiz, ushbu tugmani e'tiborsiz qoldirish yoki "hozir emas" tugmasini bosish kerak. Har qanday holatda ham, bu Sozlamalar barcha ixtiyoriy va har qanday vaqtda sozlanishi mumkin. Bundan tashqari, darhol barcha Google xizmatlarini sinxronlashtirishni amalga oshirishingiz mumkin, buning uchun alohida menyu paydo bo'ladi. Shundan so'ng, juda kuchli smartfon (bir xil Hayskrin) to'liq onlayn ish uchun ishlatilishi mumkin.

Hisobni yaratganingizdan so'ng darhol bajarilishi kerak bo'lgan asosiy sozlash-yangilanish rejimini sozlash. Aslida, telefon internetga ulangandan so'ng, barcha ilovalar sukut bo'yicha yangilanadi. Faqat bepul Wi-Fi bilan ishlashda bu juda qulay, ammo mobil trafik mavjud bo'lganda jiddiy xarajatlarga olib kelishi mumkin.

Shuning uchun, "Play Market" sozlamalarida "avtomatik yangilanish" bandini tanlashingiz kerak va aksincha "faqat Wi-Fi orqali tanlash mumkin. "Hech qachon" ni tanlashingiz mumkin, bu holda foydalanuvchi taqiqni olib tashlamaguncha ilovalar hech qachon yangilanmaydi.



Play Marketda navigatsiya va yuklab olish Navigatsiya juda oddiy va tushunarli. Barcha ilovalar toifalarga bo'linadi va alohida-alohida "eng yaxshi to'langan" va "eng yaxshi bepul", "eng yangi" va "eng ko'p sotilgan" deb nomlanadi. Kategoriyalar barmoq bilan ochiladi, ro'yxat ekran bo'ylab

harakatlanadi.

Kerakli ilovalarni toifalar orqali yoki to'g'ridan-to'g'ri "Play Market" orqali qidirish mumkin. Mahsulotning aniq (yoki kamida taxminiy) nomi ma'lum bo'lsa, oxirgi usul juda qulay. Ilovani tanlagandan so'ng (oddiy teginish bilan) uning tavsifi ochiladi. Keyin "o'rnatish" tugmasini bosib va o'rnatish tugagandan so'ng – "ochiq". O'rnatish vaqtida siz ilovani talab qiladigan ruxsatlarga e'tibor berishingiz kerak. Agar pulli yuklab olish amalga oshirilsa, karta hisobidan pul avtomatik ravishda o'chiriladi. Agar karta bog'lanmagan bo'lsa, pulli dastur yuklab olinmaydi.

Apple store dan foydalanish tartibi

App Store-da mavjud bo'lgan millionlab ajoyib dasturlar iPhone va iPod touch-ning haqiqiy kuchini ochib beradi.

AppStore-Apple tovar qurilmalari uchun elektron ilovalar va dasturlar do'koni:

- iPhone;
- iPad;
- iPod
- Touch;
- MacBook;
- Mac;
- Apple Watch;
- Apple TV.

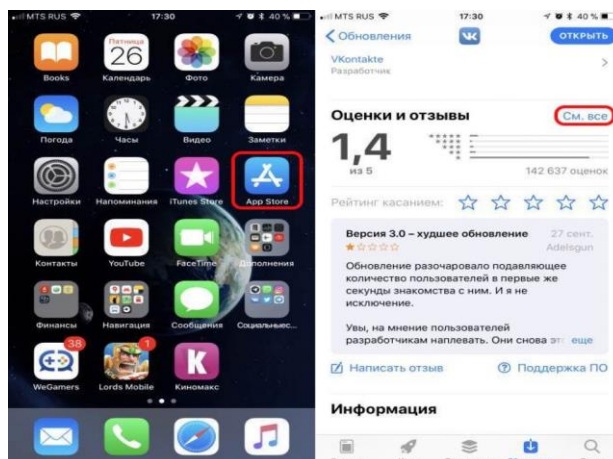


App Store logotipi

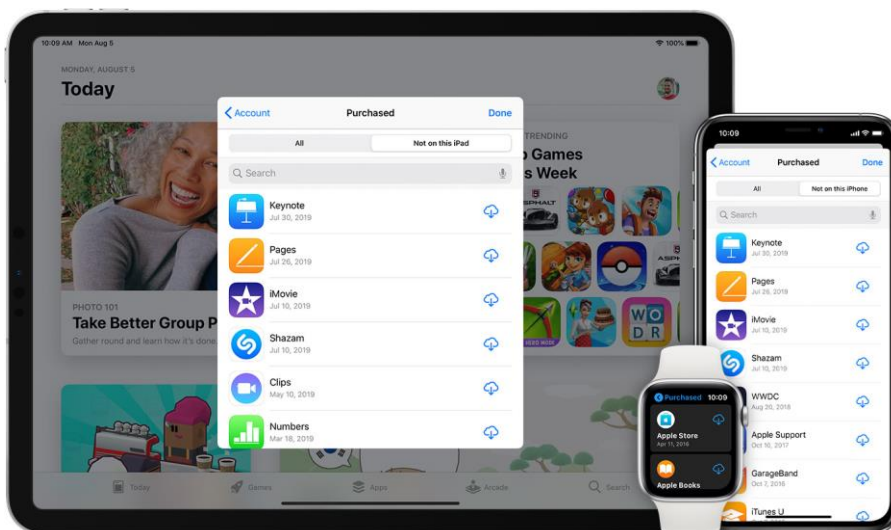
App Store-bu nima? App Store "App Store" deb tarjima qilingan. Mobil ilovalar formatini qo'llab-quvvatlaydigan yangi iPhone 3G keng jamoatchilikka taqdim etilgandan so'ng, iTunes yangilanishi paytida 2008 yil 10 iyun kuni ishga tushirildi.



AppStore-dan foydalanish juda oson. Dastur allaqachon har bir yangi Apple qurilmasiga o'rnatilgan. Har qanday ilovani yuklab olish imkoniyatini olish uchun siz hisob-kitob qilishingiz va Apple ID-ni kiritishingiz kerak. Shundan so'ng siz "Download" yoki "Get" belgisini bosish orqali gadgetingizga yoqqan har qanday ilovani yuklab olishingiz mumkin. Keyin dastur Apple ID parolini kiritishni yoki Touch ID yoki Face ID bilan tasdiqlashni talab qiladi.



AppStore yuklab olishdan oldin Apple ID talab qiladi



Apple qurilmalarida ilovalarni sotib olish

Xarid qilish "Apple ID"ga" to'lov va etkazib berish " bo'limiga bank kartasi ma'lumotlarini qo'shish orqali amalga oshiriladi. U erda siz to'lov usulini saqlashingiz va barmoq izi bilan sotib olishni tasdiqlashingiz mumkin.

AppStore mobil telefonida bir nechta bo'lim mavjud. Ilovaning asosiy sahifasida birinchi narsa do'konni tavsiya etadigan eng yaxshi dasturlarga ega bo'lgan bo'limdir. Har kuni xizmat kontentni yangilaydi, ba'zi dasturiy ta'minotning yashirin imkoniyatlarini ochib beruvchi maslahatlar va maqolalar nashr etadi. Bundan tashqari, eng katta talabga ega bo'lgan ilovalarni yangilash taklif etiladi. AppStore tomonidan tavsiya etilgan yangi ilovalar uchun reklamalarni topishingiz mumkin. Bu erda:

- fotografiya asosida ishlaydigan skanerlar, ofis dasturlarining elektron versiyalari, jurnallar va hatto ish qidirish ilovalari kabi ish dasturlari;
- turli xil o'yinlar va barcha yoshdagilar uchun;
- pedometrlar, odat izdoshlari, suv izdoshlari, kaloriya hisoblagichlari kabi sog'liqni saqlash dasturlari;
- fotosuratlar va videolarni qayta ishlash dasturlari, shuningdek effektlar bilan suratga olish;
- musiqa tinglash dasturlari. iPhone-da turli xil dasturlarni yuklab olishingiz

mumkin.



iPhone-da turli xil dasturlarni yuklab olish imkoniyati mavjud

Nazorat savollari

1. Mobil ilovalarni joylash uchun platformalar haqida ma'lumot bering?
2. Play Market dan foydalanish ketma-ketligini tushuntiring?
3. Apple Store dan foydalanish ketma-ketligini tushuntiring?
4. Play Marketdan ro'yxatdan o'tish qay tartibda amalga oshiriladi?

1.4. Mobil ilova ishlab chiqish tamoyillari, talablarni o'rganish va texnik topshiriqni shakllantirish, platformani tanlash

Reja:

1. Mobil ilova ishlab chiqish tamoyillari
2. Texnik topshiriqni shakllantirish
3. Platformani tanlash.

Hozirgi kunda zamonaviy odamni qo'lida mobil moslamasiz tasavvur qilish qiyin.

Telefonlar, smartfonlar, planshetlar, netbuklar va boshqa aloqa vositalari bizning hayotimizni osonlashtirish va ta'minlash uchun mo'ljallangan tezkor kirish, kerakli ma'lumotlar dunyoning istalgan burchagida turib olish imkoniyati mavjud. Bundan tashqari, bozor o'sib bormoqda mobil ilovalar- orqali aloqa qilish, xaridlarni amalga oshirish, mehmonxonalarni bron qilish, taksiga qo'ng'iroq qilish, ovqatga buyurtma berish, vaqtni 100 % ga o'tkazish mobil Internet usullar uni dasturlarga bag'ishlashadi.

Biznesda, ilova veb-saytga ega bo'lgandek zarur bo'ladi. Shu sababli, professional mobil ishlab chiquvchilarga talab har yili o'sib bormoqda.

Mobil ilovani yaratish bosqichlari



Tahlil

Har bir dastur g'oyadan boshlanadi.



Kelajakda xizmat bilan shug'ullanish kerak, nima bizga aytib, va tahlilchilar yig'ish uchun biz davom etmoqda. Bozorning chuqur kesilishi, mavjud echimlarni tahlil qilish, raqobatchilar va xaridorlarning xatti-harakatlarini o'rganish ... tahlilning har bir bosqichida biz oxirgi foydalanuvchini eslaymiz va mijozning hayot aylanishini o'ylaymiz. Bu odamlar yangi ilovadan qanday foydalanishi va iloji boricha qulay, tushunarli va foydali bo'lishini birgalikda tushunishga yordam beradi. Bunday xizmat sizning biznesingizga foyda keltiradi.

Tahlil-bu juda muhim bosqich. Undan voz kechmang va texnik topshiriqdan loyihada ishlashni boshlang. Tahlil jarayonida bozorda kim ekanligini, kimga rahbarlik qilishni va qanday qilib yaxshiroq qilishni tushunamiz. Biz odatda eng yaxshi amaliyotlarni to'playmiz va mijozga 100% ishlaydigan tasdiqlangan echimlarni taklif qilamiz.

Tayyorgarlik. Fikr tug'ilganidan keyingi qadam loyiha ustida ish rejasini tuzishdir. Uning eng oson yo'li raqobatchilarni tahlil qilishdan boshlashdir. Bozorda siz yaratmoqchi bo'lgan o'xshash mobil ilovalarni toping va quyidagi omillarni tahlil qiling: Yuklab olishlar soni - agar dastur yuklansa, unda kimdir kerak. Baholash va sharhlar - zaif nuqtalarni taqdim etishga va foydalanuvchilarga yoqadigan "chiplar" ni o'z mahsulotiga moslashtirishga yordam beradi. Shu bilan birga, istemolchilarning e'tirofini qanday qo'lga kiritganini tushunishga harakat qiling.



Monetizatsiya. Ilovadan xaridlar, obuna yoki premium versiyalar uchun to'lovlar, reklama joylashtirish, ma'lumotlarni sotish - ilovani bepul tarqatish sarf xarajatlar monetizatsiya qilishning usullari hisoblanadi. Eng yaxshi variantni tanlash uchun, odatda raqobatchilarni nima qilayotganini ko'rib chiqiladi va monetizatsiyani qaysi nuqtada boshlashni aniqlanadi va buning so'ngida **marketing** bilan shug'ullaniladi

Texnik topshiriq

Biz kelajakdagi ilovaning funktsionalligi va dizaynining batafsil tavsifini tuzamiz. Biz foydalanuvchi belgilarini aniqlaymiz, foydalanuvchi hikoyalarini tasvirlaymiz (foydalanuvchi hikoyasi), foydalanuvchi sayohat xaritasini tuzamiz

(Customer Journey Map) va xizmat uchun texnik talablarni shakllantiramiz. Ya'ni, biz qanday dastur bo'lishi kerakligini, nima qilishi kerakligini va qanday ishlashini aniqlaymiz.

Ushbu texnik topshiriq (TZ) tufayli dizaynerlar va ishlab chiquvchilar jamoasi mijoz qanday xizmatni olishni istayotganini aniq tushunadi va dastlabki g'oyani bosqichma-bosqich amalga oshiradi.

Natijada nima:

- ilovada bo'lishi kerak bo'lgan funktsiyalar ro'yxati;
- interfeys, foydalanuvchi rollari, xavfsizlik, ishlash va boshqa funktsional bo'lmagan talablar;
- ushbu talablarning barchasi qanday amalga oshirilishini tavsiflash;
- loyiha smetasi.

Mobil ilovani ishlab chiqishning eng muhim bosqichi

Texnik topshiriqni ishlab chiqish bosqichlari

Dasturni ishlab chiqish uchun TP miqdori dasturning hajmi va murakkabligiga bog'liq bo'ladi.

Maqsadlarni aniqlash va tadqiq qilish

Rivojlanish jarayonida biz mijoz bilan doimiy muloqotda bo'lamiz. Biz ilovaning maqsadini aniqlab olishimiz va rivojlanish bilan bog'liq bo'lgan haqiqiy manba ma'lumotlarini olishimiz kerak.

Asosiy tuzilma

Biz oddiy tilda hamma narsani tasvirlaydigan TZ ning soddalashtirilgan versiyasini ishlab chiqamiz. Biz mijoz bilan barcha jarayonlarni tushunishda tekshiramiz — agar yangi ma'lumotlar paydo bo'lsa, biz ularni keltiramiz, agar bo'lmasa, biz batafsil o'rganishni davom ettiramiz

Texnik talablar va cheklovlar

Laynerning turiga qarab, ilova va talablarga ega bo'lgan doirani batafsil bayon qilamiz. Misol uchun, serverga yuk haqida qo'shimcha ma'lumot berilishi mumkin: u nima bo'ladi va uni qanday ta'minlash kerak.

Mantiq

Oqim sxemalari yordamida biz kichik tizimlar orasidagi dasturni va o'zaro ta'sirni aniq tasvirlab beramiz, shunda loyihani amalga oshiruvchi texnik mutaxassis hech qanday savol bermaydi.

Mijoz va ma'muriy interfeys

Grafikda biz interfeysning mijoz va dastur administratori uchun qanday ko'rinishini, qanday ranglar va elementlardan foydalanishni tasvirlaymiz.

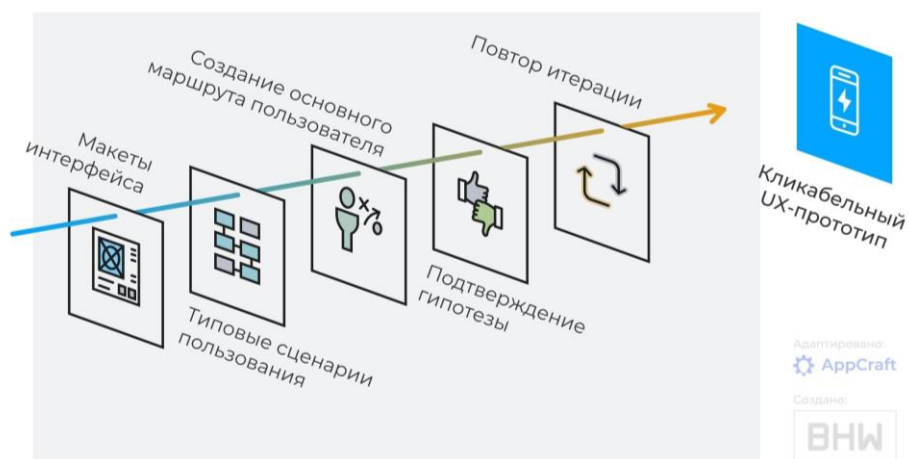
Server qismi va integratsiya

Dasturiy ta'minot uchun talablarni tasvirlab beramiz.

Texnik topshiriq-loyihaning kelajagi bog'liq bo'lgan rivojlanishning birinchi bosqichi. Barcha nuances va xususiyatlarni hisobga olish muhim: har qanday keyingi o'zgarishlar va tuzatishlar moliyaviy xarajatlar va ishga tushirishni kechiktirish.

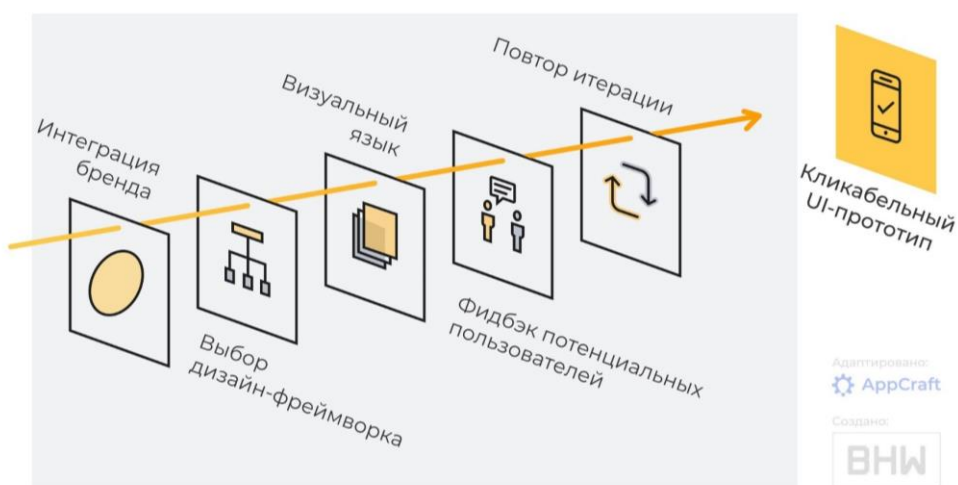
UX-dizayn .Ichki arxitektura

Ichki arxitektura mobil ilovaning funksiyasiga va ma'lumotlarni qayta ishlash va saqlash uchun tanlangan usulga bog'liq. Odatda biz ikkita ro'yxatni tuzamiz – ilovaning xususiyatlari va asosiy foydalanuvchi xaritasi. Ular kelajakdagi barcha me'moriy ishlar uchun asos bo'lib xizmat qiladi.

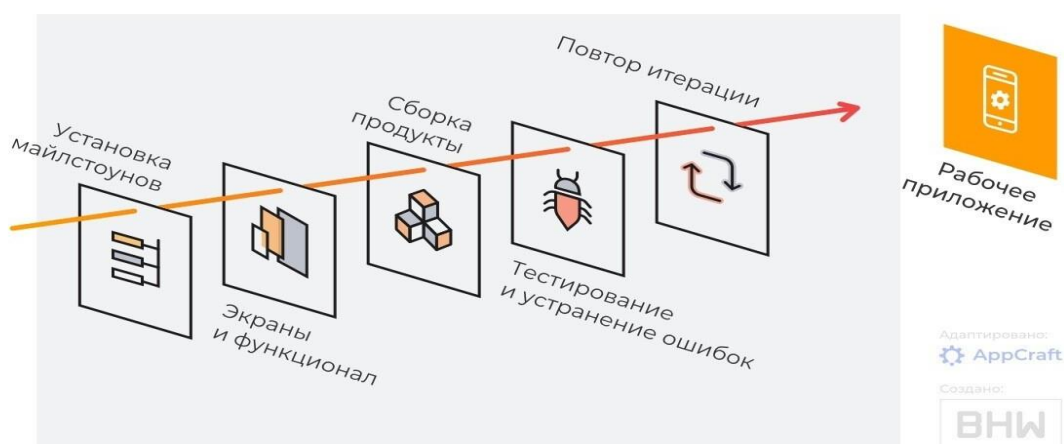


Vairframe. Keyin biz taxtada yoki qog'ozda ekranlarni chizamiz va ularga kerakli elementlar va ma'lumotlarni joylashtiramiz. Bir xil element bir nechta ekranlarda mavjud bo'lishi mumkin bo'lsa-da, u "uy"bo'lishi kerak. Ushbu bosqichda loyiha har qanday o'zgarishlarni amalga oshirishi mumkin. Asboblar: Balsamiq, Sketch, Axure, Figma, RealtimeBoard, Wireframe.cc va Whimsical

UI-dizayn. Uslublar-bu mobil ilovaning "ichki qismi" ni tugatish va uning mavjudligini oshirish uchun qurilish materiallari. Yaxshi o'ylangan uslubsiz, dizayn elementlari ranglarni o'zgartiradi va ekranda suzadi, foydalanuvchini chalkashtirib yuboradi. Mobil ilovaning uslubi bo'yicha qo'llanma imkon qadar batafsil bo'lishi va tomoshabinlarning xususiyatlariga tayanishi kerak. Tajribali UI-dizayner ranglar palitrasi, shriftlar va vidjetlar (tugmalar, shakllar, piktogramma va boshqalar) uchun yuzlab variantlarni taklif etadi.



Tizim arxitekturasi. Mobil ishlab chiqarishda juda ko'p texnologiya mavjud. Ba'zilar deyarli xarajatlarni talab qilmaydi, lekin yuqori mahsuldorlik bilan farq qilmaydi, boshqalari esa ajoyib natijalar uchun katta investitsiyalarni talab qiladi.



Mobil ilovalarni ishlab chiqishda eng yaxshi dastur

[1. Appy Pie](#)

[2. Zoho Creator](#)

[3. AppSheet](#)

[4. Appian](#)

[5. Appery.io](#)

Raqamli xizmatlaringizga qo'shimcha ma'lumot va kommunikatsiyalar qatlamlarini taqdim etish uchun oshirish va mashinada o'qitish imkoniyatlari tobora kengayib borayotgan texnologiya rivojlanib bormoqda. Nafaqat iOS yoki Android uchun, balki aqlli televizorlar, o'yin pristavkalari va boshqa texnik vositalar hamda dasturiy ta'minot yechimlarini loyihalash va kodlashni taklif qiladigan ko'plab kompaniyalar mavjud.

Shu bilan birga, asosiy shablonlar va konfiguratsiyalardan oq yorliqli ilovalarni yaratish uchun dasturiy ta'minotni ishlab chiqish platformalari ham mavjud. Buning maqsadi, korxonalariga talabga binoan o'zlarining ichki dasturlarini yaratishni yoki hatto ochiq bozor uchun dasturlarni yaratishni osonlashtirishdir.

1. Appy Pie



Ilova ishlab chiqish uchun eng yaxshi platforma

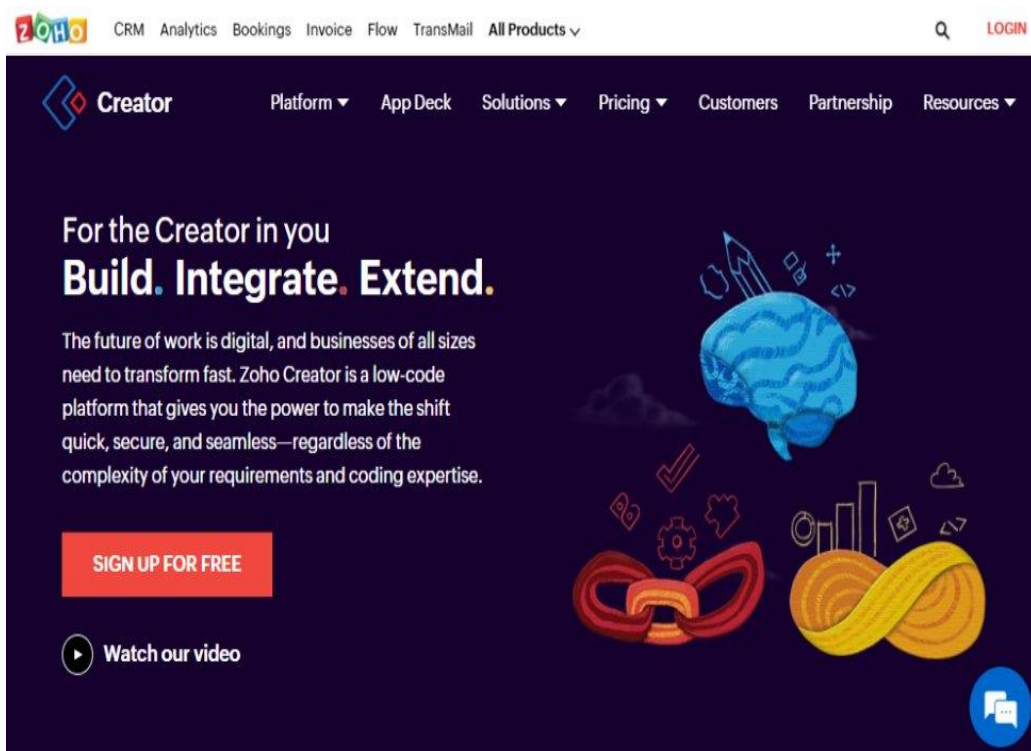
[Appy Pie](#) - bu hech qanday kodlashni talab qilmasdan o'zingizning ilovalaringizni ishlab chiqishga imkon beruvchi dasturiy ta'minot platformasi. Sizga kerakli dasturni yaratish uchun ishlatish uchun qulay bo'lgan bir qator variantlar va vositalar mavjud.

Jarayon nisbatan sodda bo'lib, dizaynni tanlash va uni shaxsiylashtirish, kerakli xususiyatlarni qo'shish, so'ngra Google Play va Apple AppStore-da nashr etishni o'z ichiga oladi.

Amaldagi interfeys - bu chat-botlar, sun'iy intellekt, kengaytirilgan yoki virtual haqiqat kabi funktsiyalarni qo'shishga imkon beruvchi oddiy tortish va tushirish tizimi. Rivojlanishda yordam berish uchun sizga yordam beradigan o'quv platformasi va o'quv videokliplari to'plami taqdim etiladi.

Zoho Creator

Moslashuvchan dastur ishlab chiqaruvchisi



[Zoho Creator](#) - bu yangi dastur yaratishga yoki tayyor dasturdan foydalanishga imkon beruvchi ko'p platformali dastur tuzuvchisi.

- Dasturiy ta'minot, masalan,
- savdo-sotiqni boshqarish,
- buyurtmalarni boshqarish,
- tadbirlarni boshqarish,
- logistika yoki ishga yollashni kuzatib borish

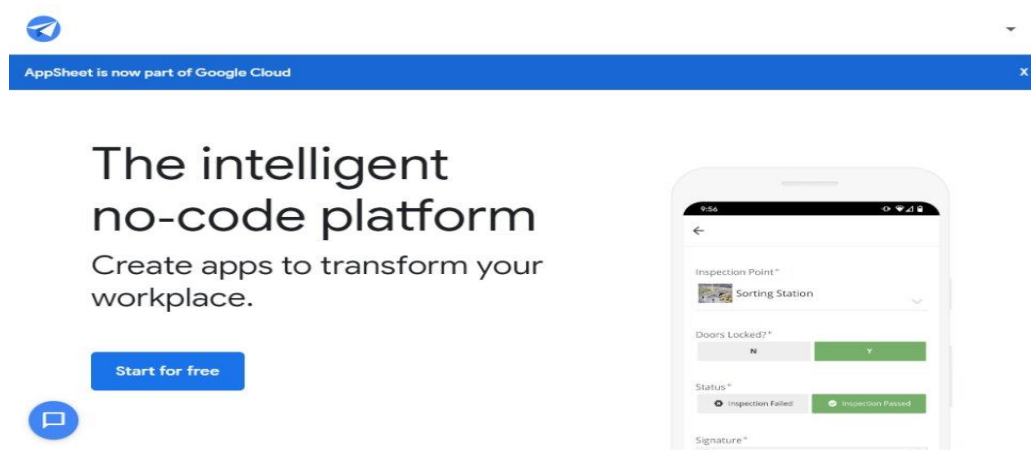
kabi funktsiyalar uchun ilovalar yaratish kabi vazifalarni soddalashtirish va avtomatlashtirishga intilayotgan korxonalariga qaratilgan.

Biroq, nima qilishni xohlasangiz ham, uni o'zingiz xohlagan tarzda sozlashingiz mumkin.

Dastlab o'z kompaniyalari ehtiyojlarini qondirish uchun dasturlarni ishlab chiqmoqchi bo'lgan korxonalariga qaratilgan bo'lsa, Zoho Creator keng bozor uchun dasturlarni ishlab chiqishda ham foydalanish mumkin.

AppSheet

Biznes yechimlari uchun dasturlar



[AppSheet](#) - bu sizning biznesingiz uchun buyurtma qilingan dasturlarni yaratishga imkon beradigan, boshqa kodlar bilan yozish yoki ishlab chiqishga imkon [bermaydigan](#) boshqa platforma.

Elektron jadval formatidagi o'zingizning bulutli ma'lumotlaringiz asosida siz o'zingizning ilovangizni joylashtirishdan oldin kerakli variantlar va xususiyatlar ustida ishlash uchun shablonni tanlashingiz mumkin.

Google Sheets va Forms, Office 365 dan Excel va hatto Salesforce-dan emas, balki bir nechta manbalardan ma'lumotlarni birlashtirish oson. Shu bilan bir qatorda, Dropbox yoki OneDrive-da saqlangan elektron jadvallarni yoki to'g'ridan-to'g'ri MySQL yoki SQL Server-dan import qilishingiz mumkin.

Buning ustiga siz GPS va xaritalar, rasmlarni olish, kodlarni skanerlash, imzolarni yozib olish, jadvallar va elektron pochta xabarnomalari kabi xususiyatlarni qo'shishingiz mumkin.

Appian



[Appian](#) "past kodli" dasturlarning rivojlanishiga ko'maklashadi, ular o'zlarining reklama g'oyasini ishlab chiqish va dasturni to'ldirish orasida sakkiz hafta ichida reklama qiladilar.

Appian 1999 yilda Maykl Bekli, Metyu Kalkins, Robert Kramer va Mark Uilson tomonidan tashkil etilgan.

Appian dasturlarini ishlab chiqish dasturiy ta'minotining asosiy yo'nalishi kompaniya ma'lumotlarini foydali va mazmunli ravishda taqdim etish uchun avtomobil jarayonlari va sun'iy intellekt yordamida optimallashtirish, biznes uchun korporativ dasturlardir.

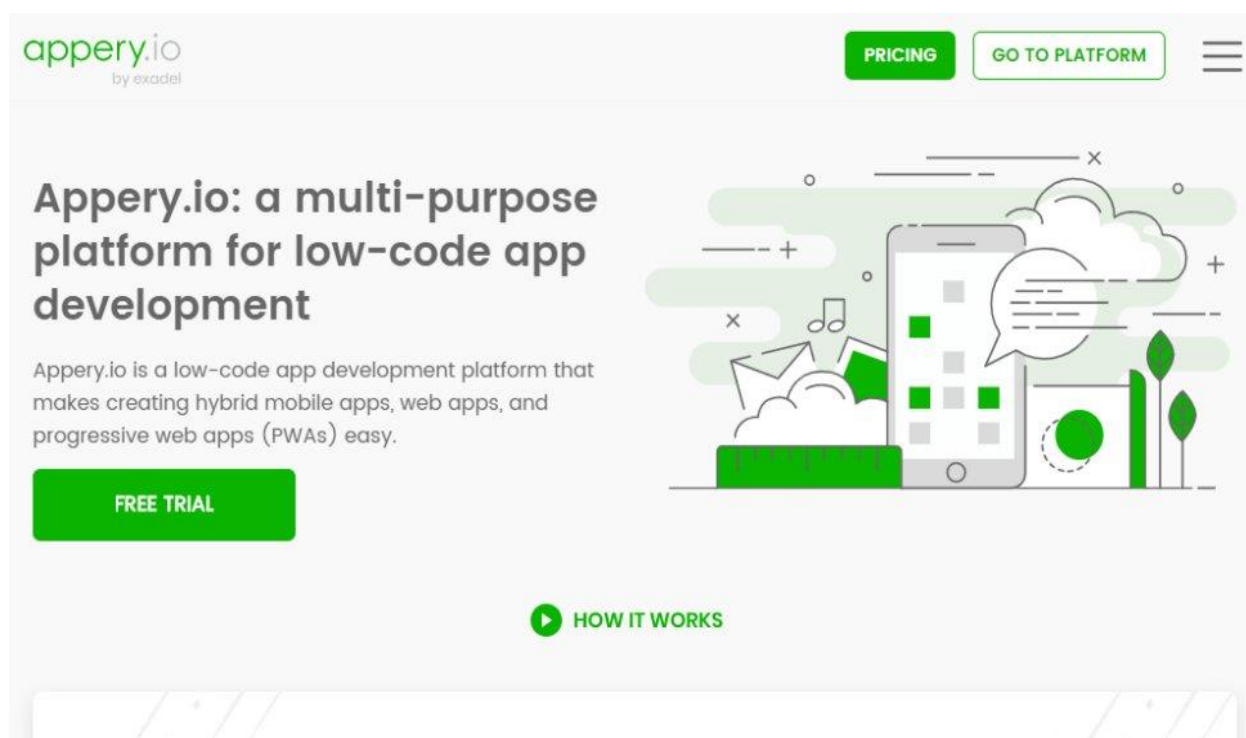
Bundan tashqari, bir nechta manbalardan olingan ma'lumotlarni tashkil qilish orqali ma'lumotlar birlashtirilib, boshqaruv jarayonlari va operatsiyalargacha bo'lgan ish oqimlari to'g'risida hamma narsalar haqida tushuncha berish uchun haqiqiy imkoniyatlar yaratilishi mumkin.

Rivojlanish jarayonining o'zi kerakli funktsiyalarni tanlash va ular orqali ma'lumotlarni qanday yo'naltirish kerakligini ingl. Tahrirlovchidan foydalanishni o'z ichiga oladi. Maqsad ma'lumotni oqilona boshqarish uchun murakkab variantlarni sodda tarzda o'rnatishga imkon berishdir.

Tugallangandan so'ng, dizayn Android va iOS uchun veb-dastur yoki mahalliy dastur sifatida saqlanishi mumkin. Bo'sh vaqtni keltirib chiqarmay, qo'shimcha o'zgarishlar tezda amalga oshirilishi mumkin.

Appery.io

Raqamli mobil platformalar uchun



[Appery](#) - bu dasturlarni ishlab chiqaruvchi, ishlab chiqaruvchilar uchun o'z dasturlarini yaratish uchun platformalar taklif qiladigan dastur ishlab chiqaruvchisi. Ilovani yaratish interfeysidan foydalanish va ma'lumotlar manbalarini tanlash bilan bir qatorda HTML 5 va Javascriptdan foydalanish kabi oson.

Maqsad - tezroq va arzonroq ishlab chiqish jarayoni uchun dasturni ishlab chiqishni boshlash. Jarayon bir haftagacha davom etishi mumkin, shu vaqt ichida Appery o'rnatish, sozlash, integratsiya, sinov va yakunlash uchun o'qitishni o'rnatadi.

Ishlash vaqtining ko'payishi bilan bir qatorda, Appery kengaytiriladigan narsalarga e'tibor qaratish imkonini beradi, shu bilan bir xil ilovalarni osongina va sezilarli ishlab chiqish vaqtini talab qilmasdan o'zgartirish mumkin.

Ochiq platformada qurilgan Appery, rivojlanayotgan ilovalardagi replikatsiya ehtiyojini kamaytirish uchun ehtiyojlar asosida bir xil dasturdan bir

nechta dastur ishlab chiqishga imkon beradi. Bundan tashqari, konfiguratsiyalarni oldindan o'rnatilgan sozlamalarga bog'liq emas, balki o'zgartirilishini ta'minlash orqali, bu dasturlarni korxonaga ehtiyojlariga qarab moslashuvchan rivojlantirishga imkon beradi.

Appery-da reklama uchun hech qanday doimiy to'lovlar yo'q, shuning uchun kotirovka bo'yicha qo'shimcha ma'lumot olish uchun kompaniyaga murojaat qilish kerak bo'ladi.

Nazorat savollari

1. Mobil ilova ishlab chiqish tamoyillari haqida umumiy ma'lumotlar bering?
2. Texnik topshiriqni shakllantirish qay tartibda amalga oshiriladi?
3. Mobil ilovalarni ishlab chiqishda qanday platformalardan foydalaniladi Platformani tanlash.

1.5. Mobil ilova dizaynini ishlab chiqish, UI/UX dizayn

Reja:

1. UX dizayn
2. UI dizayn

UX-dizayn

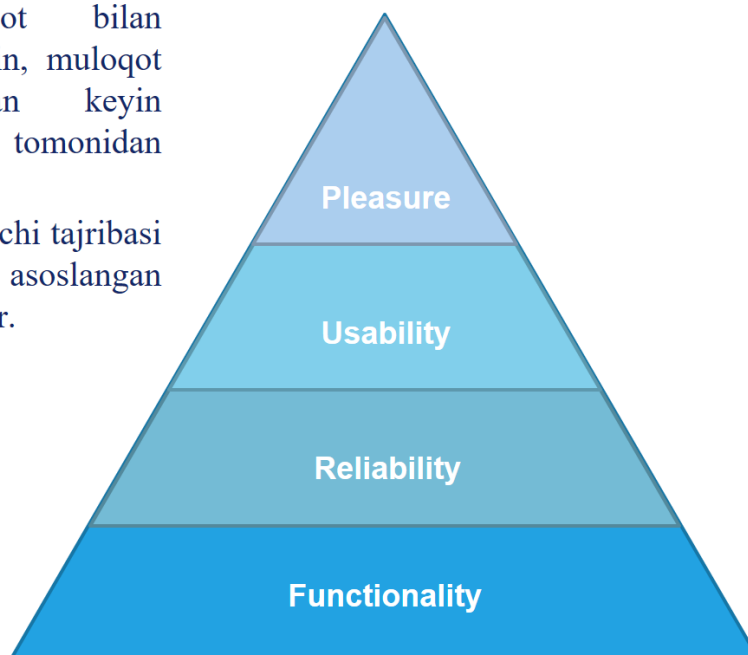
(user experience design)

Piramida talablar

- ❖ Mahsulot talablarini piramida darajalariga ko'ra joylashtirilishi:
- ❖ Mahsulot to'liq funktsionallikka ega bo'lishi kerak
- ❖ Mahsulot foydalanish uchun qulay bo'lishi kerak (mavjudligi)
- ❖ Mahsulot bilan ishlashdan foydalanuvchi hissiy zavq olishi kerak
- ❖ Mahsulot ishonchli ishlashi kerak

UX-dizayn piramidasi

- ❖ UX dizayni-mahsulot bilan muloqot qilishdan oldin, muloqot vaqtida va undan keyin foydalanuvchi tomonidan egallanadigan tajriba,
- ❖ UX dizayni foydalanuvchi tajribasi va xulq-atvoriga asoslangan interfeysni loyihalashdir.



UX (angl. user experience) so'zma-so'z "foydalanuvchi tajribasi" degan ma'noni anglatadi. Kengroq ma'noda, bu sayt yoki dastur bilan o'zaro aloqada foydalanuvchi olgan barcha tajribalar haqida tushunchadir. UX-dizayn funksiyalari, mahsulotning moslashuvchanligi va foydalanuvchilarga qanday his-tuyg'ular keltirishi uchun mas'uldir. Interfeys qanchalik aniq bo'lsa, foydalanuvchi natijani olish va maqsadli harakatni amalga oshirish osonroq bo'ladi.

Buyruq satri o'rniga oynalarni ishlatish g'oyasi 1984 yilgacha mavjud bo'lgan, ammo grafik interfeysni birinchi Apple dizaynerlari ommaviy ravishda taqdim etgan.

UX-dizayner (eng. user experience designer)-foydalanuvchilarning ehtiyojlarini o'rganadigan, interfeysning mantiqiy sxemalarini tuzadigan, maqsadli auditoriyada prototiplarni sinovdan o'tkazadigan va UI dizaynerlari uchun texnik topshiriqni tuzadigan dizayner.

UI dizayn

UI (*angl. user interface*) "foydalanuvchi interfeysi" deb tarjima qilinadi.

UI dizayni-foydalanuvchi tajribasi va maqsadli auditoriyani o'rganish asosida ishlab chiqilgan prototipni ko'rish jarayoni.

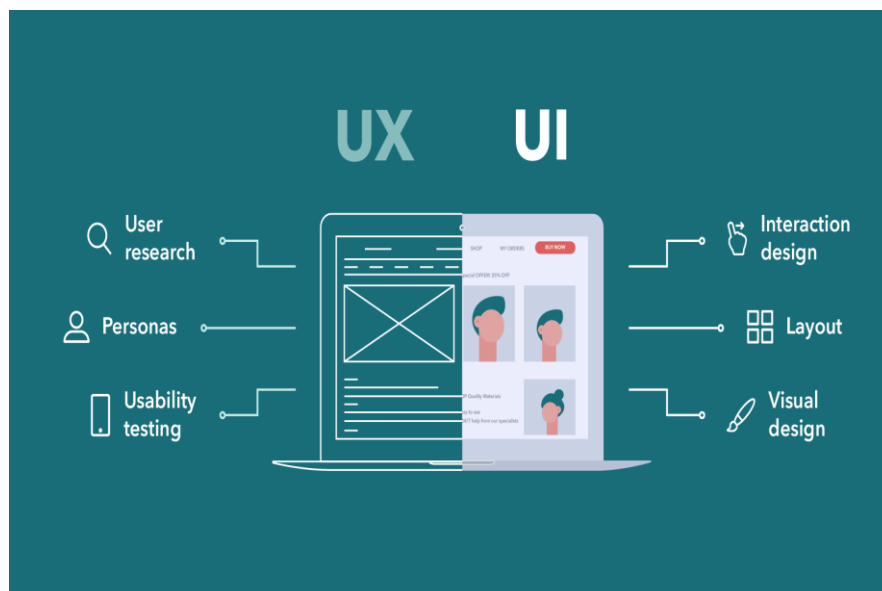
UI dizayni- interfeysning grafik qismida ishlashni o'z ichiga oladi: animatsiya, rasm, tugmalar, menyular, skanerlar, fotosuratlar va shriftlar.

UI-dizayner interfeysdagi ob'ektlarning rang palitrasi va joylashuvini belgilaydi: ochiladigan menyu to'g'ri ishlayaptimi yoki yo'qmi, "buyurtma" tugmasini bosish qulaymi, shaklni to'ldirish osonmi, smartfondan matn yaxshi o'qiladimi, bu yoki boshqa harakatlar uchun sayt qanday xabarni beradi kabi muammolarni hal qiladi

UI-dizaynerning asosiy vazifasi foydalanuvchini mahsulotni qanday ishlatishni tez va stresssiz tushunishga yordam berishdir: veb — sayt, dastur, dastur, to'lov terminali, mikroto'lqinli pech yoki masofadan boshqarish pulti. Buning uchun dizayner interfeysning asosiy talablarga javob berishiga ishonch hosil qiladi.

UX / UI dizayni Sifat talablari:

- ❖ Ravshanlik
- ❖ Qisqartirish
- ❖ Tanib olish
- ❖ Javobberish
- ❖ Doimiylik
- ❖ Estetika
- ❖ Samaradorlik
- ❖ Xushyoqish



Nazorat savollari

1. UX dizayn haqida ma'lumot bering?
2. UI dizayn haqida ma'lumot bering?

II BOB. KROSS PLTAFORMALI DASTURLASH TEXNOLOGIYALARI

Ushbu bobda Kross pltaformali dasturlash texnologiyalari: DART => Flutter texnologiyasi, Android Studioni o'rnatish va sozlash haqida umumiy ma'lumotlar keltirib o'tilgan.

2.1. Kross pltaformali dasturlash texnologiyalari: DART => Flutter texnologiyasi, Android Studioni o'rnatish va sozlash

Reja:

1. Kross platformali dasturlash texnologiyalari
2. DART => Flutter texnologiyasi
3. Android Studioni o'rnatish va sozlash

Mobil ilovalar bozori o'n yildan ortiq vaqtdan beri davom etmoqda, biroq u hali ham jadal rivojlanmoqda. Kompaniyalar tomonidan mobil ilovalarni yaratishga bo'lgan talab doimiy ravishda o'sib bormoqda va u hali ham taklifdan

sezilarli darajada oshib bormoqda, bu esa rivojlanishning doimiy xarajatlariga olib keladi. Ushbu jarayonni arzonlashtiradigan echimlardan biri bir xil dasturiy ta'minot kodi barcha platformalarda ishlatilganda o'zaro faoliyat platforma ishlab chiqishdir.

Odatda dastur yaratishda umuman dasturlashda siz Android uchun alohida va iOS / Apple uchun alohida , Windows uchun alohida , Web uchun alohida kod yozishingiz kerak. Masalan, quyida Operatsion tizimlar va ular uchun foydalaniladigan dasturlashtillarini ko'rishingiz mumkin.

- Apple iOS: Objective C yoki Swift
- Windows yoki Linux OS lariga desktop ilova yaratish uchun: C++, C#, Python.
- Android: Java yoki Kotlin.
- Windows Phone: C # va XAML .
- Web site uchun: HTML ,CSS, JavaScript , Python.

Afsuski, yuqoridagi tizimlarning alohida tillariga bo'lgan ehtiyoji quyidagi kamchiliklarga ega bo'lishi mumkin:

Xarajatlar - Har bir operatsion tizim uchun dastur yaratish va unga xizmat ko'rsatish ancha qimmatga tushadi.

Professionallik - Mobil ilovalarni dasturlash uchun 2-3 xil tilda yuqori darajadagi mahorat talab etiladi.

Bir xillik - Har bir platformaning o'z foydalanuvchi interfeysi, standartlashtirilgan vidjetlari va xususiyatlari bo'lgani uchun, ilovalar turli xil operatsion tizimlar uchun turlicha ko'rinishda bo'ladi va bu esa foydalanuvchi uchun salbiy qarashga olib kelishi mumkin.

Kross Platformalari orqali biz bir marta kod yozish orqali Android, iOS, Desktop va Web dasturlar yaratishimiz mumkin bo'ladi.

Kross Platformalari

Kross platformali ilova - bir vaqtning o'zida bir nechta operatsion tizimlarda ishlaydigan ilovalardir.

Kross platformada dasturchilarning vazifasi barcha operatsion tizimlarda yaxshi ishlaydigan kodni yozishdir.

Mahalliy va kross platformali ilovalarni ishlab chiqish o'rtasidagi farq.

Mahalliy-muayyan operatsion tizim uchun ishlab chiqilgan ilovalar. Muayyan muammolarni hal qilish uchun mos bo'lgan texnologiya(tillar) to'plami ishlatiladi.

Cross-platformaviy rivojlanish yangi biznes imkoniyatlarini ochadi va imkon qadar tezroq ishga tushirishga imkon beradi. Agar mahsulot juda murakkab bo'lmasa, aniq, muayyan vazifalarni bajarsa, bu tez va samarali boshlash uchun eng yaxshi yechimdir.

Ilovalar uchun 5 ta eng yaxshi kross platformalar

Xamarin

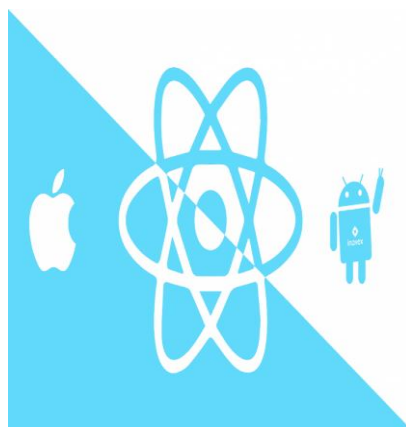
Microsoft kompaniyasi tomonidan yaratilgan ramka. U nafaqat turli mobil qurilmalar uchun ilovalarni, balki Windows uchun dasturlarni ham ishlab chiqishi mumkin. C# + .NET dasturlash tillarini qo'llab-quvvatlaydi. nisbatan yaqinda paydo bo'lgan, har qanday operatsion tizim bilan mos keladi va mahalliy platformalardan ajralib turadigan kross platformalarni yaratishga imkon beradi. IOs va Android uchun katta kutubxona, kompilyatorlar mavjud.



React Native

React Native-o'zaro kross platformali ilovalarni ishlab chiqish uchun ishlatiladigan eng mashhur fremvorklardan biri. U ilovaning mahalliy turini taqdim eta oladi.

Faqat elementlarni to'plash, ierarxiyani yaratish, uchinchi tomon xizmatlarini integratsiya qilish va React Native tilida interfeys yaratish kerak.



Flutter

Flutter o'z dasturlash tili **Dart**dan foydalanadi. Ishlab chiquvchilar ushbu fremwork bilan ishlash uchun uni o'rganishlari kerak bo'lsa-da, u rivojlanish uchun keng imkoniyatlar ochadi. Uning ishlash prinsipi oddiy: u turli platformalarda bitta kodni ishlatadi, lekin uni mahalliyga aylantirmaydi.



- | | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none">1. Imkoniyatlari:2. har qanday operatsion tizimni tushunadigan tilni ishlatadi;3. ko'pgina ishlab chiquvchilar ushbu platformaga o'tmoqdalar;4. oson o'rganish, katta hujjat kutubxonasi va faol qo'llab-quvvatlash;5. eng murakkab g'oyalarni amalga oshirishga imkon beruvchi keng vositalar to'plami;6. yuqori ishlash, ilovalar ishonchli va tezdir. | <ol style="list-style-type: none">1. Kamchiliklari:2. Visual muharriri juda yuqori sifatli emas;3. API juda xom, spetsifikatsiya "tezda" o'zgarishi mumkin;4. to'g'ri ishlashi kerak bo'lgan ba'zi narsalar hali ham xatolar chiqaradi;5. vidjetlar bilan qiyin ish, ko'pincha ularni nativada alohida-alohida ishlab chiqish kerak. |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Android Studio-Android mobil platformasi uchun ilovalarni ishlab chiqish vositasi bo'lgan dastur.

Android Studio raqobatchidan ko'p jihatdan ustun turadi:

- rivojlanish muhitining moslashuvchanligi;
- katta xususiyatlar to'plami;
- ishlab chiquvchiga moslashtirilgan rivojlanish jarayoni.

Android operatsion tizimi uchun ilovalar va kommunal yaratish paytida, dasturiy ta'minot foydalanuvchi Real vaqtda loyiha o'zgarishlarni ko'rish imkoniyati mavjud.

Android Studio-bu universal rivojlanish muhiti, chunki u kelajakdagi ilovalarning nafaqat smartfonlarda, balki planshetlarda ham, ushbu operatsion tizim asosida ishlaydigan ko'chma kompyuterlarda ishlash uchun optimallashtirishga imkon beradi.

Android Studio Xususiyatlari

Dastur turli xil ekranlarga ega bo'lgan qurilmalarda ilovaning to'g'ri ishlashini tekshirish imkonini beruvchi emulyatorni o'z ichiga oladi. 18:9 aspekt nisbati bilan ekranlar o'rnatilgan smartfonlarning tendentsiyalariga kirgandan so'ng, bu xususiyat ayniqsa dolzarb bo'lib qoldi.

Emulyatorning o'ziga xos xususiyati-ilovani eng mashhur qurilmalarda ishga tushirishda taxminiy ishlash ko'rsatkichlarini ko'rish.

Eng so'nggi versiya Android Studio ilovalari uchun ishlab chiqish muhiti yangi boshlanuvchilar uchun ham juda qulay bo'ldi. Dastur kodni qadoqlash, uni etiketlash uchun barcha zamonaviy vositalarni amalga oshiradi. Ko'plab ijodkorlar tomonidan talab qilingan Drag-N-Drop funktsiyasi, komponentlarni to'g'ridan-to'g'ri rivojlanish muhitiga ko'chirishni osonlashtiradi.

Ilovalarni lokalizatsiya qilish Android Studio-ning afzalliklari ro'yxatiga kiritilgan SDK funktsiyasi bilan ancha osonlashadi.

Dasturning maqsadli auditoriyasi bilan aloqa o'rnatilgandan so'ng, Google – Google Cloud Messaging-dan vositani amalga oshirishga yordam beradi.

Yordam dasturining afzalliklari

- rivojlanish muhiti bir nechta dasturlash tillari bilan ishlashni qo'llab-quvvatlaydi, ular orasida eng mashhur – C/C++, Java.
- ishlash uchun qulay bo'lgan kod muharriri;
- faqat smartfonlar/planshetlar uchun emas, balki portativ kompyuterlar, Android TV televizorlari uchun konsollar, Android Wear qurilmalari, g'ayrioddiy ekran ko'rinishidagi yangi mobil qurilmalar uchun ilovalarni ishlab chiqish imkonini beradi;
- yangi o'yinlar, kommunal xizmatlarning to'g'riligini, ularning bir yoki bir nechta tizimda ishlashini sinab ko'rish to'g'ridan-to'g'ri emulyatorida amalga oshiriladi;
- tayyor kodni qayta tiklash;
- dasturiy ta'minotni ishlab chiqish uchun tayyor andozalar va komponentlar bilan juda katta kutubxona;
- Android n ilovasini ishlab chiqish – operatsion tizimning eng so'nggi versiyasi;
- unda xatolar uchun allaqachon yaratilgan dasturni oldindan tekshirish;
- ilovaning har bir elementini, o'yinni sinab ko'rish uchun katta vositalar to'plami;
- tajribasiz / boshlang'ich ishlab chiquvchilar uchun dasturning rasmiy veb-saytida joylashtirilgan Android studiyasidan foydalanish bo'yicha maxsus qo'llanma yaratilgan.

Kamchiliklar

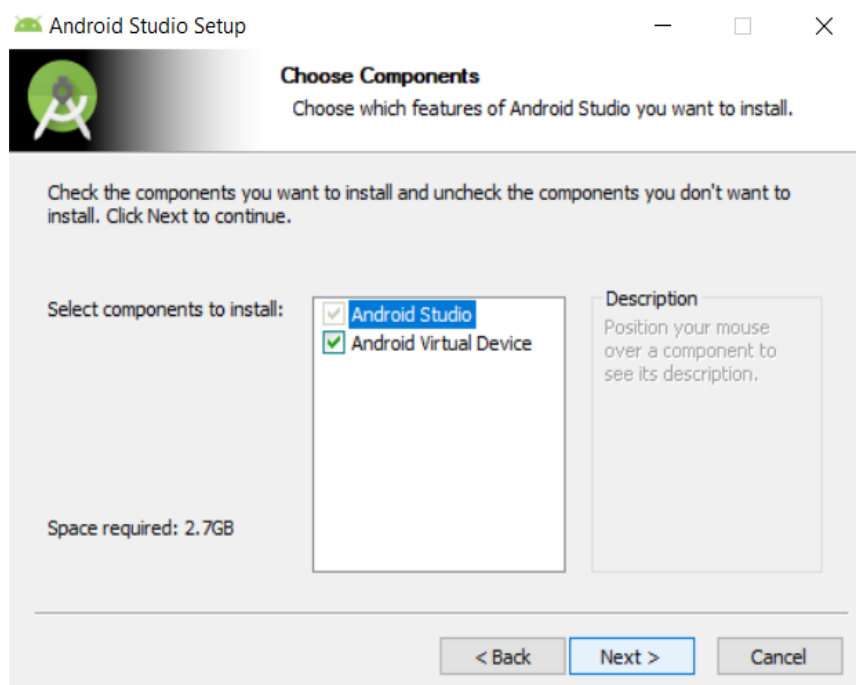
Rivojlanish muhitida o'rnatilgan Android emulyatoriga ega bo'lishiga qaramasdan, yangi ishlab chiqilgan dasturni sinovdan o'tkazishda qiyinchiliklar bo'lishi mumkin. Shunday qilib, uni ishga tushirish uchun testni rejalashtiradigan kompyuterning juda ta'sirli apparat bazasi kerak.

Yana bir kamchilik - kompyuter, Android qurilmalari uchun Java-da server loyihalarini yozish imkoni yo'q.

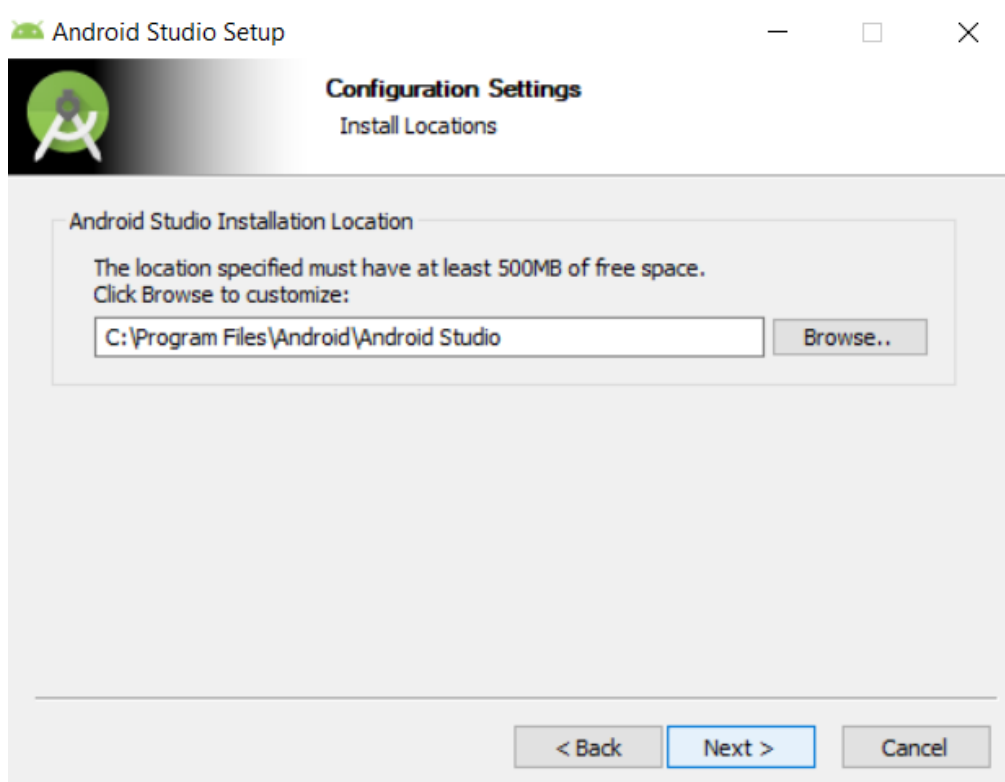
Android Studioni o'rantish jarayoni



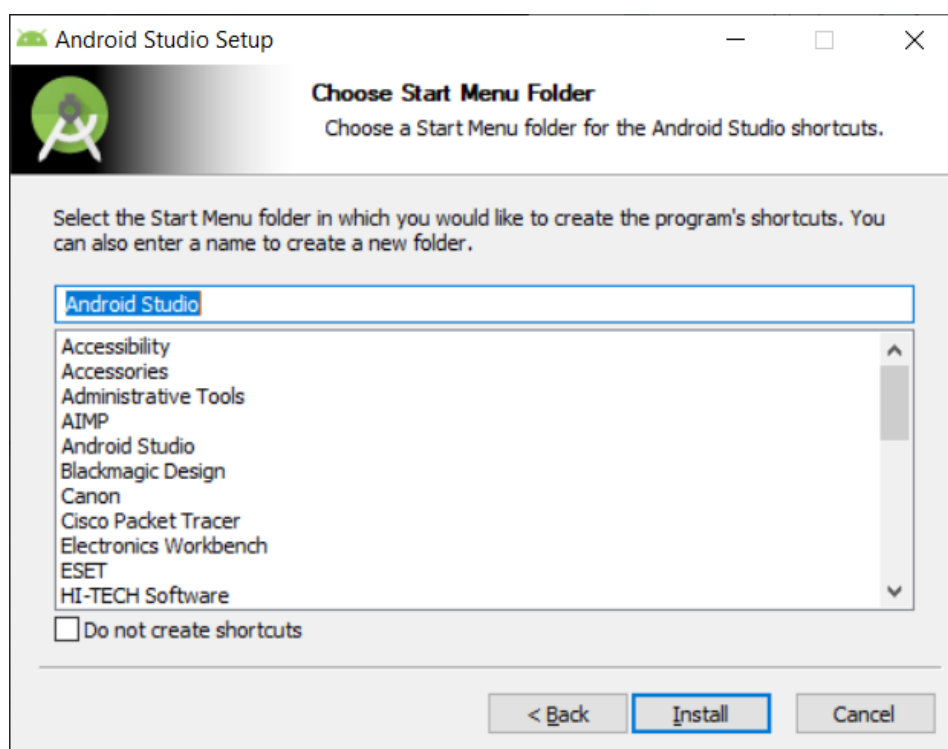
NEXT tugmasini bosamiz.



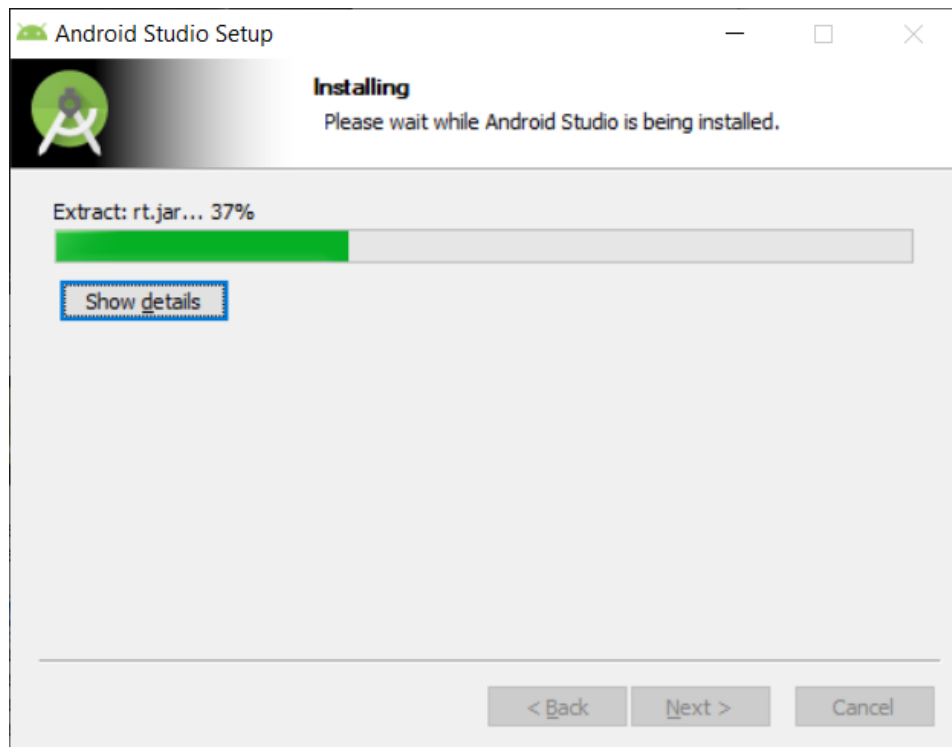
Kerakli sohani belgilab, **NEXT** tugmasini bosamiz.



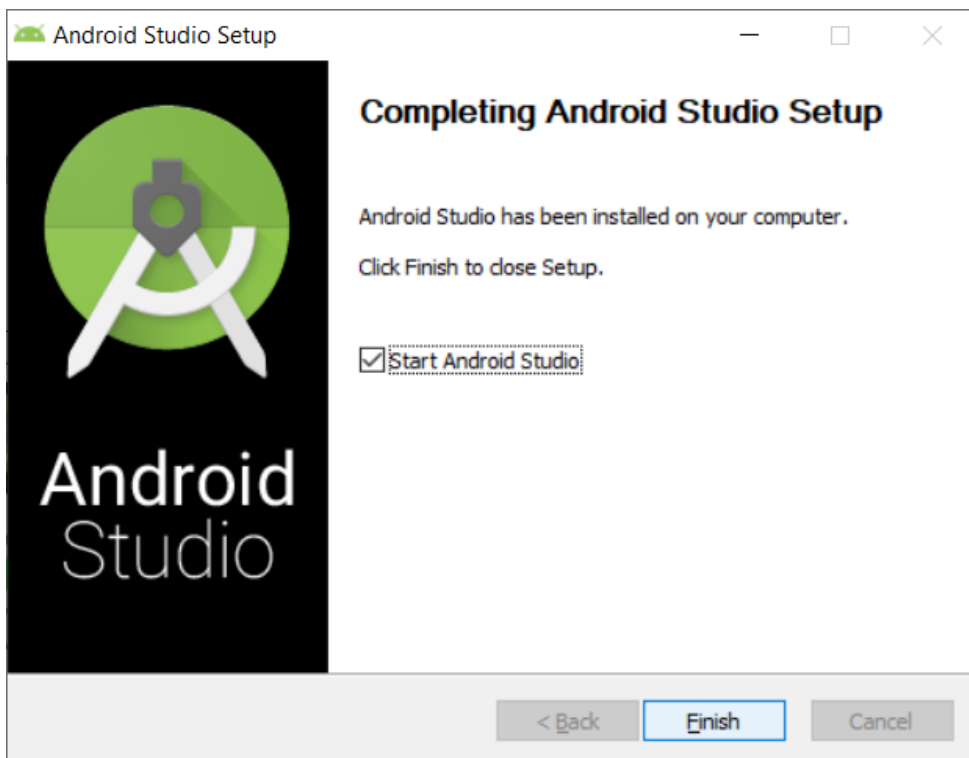
Dastur o'rnatilishi kerak bo'lgan joy tanlanadi (C diskka dasturlar o'rnatiladi.).



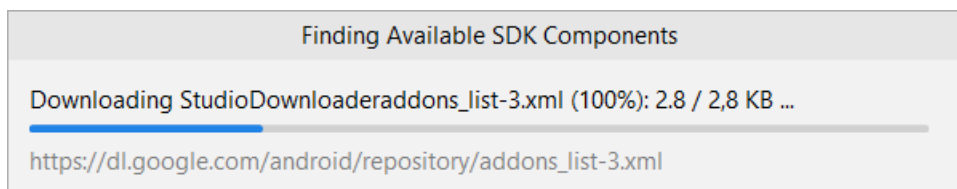
Android Studio nomli papka bilan, **INSTALL** qilamiz.



Dastur o'rnatilish jarayoni.



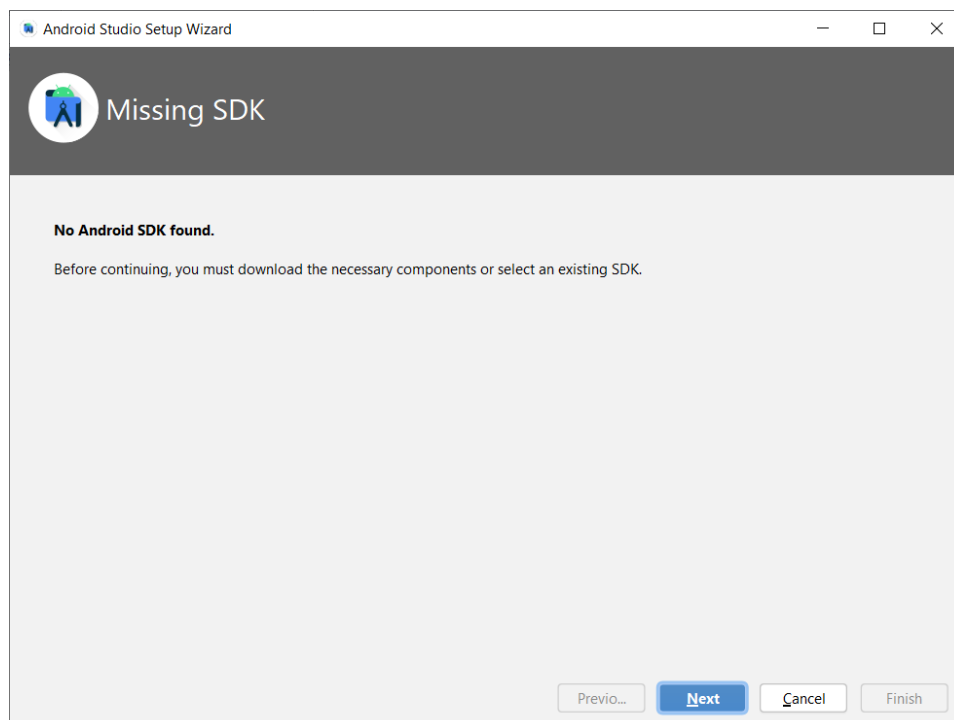
Dasturni o'rnatib bo'ldik.

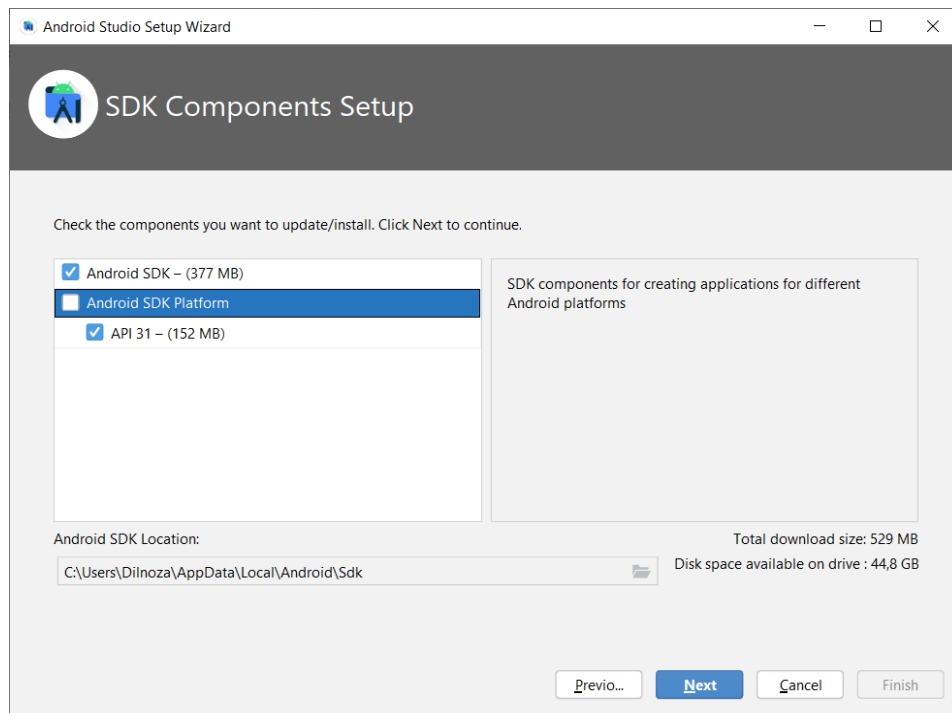


Dastur ochiladi.



Kerakli dasturiy ta'monot qismlarini o'rnatib olamiz.





Nazorat savollari

1. Kross platformali dasturlash texnologiyalari deganda nimani tushunasiz?
2. DART => Flutter texnologiyasi haqida ma'lumot bering?
3. Android Studioni o'rnatish va sozlash ketma-ketligini tushuntiring?

2.2. Flutter: Android Studio interfeysi, ilk dasturni yozish va uni emulyatorda ishlatish, komponentalar bilan tanishish

Reja:

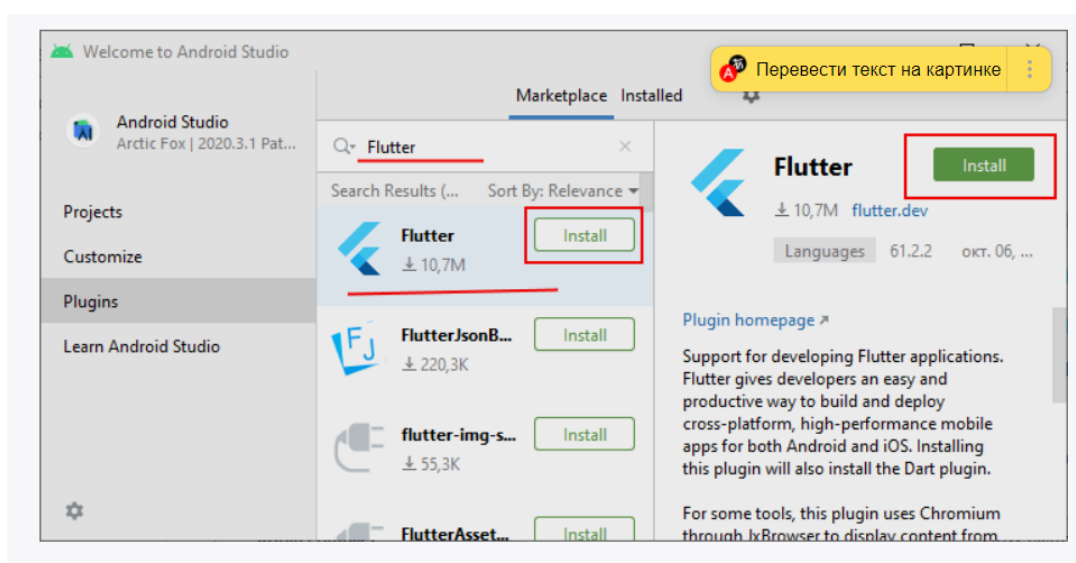
1. Flutter: Android Studio interfeysi
2. Android Studio da ilk dasturni yozish va uni emulyatorda ishlatish
3. Android Studio komponentalari bilan tanishish

Flutter Android Studio da eng oddiy matn muharriridan kodni kiritishimiz va uni konsolda tuzish imkoniyati mavjud mumkin bo'lsa-da, lekin rivojlanish muhiti dasturni yozish va qurish jarayonini sezilarli darajada kamaytirishga imkon beradi. Bundan tashqari, Android Studio Flutter-da faqat Android uchun emas,

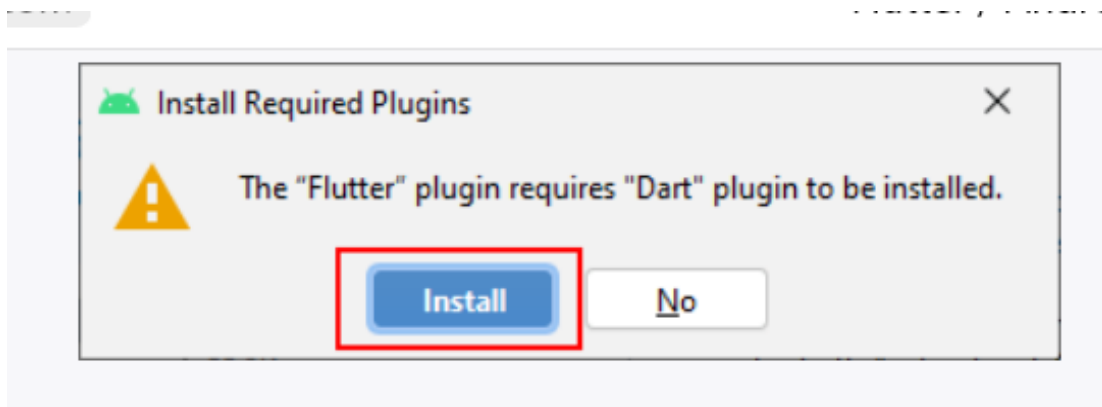
balki boshqa qo'llab-quvvatlanadigan platformalar uchun ilovalarni yaratishga imkon beradi.

Android Studio bilan ishlash uchun, avvalo, uni o'rnatish kerak. O'rnatish havolasini yuklab olish kerak <https://developer.android.com/studio>.

Odatda Android Studio Flutterni qo'llab-quvvatlamaydi, shuning uchun tegishli plaginni o'rnatishimiz kerak. Buni amalga oshirish uchun boshlang'ich ekranda Android Studio-da Plugins-ni tanlang (yoki file - > Settings> menyusiga o'ting va keyin ochilgan oynada Plugins-ni tanlang). Va pluginlar panelida Flutter plaginini topamiz:

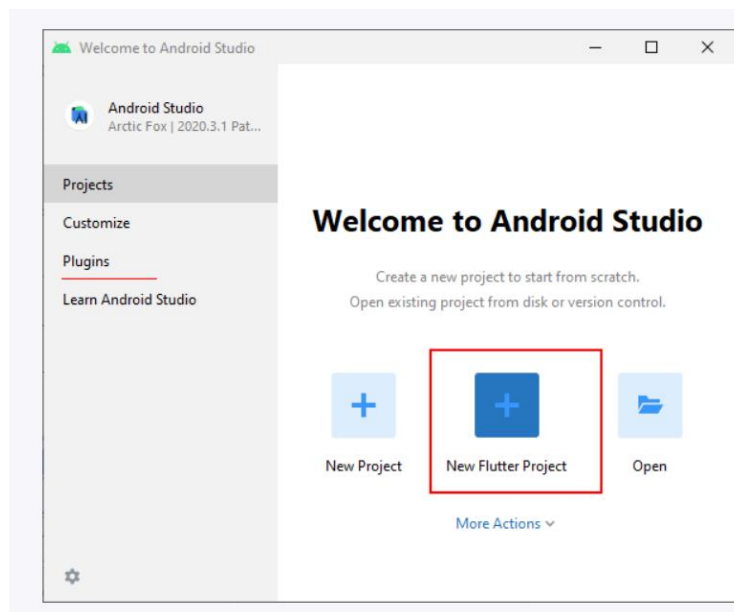


Kerakli plaginni qidirishni soddalashtirish uchun qidiruv tizimiga "Flutter" so'zini kiritishimiz mumkin va birinchi natija faqat o'rnatilishi kerak. Plaginni o'rnatishda dart uchun plaginni o'rnatish taklifi bilan oyna ham ko'rsatiladi. Bundan tashqari, uni o'rnatish uchun OK ni bosing:



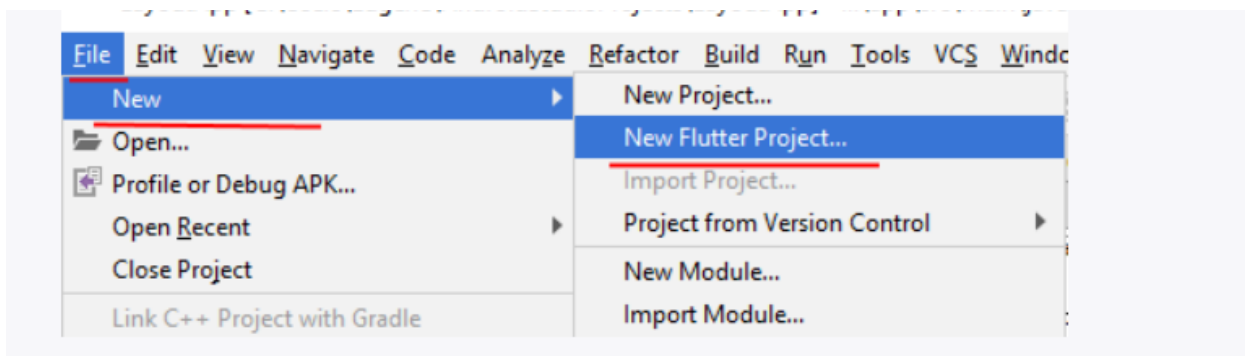
Plaginni o'rnatganingizdan so'ng Android studiyasini qayta ishga tushirishingiz kerak bo'ladi.

Android Studio-da boshlang'ich ekranda qayta ishga tushirilgandan so'ng, biz yangi Flutter loyihasi tugmachasini ko'rishimiz mumkin:

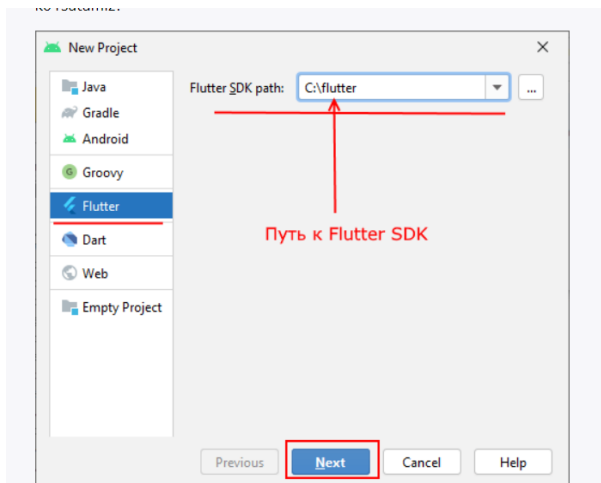


Flutter ostida loyiha yaratish uchun ushbu tugmani bosing.

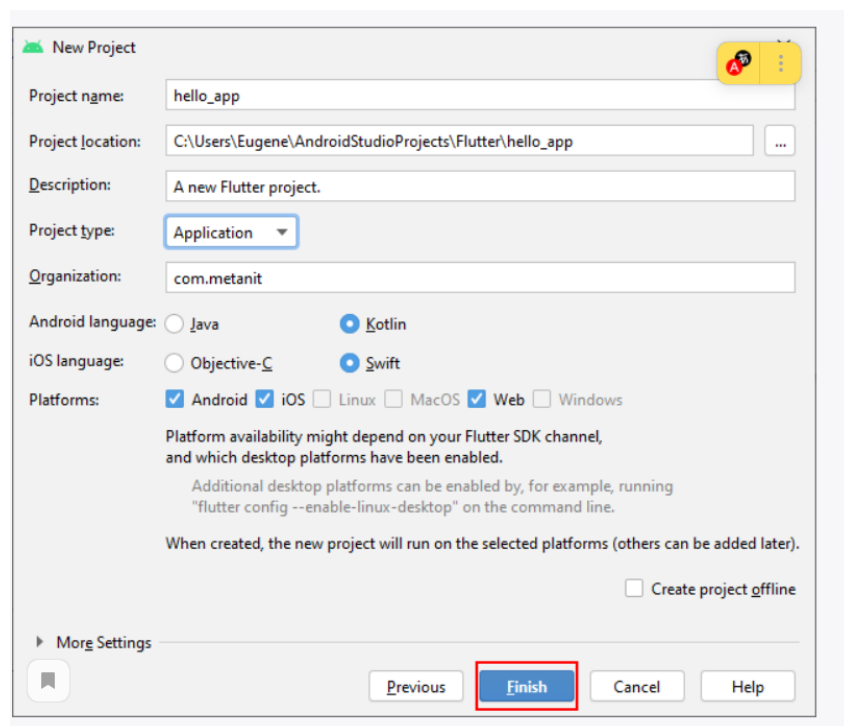
Shu bilan bir qatorda, studiyada loyiha yaratish uchun menyuga o'ting fayl - > yangi - > > yangi Flutter loyihasi > >:



Keyin biz yangi loyiha yaratish oynasini ochamiz. Chap tomonda **Flutter elementini tanlangva** flutter SDK yo'lidagi markazda **flutter SDK yo'lini** ko'rsatamiz:



Keyingi oynada bir qator loyiha parametrlarini ko'rsatamiz:



Loyiha nomi sohasida loyiha nomini beramiz. hello_app Agar taklif qilingan standart manzil mos kelmasa, loyiha joylashgan joyida loyiha manzilini o'zgartirish imkoniyati mavjud

Description maydonida loyihaning tavsifi ko'rsatilishi mumkin

Loyiha turi sohasida loyiha turi ko'rsatiladi. Odatiy bo'lib, u dastur qiymatiga ega (ya'ni, loyiha dasturni yaratish uchun mo'ljallangan). Ushbu standartni qoldiring.

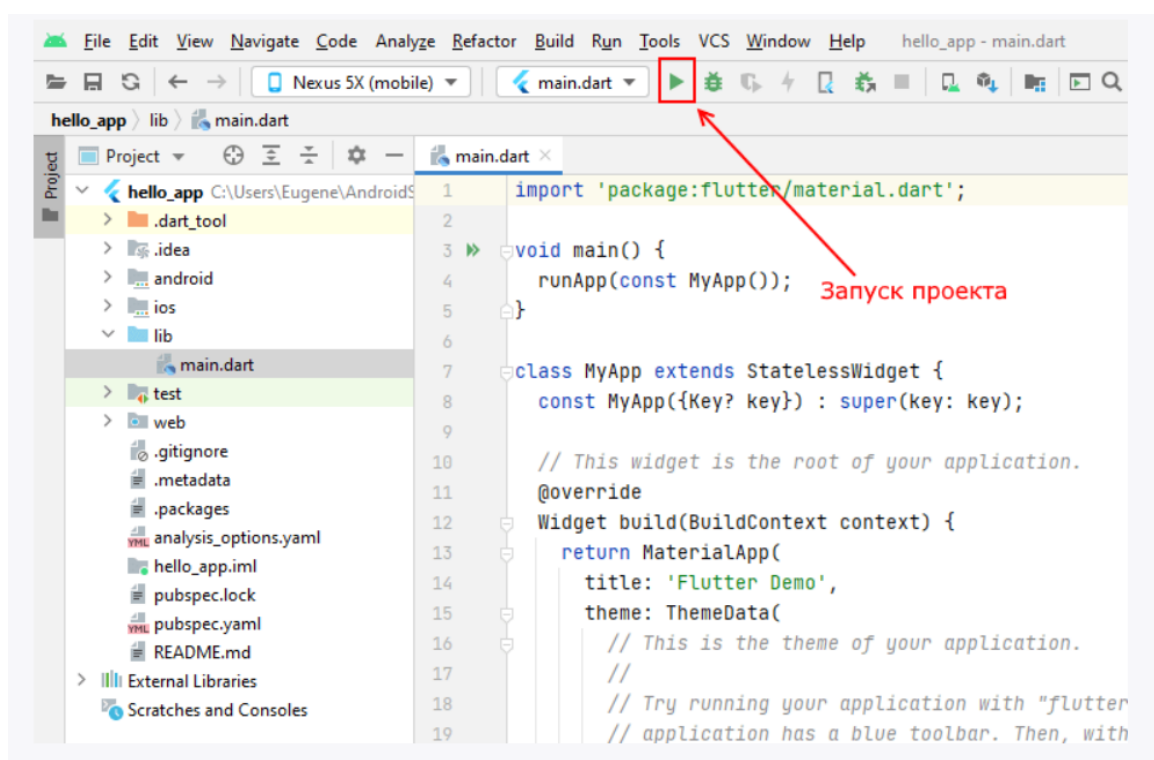
Android language maydoni Android uchun tilni ko'rsatadi. Asl qiymati qoldirish mumkin - Kotlin.

IOS language sohasida platforma iOS uchun til ko'rsatiladi. Asl qiymati qoldirish mumkin - Swift.

Platformalar sohasida siz loyihani yaratadigan platformalarni belgilashingiz mumkin.

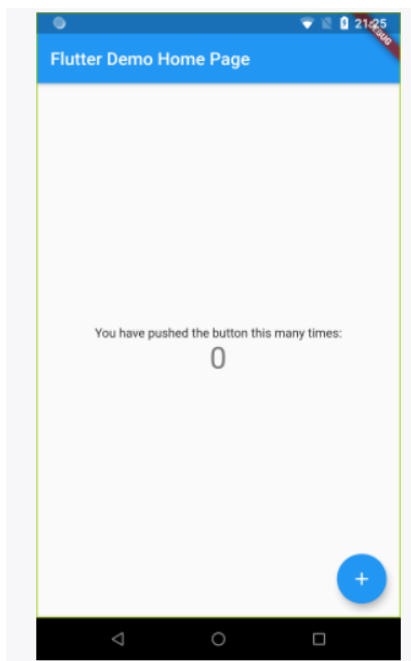
Odatda Android va iOS elementlari belgilanadi, ammo boshqa mavjud platformalarni tanlash mumkin.

Va keyin barcha sozlamalarni o'rnatganingizdan so'ng, loyihani to'g'ridan-to'g'ri yaratish uchun "tugatish" tugmasini bosing. Yaratilganidan so'ng, Android Studio yaratilgan loyihani ochadi:

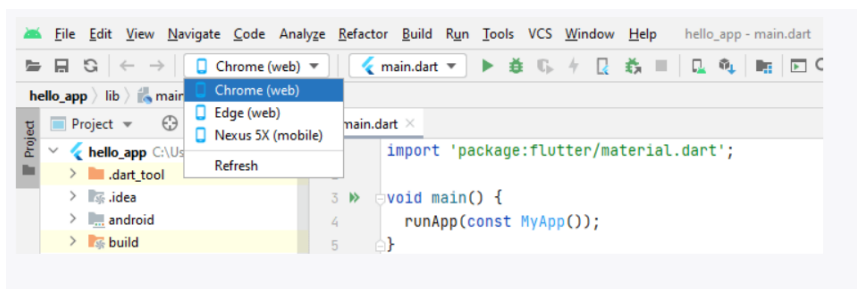


Yaratilgan loyiha jamoa yordamida konsolda o'tgan mavzuda yaratilgan bir xil tuzilishga ega bo'ladi flutter create. Studiya markazida asosiy fayl ochiladi.to'g'ri dastur kodini o'z ichiga olgan dart.

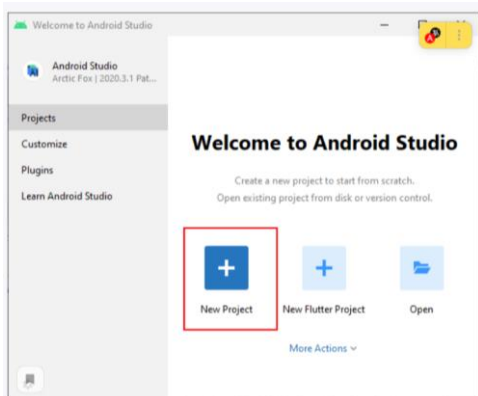
Android qurilmasini kompyuterga ulang (yoki emulyatorlardan foydalaning) va Android Studio panelida dasturni ishga tushirish uchun yashil tugmani bosing.



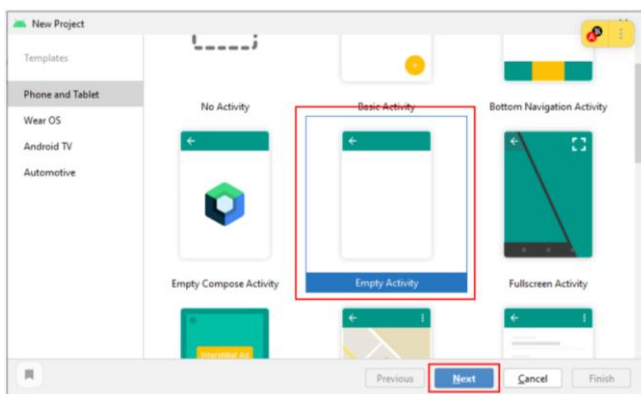
Xuddi shunday, Android studiyasida siz boshqa "qurilmalar" uchun, masalan, veb-sayt ostida loyihani ishga tushirishimiz mumkin. Buning uchun faqat asboblari panelidagi tegishli qurilmani tanlashingiz kerak:



Android operatsion tizimi uchun Android Studio muhitida birinchi ilovani yaratamiz. Android studiyasini oching va boshlang'ich ekranda **yangi loyihani tanlang**:

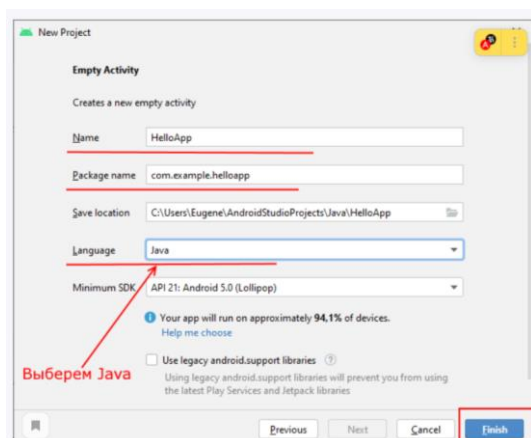


Android Studio loyihasini yaratishda dastlab loyiha shablonini tanlashni taklif qilamiz:



Android Studio turli xil vaziyatlar uchun bir qator shablonlarni taqdim etadi. Activity shablonini tanlang va keyingi tugmani bosing.

Shundan so'ng, yangi loyiha sozlamalari oynasi ko'rsatiladi:



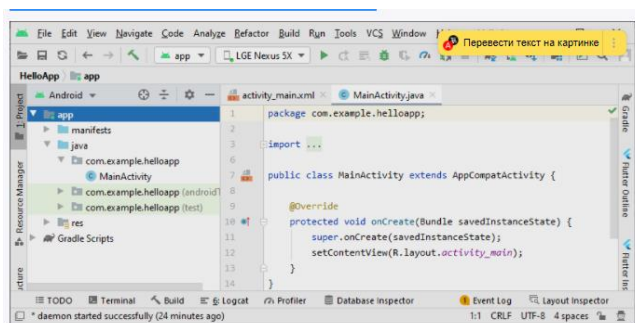
Yangi loyihani yaratish oynasida uning dastlabki sozlamalarini o'rnatishimiz mumkin:

- Ism maydonida ilovaning nomi kiritiladi. HelloApp nomini ko'rsatamiz
- Paket nomi maydonida ilovaning asosiy klassi joylashgan paketning nomi ko'rsatiladi. Bunday holda, test loyihalari uchun bu qiymat katta ahamiyatga ega emas, shuning uchun biz maqomni o'rnatamiz.example.helloapp.
- Save JoyXarita maydonida loyiha fayllarining joylashuvi qattiq diskda o'rnatiladi. Standart qiymatni qoldirish mumkin.
- Til sohasida dasturlash tili sifatida Java-ni ko'rsatamiz .

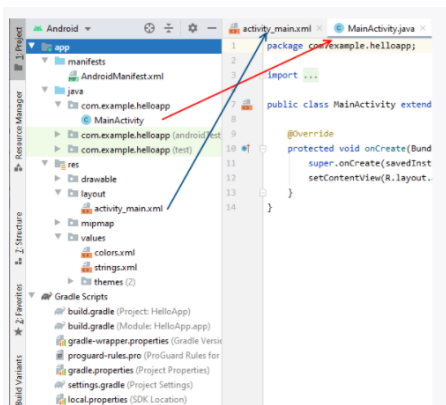
- Minimal SDK maydonida SDKNING eng kam qo'llab-quvvatlanadigan versiyasi ko'rsatiladi. Standart API 21: Android 5.0 (Lollipop) ni qoldiring, ya'ni bizning ilovamiz android 5.0 dan boshlab ishga tushirilishi mumkin, ya'ni qurilmalarning 94%. Qadimgi qurilmalarda ishga tushirish mumkin bo'lmaydi.

Shuni esda tutish kerakki, SDK versiyasi qanchalik baland bo'lsa, qo'llab-quvvatlanadigan qurilmalar soni kamroq.

Keyin Finish tugmasini bosamiz va Android Studio yangi loyihani yaratadi:



Birinchidan, loyihaning tuzilishini qisqacha ko'rib chiqamiz.



Android loyihasi turli modullardan iborat bo'lishi mumkin. Odatda, biz loyihani yaratganimizda, bitta modul-app yaratiladi. Modul uchta pastki papkaga ega:

- manifestlar: AndroidManifest manifest faylini saqlaydi.xml, dastur konfiguratsiyasini tavsiflaydi va ushbu dasturning har bir tarkibiy qismini aniqlaydi.

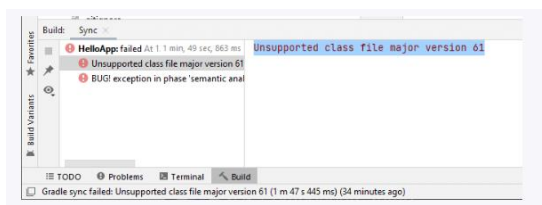
- java: java-da kod fayllarini saqlaydi, ular alohida paketlar bo'yicha tuzilgan. Shunday qilib, papkada com.example.helloapp(uning nomi loyihani yaratish bosqichida ko'rsatilgan) dasturda MainActivity fayli mavjud. Java kodi Java tilida, ya'ni dastur boshlanganda dastur ishga tushirilgan MainActivity sinfini ifodalaydi
- res: ilovada ishlatiladigan resurslarni o'z ichiga oladi. Barcha resurslar pastki papkalarga bo'linadi.
 - chizilgan papka ilovada ishlatiladigan tasvirlarni saqlash uchun mo'ljallangan
 - layout papkani grafik interfeysni aniqlaydigan fayllarni saqlash uchun mo'ljallangan. Odatda, bu erda activity_main fayli mavjud.xml, asosiy sinf uchun interfeysni xml sifatida belgilaydi
 - mipmap papkalari turli xil ekran ruxsati bilan ilova belgilarini yaratish uchun mo'ljallangan rasm fayllarini o'z ichiga oladi.
 - values papkasida resurslar to'plamlarini o'z ichiga olgan turli xil xml fayllari saqlanadi-ilovada qo'llaniladigan turli ma'lumotlar. Odatda, bu erda ikkita fayl va bitta papka mavjud:
 - Kesh / <url>xml ilovada ishlatiladigan ranglarning tavsifini saqlaydi, faylga yozish mumkin emas.
 - Kesh / <url>xml ilovada ishlatiladigan string resurslarini o'z ichiga oladi, faylga yozish mumkin emas.
 - mavzular papkalari ikkita dastur mavzusini - yorug'lik (kunduzi) va qorong'i (tungi)

Gradle Scriptsning alohida elementi dasturni qurishda ishlatiladigan bir qator skriptlarni o'z ichiga oladi.

Ushbu tuzilishda MainActivity faylini tanlashimiz kerak bo'ladi. Android Studioda ochilgan va ilovaning mantig'ini o'z ichiga olgan java va aslida ilovaning bajarilishi boshlanadi.

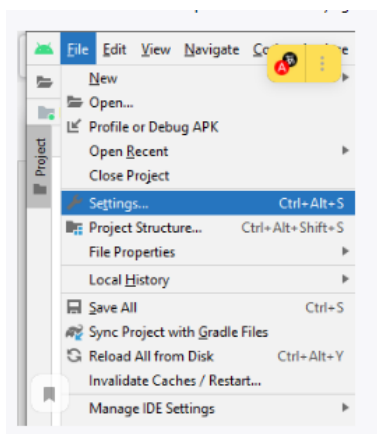
Paydo bo'lishi mumkin bo'lgan muammolar

Java dasturini yaratish uchun ishlatiladi. Va dasturni yaratish uchun Gradle infratuzilmasi qo'llaniladi. Biroq, Gradle ning joriy ishlatilgan versiyasi standart JDK versiyasiga mos kelmasligi mumkin. Va bu holda, Android Studio, masalan, unsupported class file xatosi major version 61 kabi xatolarni ko'rsatishi mumkin:

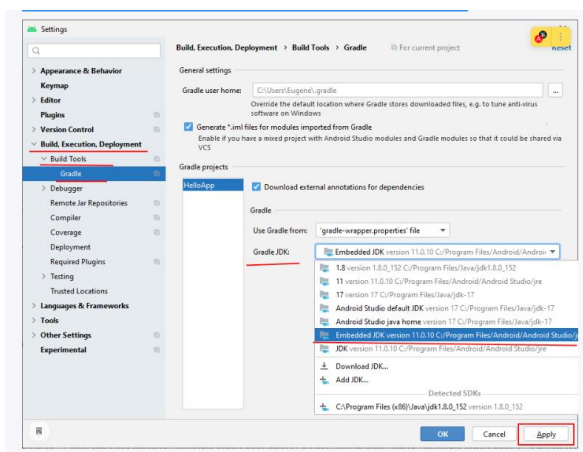


Ushbu xato JDK 17 versiyasi joriy Gradle versiyasiga mos kelmasligini aytadi. Va biz kichikroq versiyani ishlatishimiz kerak.

Ushbu muammoni hal qilish uchun studiyaga File ->Settings > (MacOS-da Android Studio - > Preferences>)



Keyin ochilgan Sozlamalar oynasida Menyu **elementiga o'ting Build, ijro etish, Deployment - > Build Tools - > > Gradle>>** va keyin JDK versiyasini o'zgartiradigan Gradle JDK maydonini toping. 11 va undan yuqori versiyalari bo'lishi kerak. Odatda, Android Studio bilan birga JDK-ning o'rnatilgan va qo'llab-quvvatlanadigan versiyasi-hozirgi vaqtda jdk 11. Va u JDK ro'yxatida tanlanishi mumkin:



Android Studio bilan birga keladigan JDK versiyalarini tanlash uchun eng maqbul nuqta "embedded JDK version" deb ataladi.... Ekran tasvirida ko'rinib turganidek, bu versiya 11, lekin keyingi Android Studio yangilanishlari bilan ushbu versiya o'zgarishi mumkin.

O'zgarishlardan so'ng, birinchi OK tugmasini keyin Apply tugmasini bosamiz va loyihani ishga tushirishni takrorlang.

Loyihani ishga tushirish

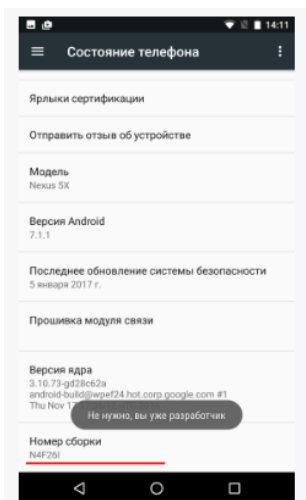
Yuqorida yaratilgan loyiha allaqachon funkcionallikni o'z ichiga oladi. To'g'ri, bu funkcionallik deyarli hech narsa qilmaydi, faqat " Salom dunyo!". Shunga qaramay, bu aslida biz boshlashimiz mumkin bo'lgan dastur.

Dasturni ishga tushirish va sinovdan o'tkazish uchun emulyatorlarni yoki haqiqiy qurilmalardan foydalanishimiz mumkin. Lekin ideal holda, haqiqiy qurilmalarda sinovdan o'tish yaxshiroqdir. Bundan tashqari, emulyatorlar katta apparat resurslarini talab qiladi va har bir kompyuter emulyatorlarning talablarini tortib ololmaydi. Sinov uchun mobil qurilmadan foydalanish uchun faqat kerakli drayverni o'rnatish kerak bo'lishi mumkin.

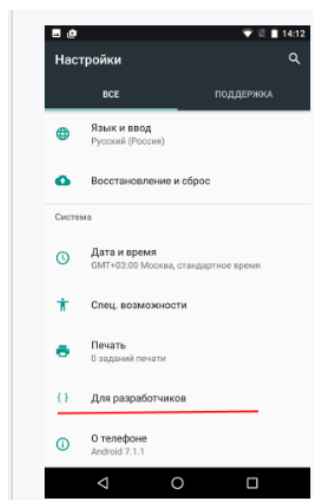
Telefonda ishlab chiquvchi rejimi

Odatda smartfonlardagi ishlab chiquvchi imkoniyatlari yashiringan. Bu imkoniyatni ochish quyidagi ketma-ketlik orqali amalga oshiriladi

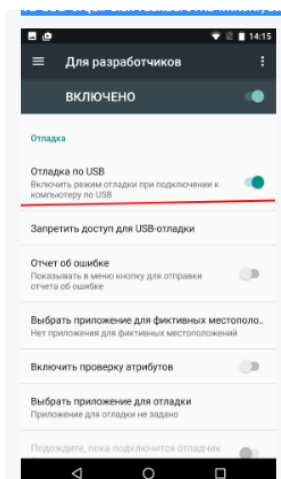
Sozlamalar > telefon Haqida (Settings > telefon Haqida) (Android 8 bo'ladi Sozlamalar > Tizim > telefon Haqida (Sozlamalar > Tizim > telefon Haqida)) va matbuot yetti marta bo'yicha Qurish Soni (qurish Soni).



Endi USB disk raskadrovka usulini yoqishingiz kerak. Buni amalga oshirish uchun Sozlamalar > System > Advanced > > > Developer options > > yoki Sozlamalar > SYSTEM > qo'shimcha > > > ishlab chiquvchilar uchun > > (Android 8-da Sozlamalar > System > > Developer options > > yoki Sozlamalar > SYSTEM > ishlab chiquvchilar uchun > >).

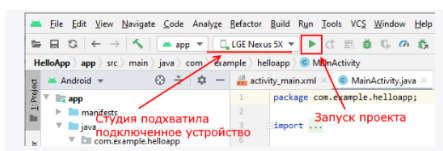


USB orqali disk raskadrovka imkoniyatini yoqing:

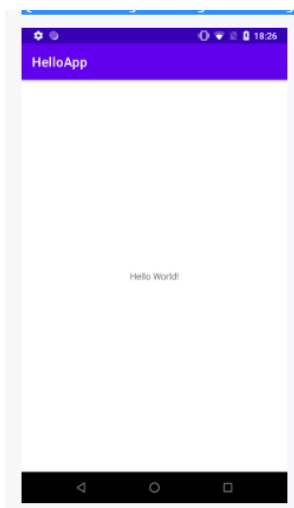


Иловани ishga tushirish

Qurilmani Android OS bilan ulang (agar biz haqiqiy qurilmada sinab ko'rsak) va asboblardan panelidagi yashil tugmani bosish orqali loyihani ishga tushiring.



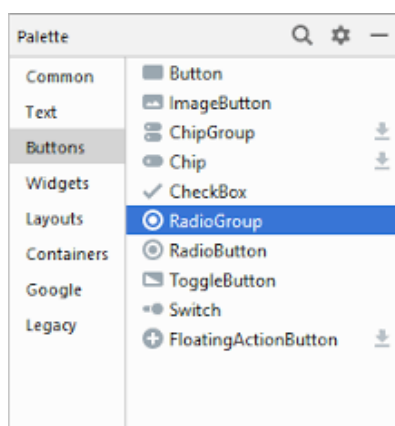
Qurilmani tanlang va OK tugmasini bosing. Va ishga tushirilgandan so'ng, biz ilovani qurilma ekranida quyidagicha ko'ramiz



RadioGroup

Ushbu sinf radio tugmalari to'plami uchun bir nechta istisno doirasini yaratish uchun ishlatiladi. Radio guruhiga tegishli bitta radio tugmachani tekshirish shu guruh ichidagi ilgari tekshirilgan radio tugmachasini tekshirmaydi.

Muayyan radio tugmachasini olib tashlashning iloji bo'lmasa-da, tekshirilgan holatni olib tashlash uchun radio guruhini tozalash mumkin.



RadioButton

Radio tugmasi - bu tekshiriladigan yoki belgilanmagan ikkita holat tugmachasi. Radio tugmasi belgilanmaganida, foydalanuvchi uni bosish yoki bosish orqali tekshirishi mumkin.



ToggleButton

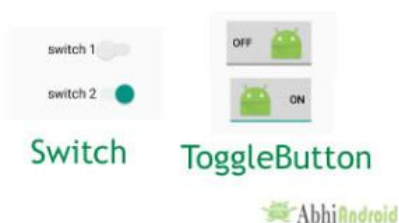
toggleButton foydalanuvchiga ikkita holat orasidagi sozlamani o'zgartirishga imkon beradi.

[ToggleButton](#) Ob'ekt bilan tartibingizga asosiy o'tish tugmachasini qo'shishingiz mumkin . Slayderni boshqarishni ta'minlaydigan o'chirgich deb

nomlangan boshqa bir tugma tugmachasini taqdim etadi va uni [Switch](#) ob'ekt bilan qo'shishingiz mumkin .

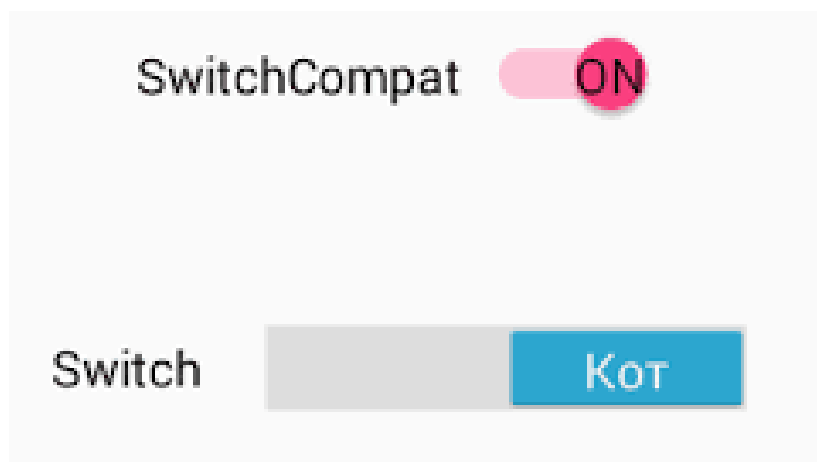
Agar tugmachaning holatini o'zgartirish kerak bo'lsa,

[CompoundButton.setChecked\(\)](#) yoki [CompoundButton.toggle\(\)](#) usulidan foydalanishingiz mumkin .



Switch

Switch - bu ikkita variantni tanlashi mumkin bo'lgan ikkita holatni o'zgartirish tugmachasi vidjeti. Tanlangan variantni tanlash uchun imkoniyatini beradi.



TextView-eng ko'p ishlatiladigan komponentlardan biri. Uning yordami bilan foydalanuvchi dasturda harakat qilish uchun qulayroqdir.

Matnni foydalanuvchiga ko'rsatadigan foydalanuvchi interfeysi elementi. *Foydalanuvchi tomonidan tahrirlanadigan matnni taqdim etish imkoniyatini beradi.*

Quyidagi kod namunasi matn ko'rinishining tarkibini o'zgartirish uchun XML tartibi va kodi bilan berilgan:

<LinearLayout

```
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
        <TextView
            android:id="@android:layout_height="wrap_content"
            android:layout_width="wrap_content"
            android:text="@string/hello" />
    </LinearLayout>
```

Код TextView в XML:

```
<TextView android:id="@+id/simpleTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="AbhiAndroid" />
```

id: id - bu matn ko'rinishini aniqlash uchun ishlatiladigan atribut. Quyida biz matn ko'rinishining identifikatorini o'rnatadigan namunaviy kod mavjudni keltirib o'tamiz:

```
<TextView
    android:id="@+id/simpleTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
    layout_width="wrap_content"
    layout_height="wrap_content"/>
```

gravitatsiya atributi - bu chapga, o'ngga, markazga, yuqoriga, pastga, center_vertical, center_horizontal va hk kabi matnlarning joylashuvini boshqarish uchun ishlatiladigan **ixtiyoriy atribut**.

Quyida tushuntirishga ega namuna kodi keltirilgan bo'lib, unda TextView matni uchun center_horizontal gravitatsiyani o'rnatilgan:

```
<TextView
```

```

android:id="@+id/simpleTextView"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="AbhiAndroid"
android:textSize="20sp"
android:gravity="center_horizontal"/> <!--center horizontal gravity-->

```

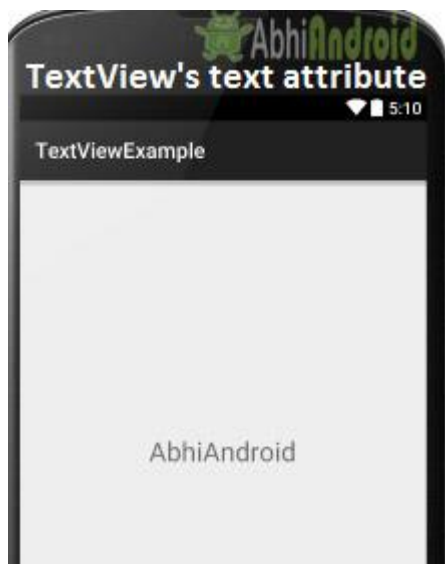


text: Matn atributi matnni namoyish qilishda matnni o'rnatish uchun ishlatiladi. Matnni xml va java sinfida o'rnatishimiz mumkin. Quyida biz "AbhiAndroid" matnini matn ko'rinishida joylashtirgan tushuntirish bilan namuna kodini keltiramiz:

```

<TextView
    android:id="@+id/simpleTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerInParent="true"
    android:textSize="25sp"
    android:text="AbhiAndroid"/>><!--отобразить текст как AbhiAndroid-->

```



`textColor`: `textColor` atributi matn ko'rinishini matni rangini belgilash uchun ishlatiladi. Rang qiymati "#argb", "#rgb", "#rrggbb" yoki "#aarrggbb" shakllariga ega.

Quyida biz tushuntirilgan kod namunasi bo'lib, unda biz ko'rsatilgan matnning rangini qizil rangga o'rnatdik.

```
Android:параметр textSize="25sp"
```

```
Android:Андроид свойства textColor="#также F00"/><!--red color for text view-->
```

`textSize`: `textSize` atributi matn ko'rinishini matni hajmini belgilash uchun ishlatiladi. Matn hajmini sp (masshtabdagi mustaqil piksel) yoki dp (zichlik pikseli) ga o'rnatishimiz mumkin.

```
<TextView
```

```
android:id="@+id/simpleTextView"
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:text="AbhiAndroid"
```

```
android:layout_centerInParent="true"
```

```
android:textSize="40sp" /><!--Set size-->
```

textStyle: textStyle atributi matn ko'rinishini matn uslubini o'rnatish uchun ishlatiladi. Matnning mumkin bo'lgan uslublari qalin, kursiv va normaldir. Agar biz matnli taqdimot uchun ikki yoki undan ortiq uslublardan foydalanishimiz kerak bo'lsa, unda buning uchun "|" operatoridan foydalaniladi.

background: fon atributi matn ko'rinishini fonini o'rnatish uchun ishlatiladi. Matn ko'rinishining foniga rang yoki naqsh o'rnatishimiz mumkin.

Edit text—elektron pochta xabarnomasida foydalanish uchun matnli shaklda taqdim etilgan ma'lumotlar formati.

Masalan. Bunda Ism, Familyangizni, Sharifingizni kiritish imkoniyati mavjud:

Password. Bu [autentifikatsiya](#) jarayonida foydalanuvchi identifikatorini tekshirish uchun ishlatiladigan belgilar qatori. Parollar odatda foydalanuvchi nomi bilan kon'yunkturada ishlatiladi; ushbu kod foydalanuvchiga qurilmaga, dasturga yoki veb-saytga kirishga ruxsat berish yaratiladi va ular faqat foydalanuvchiga ma'lum bo'ladi. Parol uzunligi bo'yicha har xil bo'lishi mumkin va harflar, raqamlar va maxsus [belgilarni](#) o'z ichiga olishi mumkin. Parol foydalanuvchi [shaxsiy identifikatsiya raqami](#).

E-mail. Elektron pochta (e-mail) — ma'lumotlarni uzatish tarmog'i orqali axborotlarni bir foydalanuvchi elektron qutisidan boshqasining jo'natish, qabul qilish va ma'lum vaqtgacha saqlanishini ta'minlovchi dasturiy texnik vositalar to'plami. E-Mail axborotlarni tarmoqning bir punktidan boshqasiga tezkor uzatishni ta'minlaydi. Aniqroq aytganda, elektron pochta - bu ma'lum bir shaxsga yoki shaxslar guruhiga tarmoq orqali yuborilgan [matnlar](#), [fayllar](#), [rasmlar](#) yoki boshqa [qo'shimchalarni](#) o'z ichiga olgan xabar.

Number signed. Bunda faqat 0 dan 9 gacha raqamlarni kiritish mumkin.

Matn yozish imkoniyati mavjud emas.

Matematikada har qanday bazadagi manfiy sonlar ularni minus ("-") belgisi bilan old qo'shimchalar bilan ifodalanadi.

Biroq, kompyuter texnikasida raqamlar qo'shimcha belgilersiz, faqat bitlar ketma-ketligi sifatida ifodalanadi.

AutoCompleteTextView .

Android-da AutoCompleteTextView - bu EditText-ga o'xshash ko'rinish, faqat foydalanuvchi matn kiritishi bilan avtomatik ravishda tugatish takliflari ro'yxatini ko'rsatadi. Takliflar ro'yxati foydalanuvchi aslida Editbox tarkibini shu bilan almashtiradigan elementni tanlashi mumkin bo'lgan ochiladigan menyuda ko'rsatiladi.



CheckedTextView

Android-da CheckedTextView tekshiriladigan interfeysni ushlab turadigan va uni ko'rsatadigan muntazam TextView kengaytmasi. Unda bir nechta matn bilan birga tasdiqlash katakchasi mavjud. Bu asosan ListView-da ishlatiladi, bu erda biz qaysi element tanlangan yoki tanlanmaganligini ko'rsatishni xohlaymiz. checkmark atributu grafik yoki chizilgan CheckedTextView ob'ektini ta'minlash uchun ishlatiladi.

Nazorat savollari

1. Flutter: Android Studio interfeysi haqida ma'lumot bering?
2. Android Studio da ilk dasturni yozish va uni emulyatorda ishlatish qay tartibda amalga oshiriladi?
3. Android Studio komponentalari haqida ma'lumot bering?

2.3.Flutter: Android Studioda asosiy komponentalar bilan ishlash: Layout, Table, ListView, Grid, List va boshqalar

Reja:

1. Flutter: Android Studioda asosiy komponentalar bilan ishlash

2. Layout, Table, ListView, Grid, List lar haqida umumiy ma'lumot

Palette panelida foydalanuvchi interfeysi komponentlari mavjud.

LinearLayout

Layout da ikkita variant **mavjud: Horizontal** va **Vertical**.

LinearLayout tartibi barcha ob'ektlar bir yo'nalishda — vertikal yoki gorizontal ravishda moslashtiradi.

Yo'nalish **Android yo'nalishi xususiyati bilan belgilanadi: orientation:**

```
android:orientation="horizontal"
```

```
android:orientation="vertical"
```

Misol:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    android:orientation="vertical">
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    android:layout_weight="1"
```

```
    android:orientation="horizontal">
```

```
<TextView
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="fill_parent"
```

```
    android:layout_weight="1"
```

```
    android:background="#aa0000"
```

```
    android:gravity="center_horizontal"
```

```
    android:text="red"/>
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:background="#00aa00"
    android:gravity="center_horizontal"
    android:text="green"/>
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:background="#0000aa"
    android:gravity="center_horizontal"
    android:text="blue"/>
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:background="#aaaa00"
    android:gravity="center_horizontal"
    android:text="yellow"/>
```

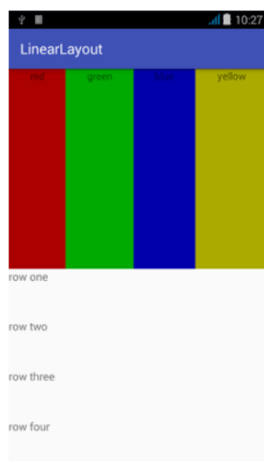
```
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="fill_parent"
    android:layout_weight="1"
    android:orientation="vertical">
    <TextView
```



```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="row one"
        android:textSize="15dp"/>
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="row two"
    android:textSize="15dp"/>
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="row three"
    android:textSize="15dp"/>
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="row four"
    android:textSize="15dp"/>
</LinearLayout>
</LinearLayout>
```

Natija:



Ushbu misolda TextView elementlarini gorizontal va vertikal ravishda joylashtirishning ikkita usuli qo'llaniladi.

LinearLayout markirovkasi android-ning qiziqarli xususiyatiga ega: layout_weight, bu ichki elementning egallaydigan joyini bildiradi. Ushbu xususiyat taqdimotning "ahamiyati" ni belgilaydi va bu elementni qolgan bo'sh joyni to'ldirish uchun kengaytirishga imkon beradi. Standart holda 0 ga teng bo'ladi.

Misol uchun, agar uchta matn maydoni mavjud bo'lsa va ularning ikkitasi 1 qiymatida bo'lsa, ikkinchisiga hech qanday qiymat (0) berilmagan bo'lsa, u holda uchinchi matn maydoni kengaytirilmaydi va bu maydon tomonidan ko'rsatilgan matn hajmi bilan belgilanadigan maydonni egallaydi. Boshqa ikkitasi uchinchi maydon bilan band bo'lmagan bo'sh joyni to'ldirish uchun bir xil darajada kengayadi. Agar uchinchi maydon 2 (0 o'rniga) hajmda belgilansa, bu maydon boshqa ikki kishiga qaraganda "muhimroq" deb e'lon qilinadi, shuning uchun uchinchi maydon umumiy maydonning 50 foizini oladi, dastlabki ikkitasi umumiy maydonning 25 foizini oladi

Bundan tashqari, Android xususiyatini ham belgilashingiz mumkin: weightSum. Agar atribut 100 qiymatini belgilasa, unda biz ichki matnlarning oladigan joy hajmini foiz sifatida qulay shaklda ko'rsatishimiz mumkin. Ushbu usul veb-ustalar tomonidan keng qo'llaniladi.

Misol:

```

    <?xml          version="1.0"          encoding="UTF-8"?><LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#0000FF"
    android:orientation="vertical"
    android:weightSum="100">
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="0dp"
        android:layout_weight="50"
        android:background="#FFFF00"
        android:gravity="center">

        <TextView
            android:id="@+id/textView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Salom"
            android:textColor="@color/black"
            android:textSize="45sp"
            duno"
        />
    </LinearLayout></LinearLayout>

```

Natija:



TabLayout/TabItem

TableLayout, masalan, Sudoku, Tic-toe va boshqalar kabi mantiqiy o'yinlar yaratishda foydalanish uchun qulay.

TableLayout uchun ba'zi qoidalar mavjud. Birinchidan, har bir ustunning kengligi ustundagi eng keng tarkib bilan belgilanadi. Ichki elementlari atributlarda **match_parent** qiymatini ishlatadi. Layout_height uchun **TableRow xususiyati** har doim *wrap_content*.

Agar **Android xususiyati**: "*" qiymatini belgilash uchun TableLayout komponentining stretchColumns bo'lsa, har bir TableRow komponentining tarkibi tartibning to'liq kengligiga cho'zilishi mumkin.

Misol

Ob-havo prognozi jadvalini yaratish

Misol uchun, ob-havo prognozi jadvalini yaratish uchun quyidagi dizayndan foydalanish mumkin:

1. Birinchi qatorda jadval sarlavhasini ko'rsatish
2. Ikkinchi qatorda taqvim turiga ko'ra sanalar ko'rsatiladi
3. Uchinchi qatorda eng yuqori haroratni ko'rsating
4. To'rtinchi qatorda eng past haroratni ko'rsating
5. Beshinchi qatorda grafiklarni ko'rsatish (yomg'ir, qor, bulut, quyosh)

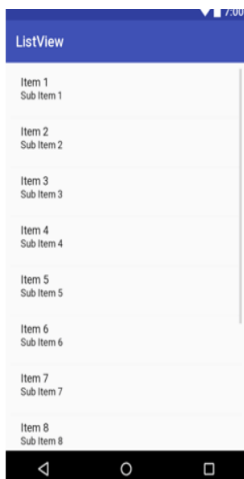


ListView

ListView-elementlarning aylanadigan ro'yxati

Biroq, agar XML kodini ko'rib chiqsak, u erda hech narsa ko'rilmaydi.

```
<ListView android:id="@+id/listView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
</ListView>
```



Mainactivity oynasiga o'tib quyidagilarni yozamiz

// kod onCreate usulida yozilgan()// ListView elementining namunasini
oling

```
ListView listView = findViewById(R.id.listView);
```

```
// string qator aniqlashfinal
```

```
String[] cat Names = new String[]
```

```
{ "Ryjik", "Barsik", "Murzik", "Murka", "Vaska","Thomasina",  
"Kristina"," Qurol", "Smoke", "Kuzya","Kiti", "Masyanya", "Simba"};
```

```
// ma'lumot adapteridan foydalaning  
ArrayAdapter<String> adapter = new  
ArrayAdapter<>(this,android.R.layout.simple_list_item_1,  
catNames);listView.setAdapter(adapter);
```

GridLayout

Android 4.0-da GridLayout (**Asboblar panelidagi Layouts bo'limi**) **nomi ostida yangi tartib paydo** bo'ldi. Bir qarashda TableLayout kabi **ko'rinishi mumkin**. Lekin, aslida, bu juda ham qulay va funktsionaldir. Va yangi platforma uchun ishlab chiqilgan yangi loyihalarda o'rganish va undan foydalanish tavsiya etiladi.

Android kutubxonasiga **Android klassi qo'shildi**. **support.v7.widget.GridLayout**, bu komponentni eski qurilmalar uchun ishlatishga imkon beradi.

Belgilar Android sinfiga tegishli. **widget.GridLayout** va TableLayout kabi ustunlar, qatorlar, hujayralar **mavjud**, ammo elementlar moslashuvchan tarzda sozlanishi mumkin.

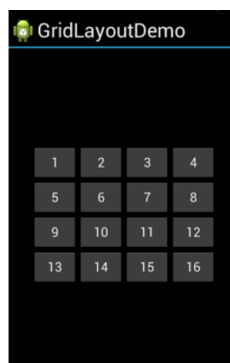
Ustunlar va qatorlar soni **Android atributlari tomonidan ishlatiladi**: **columnCount="number"** va **android: rowCount="number"**.

Misol:

```
<?xml version="1.0" encoding="utf-8"?>  
<GridLayout  
xmlns:android="http://schemas.android.com/apk/res/android"  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:layout_gravity="center"  
  android:columnCount="4"  
  android:orientation="horizontal" >  
  <Button android:text="1" />  
  <Button android:text="2" />  
  <Button android:text="3" />
```

```
<Button android:text="4" />
<Button android:text="5" />
<Button android:text="6" />
<Button android:text="7" />
<Button android:text="8" />
<Button android:text="9" />
<Button android:text="10" />
<Button android:text="11" />
<Button android:text="12" />
<Button android:text="13" />
<Button android:text="14" />
<Button android:text="15" />
<Button android:text="16" />
```

Natija:



GridView

GridView komponenti tekis jadval hisoblanadi. GridView uchun ArrayAdapter yoki BaseAdapter sinfidan olingan sinfni yaratish va uning getView usulini () qayta tuzish orqali ma'lumotlar elementlarini ko'rsatish uchun o'z maydonlarini ishlatish mumkin.

GridView uchun ustunlar soni tez-tez statik tarzda o'rnatiladi. Elementdagi qatorlar soni adapter taqdim etadigan elementlarning soniga qarab dinamik ravishda aniqlanadi.

Xususiyatlari

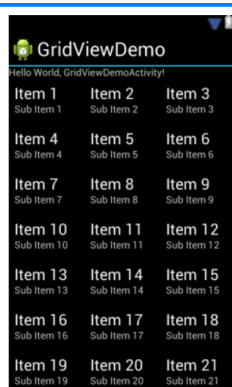
android — `numColumns`-ustunlar sonini aniqlaydi. Agar qiymat *berilgan* bo'lsa `auto_fit`, tizim mavjud maydonga asoslangan ustunlar sonini hisoblab chiqadi

android — `verticalSpacing`-jadval hujayralari orasidagi bo'sh joy hajmini belgilaydi

android:columnWidth-ustunlar kengligini o'rnatadi

android — `stretchMode`-bo'sh joy qoldig'ini belgilangan Android qiymati bilan jadval uchun ajratilganligini ko'rsatadi: `numColumns="auto_fit"`. Ustunlar xujayralari orasidagi bo'sh joy qoldig'ini kengaytirish uchun yoki `spacingWidth` — hujayralar orasidagi bo'shliqni kengaytirish uchun `columnWidth` qiymatlarini qabul qiladi

Misol:



Nazorat savollari

1. Flutter: Android Studioda asosiy komponentalari haqida ma'lumot bering?
2. Layout xususiyatlari haqida ma'lumot bering?
3. Table va ListView xususiyatlari haqida ma'lumot bering?
4. Grid va List lar haqida umumiy ma'lumot bering?

2.4. Flutter: Android Studioda oddiy kalkulyator dasturini yaratish, uni emulyatorida sinovdan o'tkazish va mobil telefonga o'rnatib ishlatish

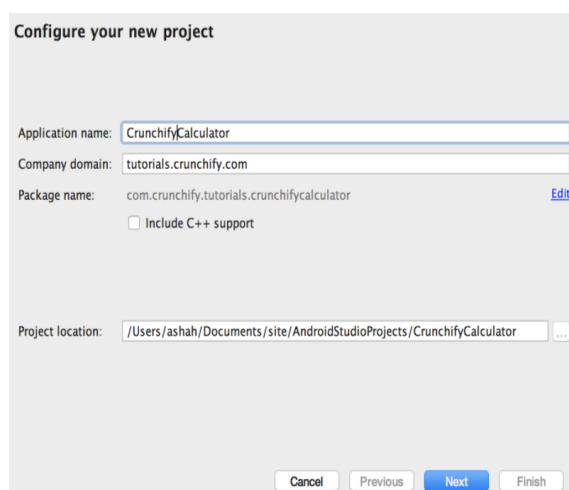
Reja:

1. Android Studioda oddiy kalkulyator dasturini yaratish
2. Kalkulyatorni emulyatoridan sinovdan o'tkazish va mobil telefonga o'rnatib ishlatish

Android studioda kalkulyator dasturini yaratish quyidagi bosqichlarda amalga oshiriladi

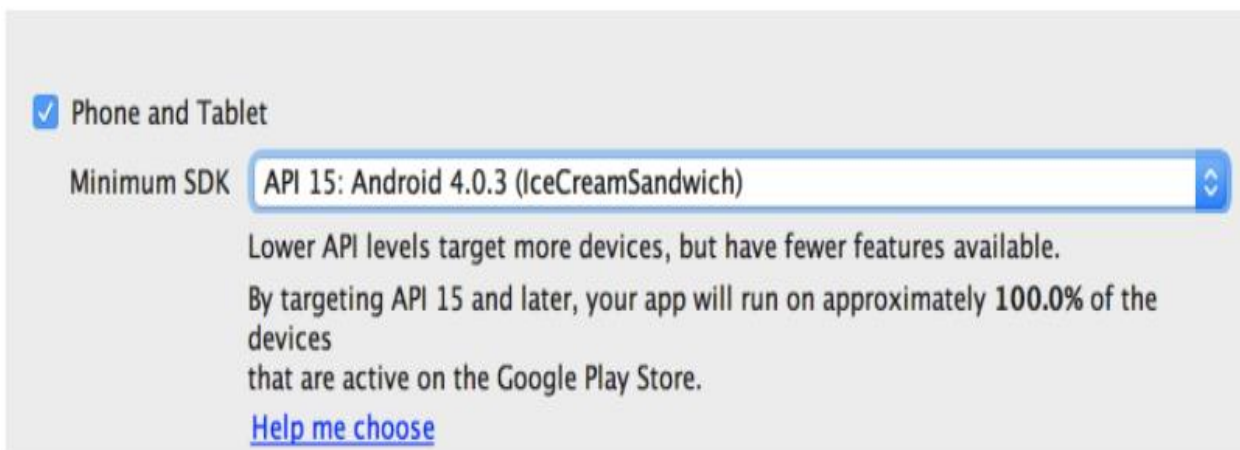
1- bosqichi. Android Studioni oching. Yangi Android Studio loyahasini ishga tushirishni bosing.

Loyihaga nom bering. Kalkulator qolgan joylari o'z holicha qoladi.



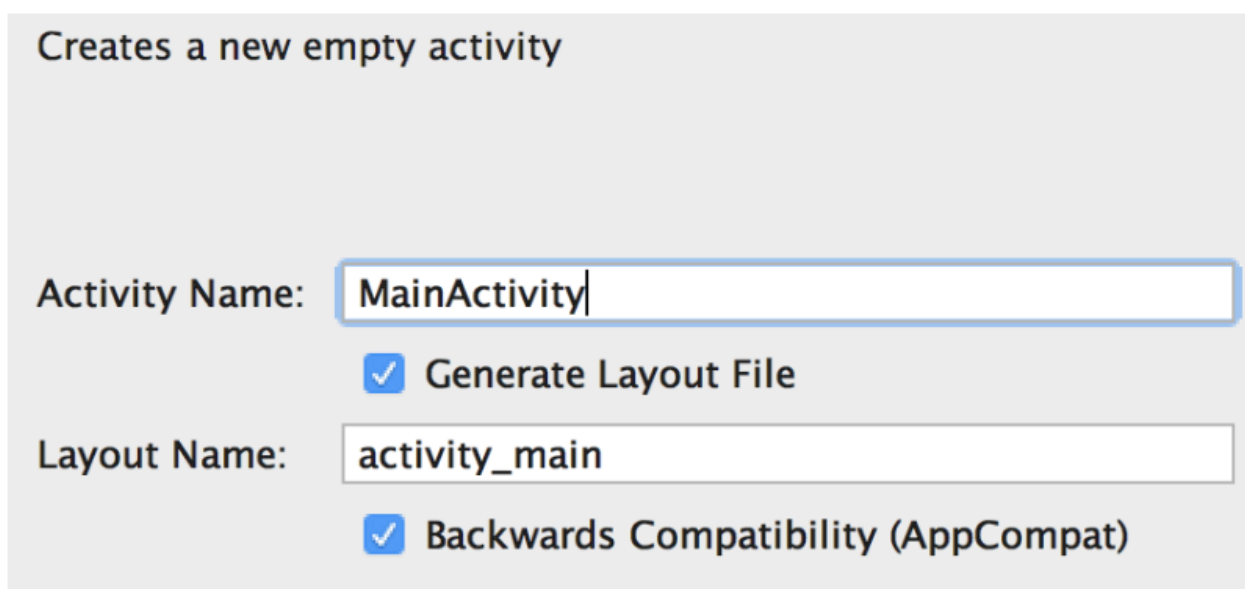
2-bosqichi

Minimal SDK API 15: Android 4.0.3(IceCreamSandwich)ni tanlaymiz , masalan, API 15 (IceCreamSandwich), chunki u qurilmalarning deyarli 94% ni qamrab oladi va deyarli barcha xususiyatlarga ega. Agar 100% qurilmani qamrab olishni istasak, API 8 ni tanlashimiz mumkin: Android 2.2 (Froyo).



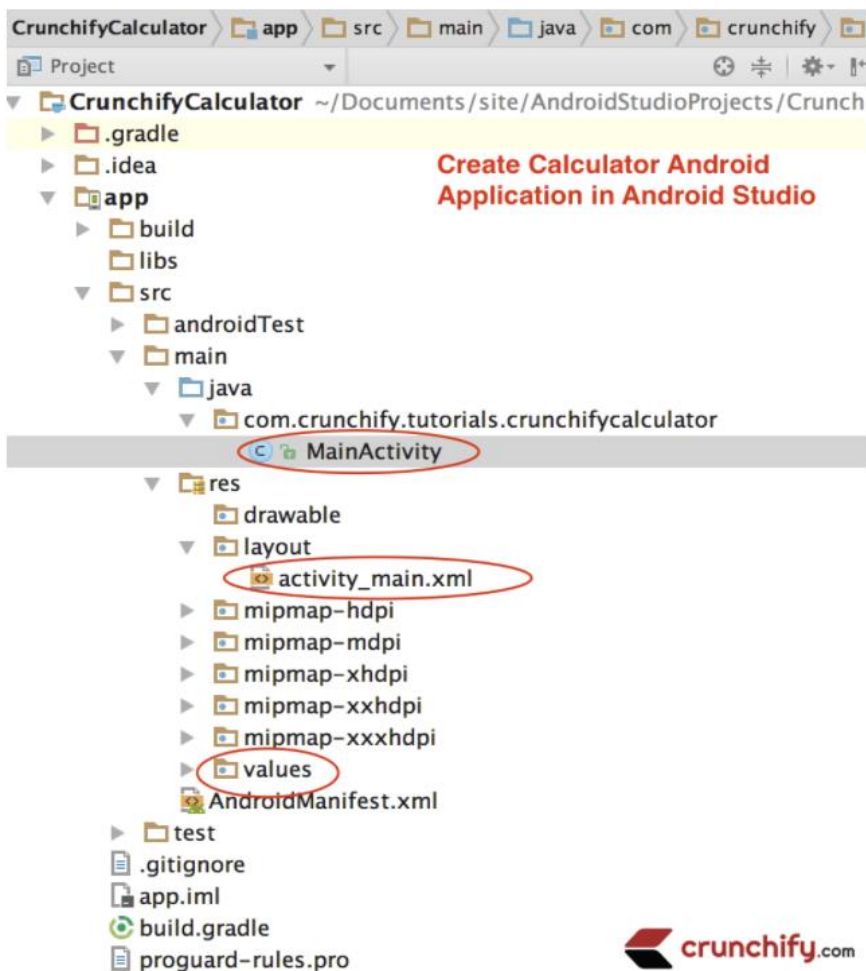
3-qadam

Empty Activity ni tanlaymiz va next tugmasini bosamiz. Qolgan joylarining o'zi qoladi.



4 bosqichi

Loyihamizni yaratish uchun "tayyor" tugmasini bosgandan so'ng, loyihamiz oynasi ochiladi.



Kalkulyator ilovasida to'rtta asosiy element mavjud:

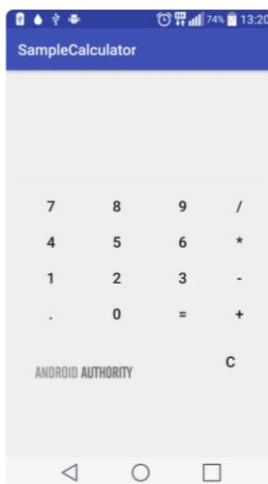
RelativeLayout - ekranda boshqa elementlar qanday stacked yoki ko'rsatilishini aniqlaydi. RelativeLayout ichidagi elementlarni bir-biriga yoki o'zlariga nisbatan joylashtirish uchun ishlatiladi.

TextView - element matnni ko'rsatish uchun ishlatiladi. Yozilgan ma'lumotni tahrirlab bo'lmaydi. Hisoblash natijalari TextView yordamida ko'rsatiladi.

EditText - TextView boshqaruviga o'xshash yagona farq shundaki, foydalanuvchilar matnni tahrirlashlari mumkin. Ammo kalkulyator faqat kiritilgan ma'lumotlarning belgilangan to'plamiga ruxsat berganligi sababli, biz uning holatini "tahrirlab bo'lmaydi" deb belgilaymiz. Foydalanuvchi raqamlarni bosganda, biz ularni EditTextga chiqaramiz.

Tugma - foydalanuvchi buyruqlariga javob beradi. Oddiy Android ilovasini yaratishda biz kalkulyatorida raqamlar va harakat operatorlari tugmalaridan foydalanamiz.

Kalkulyator ko'rinishi



Kalkulyatorning ichki komponentlari

Android telefonida ilovani yaratishdan oldin, "value One" va "valueTwo" da ishlatiladigan raqamlar mavjudligini ta'kidlaymiz. Har ikki o'zgaruvchilar uchun ham double turi mavjuddir. Biz Valueone uchun maxsus NaN qiymatini o'rnatamiz. Quidagicha yoziladi:

```
private double valueOne = Double.NaN;
```

```
private double valueTwo;
```

Ushbu oddiy kalkulyator faqat qo'shish, ayirish, ko'paytirish va bo'lish operatsiyalarini bajarishi mumkin. Shuning uchun biz ushbu operatsiyalarni taqdim etish uchun to'rtta statik belgini va quyidagi operatsiyani o'z ichiga olgan CURRENT_ACTION o'zgaruvchisini aniqlaymiz.

```
private static final char ADDITION = '+';
```

```
private static final char SUBTRACTION = '-'; private static final char  
MULTIPLICATION = '*';
```

```
private static final char DIVISION = '/';
```

```
private char CURRENT_ACTION;
```

Keyin natijani formatlash uchun DecimalFormat sinfidan foydalanamiz.

```
decimalFormat = new DecimalFormat("#.#####");
```

Raqamlarni bosish jarayoni

Android uchun yaratilgan oddiy ilovamizda foydalanuvchi raqamni yoki nuqtani bosganda, biz ushbu raqamni editText-ga qo'shishimiz kerak.

Quyidagi kod misoli, bu raqam uchun qanday amalga oshirilganligini ko'rsatadi.

```
binding.buttonZero.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view)
    { binding.editText.setText(binding.editText.getText() + "0"); } });
```

Kalkulyator yaratish

Android Studio dasturini ishga tushiramiz, yangi loyiha yaratamiz. Komponentlarning nomlarini o'z holicha qoldiraman: **faoliyat nomi** - asosiy, fayl tartibi-**activity_main**.

Ekranda 2 ta EditText va 4 ta Button tugmasidan foydalanamiz.

Misol:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
<EditText
android:layout_width="match_parent"
android:layout_height="wrap_content"
```

```
android:id="@+id/et2"
android:hint="@string/num"/>
<EditText
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/et1"
android:hint="@string/num"/>
<TextView
android:layout_width="fill_parent"
android:layout_height="20pt"
android:id="@+id/result"
android:text="Results"/>
<Button
android:layout_width="30pt"
android:layout_height="30pt"
android:id="@+id/plus"
android:text="+"
android:textSize="20pt"/>
<Button
android:layout_width="30pt"
android:layout_height="30pt"
android:id="@+id/minus"
android:text="-"
android:textSize="20pt"/>
<Button
android:layout_width="30pt"
android:layout_height="30pt"
android:id="@+id/multiply"
```

```
android:text="*"
android:textSize="20pt"/>
<Button
android:layout_width="30pt"
android:layout_height="30pt"
android:id="@+id/divide"
android:text="/"
android:textSize="20pt"/>
</RelativeLayout>
```

```
<resources>
<string name="app_name">My Application</string>
<string name="hello_world">Hello world!</string>>
<string name="action_settings">>chiqish</string>>
<string name="calculator">>Kalkulyator</string>

<string name="num">>kirish...</string>
</resources>
```

Barcha elementlarni yaratganingizdan so'ng, dizayn rejimiga o'tamiz va yaratilgan elementlarni biz xohlaganicha harakatlantirishimiz mumkin.



Biz uchun zarur bo'lgan barcha elementlarni Yaratgandan so'ng, dasturni **MainActivity.java**. ga yozamiz.

Bizga kerak:

- 1) ishlatilgan narsalarni e'lon qilish,
- 2) ushbu moslamalarni biz yaratgan elementlarga ulash,
- 3) berilgan tugmachalar bosilganda ko'rsatish,
- 4) tugmalarining har birini bosish jarayonida nima bo'lishini tushuntiring.

Dastur kodi:

```
package home.myapplication;  
import android.app.Activity;  
import android.app.AlertDialog;  
import android.support.v7.app.ActionBarActivity;  
import android.os.Bundle;  
import android.view.Menu;
```



```

import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends Activity implements View.OnClickListener {
    // Quyidagi ob'ektlarning mavjudligini e'lon qiling:
    private EditText mE1; private EditText mE2; private TextView mResult;
private Button mB1; private Button mB2; private Button mB3; private
Button mB4; private String mCount = "";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState); setContentView(R.layout.activity_main); //
        Kiritilgan dasturiy moslamalarni formatlash faylida yaratilgan elementlarga
        ulang:
        mE1=(EditText)findViewById(R.id.et1);
        mE2=(EditText)findViewById(R.id.et2);
        mResult=(TextView)findViewById(R.id.result);
        mB1=(Button)findViewById(R.id.plus);
        mB2=(Button)findViewById(R.id.minus);
        mB3=(Button)findViewById(R.id.multiply);
        mB4=(Button)findViewById(R.id.divide); // Bizning tugmachalarni bosish:
        mB1.setOnClickListener(this); mB2.setOnClickListener(this);
        mB3.setOnClickListener(this); mB4.setOnClickListener(this); } @Override
        public void onClick(View v) {
            // Biz faoliyat ko'rsatadigan o'zgaruvchilarni kiriting:
            float value1 =0;
            float value2=0;

```

```
float result=0;
// Kirish elementlari bilan value1 va value2 uchun qadriyatlarni qabul:
value1=Float.parseFloat(mE1.getText().toString());
value2=Float.parseFloat(mE2.getText().toString()); // Tugmalar harakatlarini
tavsiflang:
switch (v.getId()) { case R.id.plus:
mCount = "+";
result = value1 + value2; break;
case R.id.minus:
mCount = "-";
result = value1 - value2; break;
case R.id.multiply:
mCount = "*";
result = value1 * value2; break;
case R.id.divide:
mCount = "/";
result = value1 / value2; break;
default:
break;
}
// natijani namoyish qilishni sozlash:
mResult.setText(value1 + " " + mCount + " " + value2 + " = " + result); } }
```

Nazorat savollari

1. Android Studioda oddiy kalkulyator dasturini yaratish ketma-ketligini tushuntiring?
2. Kalkulyatorni emulyatordan sinovdan o'tkazish qay tartibda amalga oshiriladi?
3. Yaratilgan kalkulyatorni mobil telefonga o'rnatib ishlatish jarayoni aytib bering?

III BOB. FLUTTER: ANDROID STUDIODA MA'LUMOTLAR BAZASI BILAN ISHLASH

SQLite paydo bo'lishi bu mahalliy ilovalarga ma'lumotlar bazalarini joylashtirish uchun eng mashhur echimlardan biri bo'ldi. Demo loyihasida asosiy interfeysdan elementlarni yaratish, yangilash va o'chirish uchun juda oddiy vazifa boshqaruv dasturini yaratishda foydalaniladi.

3.1. Flutter: Android Studioda ma'lumotlar bazasi bilan ishlash, SQLite, MySQL MBBT lar bilan ulanish

Reja:

1. Flutter: Android Studioda ma'lumotlar bazasi bilan ishlash
2. SQLite, MySQL MBBT lar bilan ulanish

SQLite ma'lumotlar bazasiga ulanish

Android ma'lumotlar bazasini boshqarishning eng keng tarqalgan tizimlaridan biri **SQLite**-ni qo'llab-quvvatlaydi. **android.database.sqlite** paketi **SQLite** ma'lumotlar bazalari bilan ishlashga imkon beradigan sinflar to'plamini aniqlaydi. Va har bir dastur o'z ma'lumot bazasini yaratishi mumkin.

Androidda **SQLite**-dan foydalanish uchun SQL tili yordamida ma'lumotlar bazasini yaratish kerak. Shundan so'ng ma'lumotlar bazasi yo'l bo'ylab dastur katalogida saqlanadi:

DATA/data/[Ilova_nomi]/databases/[ma'lumotlar_bazasi_fayli_nomi].

Standart ravishda, Android OS allaqachon standart dasturlar tomonidan ishlatiladigan bir nechta o'rnatilgan SQLite to'plamlarini o'z ichiga oladi - kontaktlar ro'yxati, kameradan fotosuratlarini saqlash, musiqiy albomlar va hk.

Ma'lumotlar bazalari bilan ishlashning asosiy funksiyalari **android.database** paketi tomonidan ta'minlanadi. To'g'ridan-to'g'ri **SQLite** bilan ishlash uchun **android.database.sqlite** paketi mavjud.

SQLitedagi ma'lumotlar bazasi **android.database.sqlite.SQLiteDatabase** klassi yordamida ifodalanadi. Bu ma'lumotlar bazasiga so'rovlarni bajarishga, u bilan turli xil manipulyatsiyalarni bajarishga imkon beradi.

android.database.sqlite.SQLiteCursor sinfi so'rovni taqdim etadi va shu so'rovga mos qatorlar to'plamini qaytarishga imkon beradi.

android.database.sqlite.SQLiteQueryBuilder klassi SQL so'rovlarini yaratishga imkon beradi.

android.database.sqlite.SQLiteOpenHelper klassi, agar ular mavjud bo'lmasa, barcha jadvallar bilan ma'lumotlar bazasini yaratishga imkon beradi.

SQLite quyidagi ma'lumotlar turi tizimidan foydalanadi:

- **INTEGER**: javadagi int turiga o'xshash butun sonni ifodalaydi.
- **REAL**: javada float va double -ga o'xshash raqamni ifodalaydi.
- **TEXT**: javada String va char -ga o'xshash belgilar majmuini ifodalaydi.
- **BLOB**: binar ma'lumotlarning massivini ifodalaydi.

Ma'lumotlar bazasini yaratish va ochish.

Androidda Activity kodida yangi ma'lumotlar bazasini yaratish yoki ochish uchun **openOrCreateDatabase()** metodini chaqirishimiz mumkin. Ushbu metod uchta parametrni olishi mumkin:

- ma'lumotlar bazasi nomi.

- ishlash rejimini aniqlaydigan raqamli qiymat (odatda **MODE_PRIVATE** doimiysi shaklida).
- ma'lumotlar bazasi bilan ishlash uchun kursor yaratish factorysi-ni belgilaydigan **SQLiteDatabase.CursorFactory** ob'ekti ko'rinishidagi ixtiyoriy parametr.

Masalan, **app.db** ma'lumotlar bazasini yaratish:

```
SQLiteDatabase db = getBaseContext().openOrCreateDatabase("app.db",
MODE_PRIVATE, null);
```

Ma'lumotlar bazasi so'rovini bajarish uchun **SQLiteDatabase** sinfining *execSQL* usulidan foydalanishingiz mumkin.

Masalan, ma'lumotlar bazasida **users** jadvalini yaratish:

```
SQLiteDatabase db = getBaseContext().openOrCreateDatabase("app.db",
MODE_PRIVATE, null);
```

```
db.execSQL("CREATE TABLE IF NOT EXISTS users (name TEXT, age
INTEGER)");
```

Ma'lumotlar bazasidan ba'zi ma'lumotlarni olish kerak bo'lsa, unda `rawQuery()` metodi qo'llaniladi.

Masalan, ma'lumotlar bazasidan barcha ob'ektlarni olish:

```
SQLiteDatabase db = getBaseContext().openOrCreateDatabase("app.db",
MODE_PRIVATE, null);
```

```
db.execSQL("CREATE TABLE IF NOT EXISTS users (ismi TEXT, yoshi
INTEGER)");
```

```
Cursor query = db.rawQuery("SELECT * FROM users;", null);
```

```
if(query.moveToFirst()){
    String ismi = query.getString(0);
    int yoshi = query.getInt(1);
}
```

db.rawQuery() metodi qabul qilingan ma'lumotlarni olishimiz mumkin bo'lgan Cursor ob'ektini qaytaradi.

Ma'lumotlar bazasida ob'ektlar bo'lmasligi mumkin va buning uchun *query.moveToFirst()* metodi yordamida ma'lumotlar bazasidan olingan birinchi ob'ektga o'tadi. Agar bu metod false qiymatini qaytarsa, demak so'rov ma'lumotlar bazasidan hech qanday ma'lumot olmagan.

Endi ma'lumotlar bazasi bilan ishlash uchun oddiy ilova tuzamiz. Buning uchun Android Studioda yangi project yarating.

activity_main.xml faylini eng sodda grafik interfeys uchun quyidagicha o'zgartiring:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onClick"
        android:text="Click" />

    <TextView
        android:id="@+id/text_view"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="18sp" />
```

```
</LinearLayout>
```

[MainActivity.java](#) faylida biz ma'lumotlar bazasi bilan o'zaro aloqani bog'laymiz:

```
package com.example.android.sqlite;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import android.database.Cursor;
```

```
import android.database.sqlite.SQLiteDatabase;
```

```
import android.os.Bundle;
```

```
import android.view.View;
```

```
import android.widget.TextView;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
    }
```

```
    public void onClick(View view) {
```

```
        SQLiteDatabase db = getBaseContext().openOrCreateDatabase("app.db",  
MODE_PRIVATE, null);
```

```
        db.execSQL("CREATE TABLE IF NOT EXISTS users (ismi TEXT, yoshi  
INTEGER)");
```

```
        db.execSQL("INSERT INTO users VALUES ('Alisher', 24);");
```

```
        db.execSQL("INSERT INTO users VALUES ('Bahrom', 30);");
```

```

Cursor query = db.rawQuery("SELECT * FROM users;", null);
TextView textView = findViewById(R.id.text_view);
if (query.moveToFirst()) {
    do {
        String ismi = query.getString(0);
        int yoshi = query.getInt(1);
        textView.append("Ismi: " + ismi + "   Yoshi: " + yoshi + "\n");
    }
    while (query.moveToNext());
}
query.close();
db.close();
}
}

```

Bu ilovadagi tugmani bosish bilan avval *app.db* ma'lumotlar bazasida *users* yangi jadvali yaratiladi, so'ngra unga **INSERT** yordamida ma'lumotlar bazasiga ikkita ob'ekt qo'shiladi.

Keyinchalik, **SELECT** yordamida barcha qo'shilgan foydalanuvchilarni ma'lumotlar bazasidan Cursor kursori ko'rinishida olamiz.

query.moveToFirst()-ni chaqirib, biz kursorni birinchi ob'ektga o'tkazamiz va bir nechta ob'ektga ega bo'lishimiz mumkinligi sababli, **do ... while** siklida hamma kursor bosib o'tiladi.

Kursordan ma'lumot olish uchun **query.getString(0)** va **query.getInt(1)**

metodlaridan foydalaniladi. Ma'lumotlarni oladigan ustunning nomeri qavs ichida metodlarga o'tkaziladi. Masalan, yuqorida biz avval foydalanuvchi nomini String tipida, so'ngra yoshni raqam(int) sifatida qo'shdik. Bu shuni anglatadiki, nolinch ustun **getString()** metodi yordamida olingan String qiymati bo'ladi, va keyingi - birinchi ustun **getInt()** metodi qo'llaniladigan raqamli (int) qiymatdir.

Kursor va ma'lumotlar bazasi bilan ishlashni tugatgandan so'ng, biz barcha tegishli ob'ektlarni yopamiz:

```
query.close();
```

```
db.close();
```

Agar Cursorni yopmasak, xotirada muammo yuzaga kelishi mumkin.



Misol:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:text="@string/hello"
        android:textSize="18dp" android:paddingBottom="10dp"/>
    <TextView android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:text="Ismingiz:"
        android:textSize="15dp"/>
    <EditText android:id="@+id/ism" android:layout_width="157dp"
        android:layout_height="wrap_content" /?
    <TextView android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:text="Familyangiz:" />
    <EditText android:id="@+id/familya" android:layout_width="163dp"
        android:layout_height="wrap_content" />
    <Button android:id="@+id/add" android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:text="Add" />
```

```

<Button android:id="@+id/deleteall" android:layout_width="fill_parent"
android:layout_height="wrap_content" android:text="Delete All" />
<ListView android:id="@+id/contentlist" android:layout_width="fill_parent"
android:layout_height="fill_parent"/>
>

```

Quyidagi ko'rinishda layout hosil bo'ladi:



Yana bitta row.xml fayl yaratamiz, bu fayl bizga nima uchun kerak? Bu fayl bizga bazamizdagi ma'lumotlarni ko'rsatib berish uchun kerak.

row.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:layout_width="fill_parent"
android:layout_height="wrap_content">
<LinearLayout android:orientation="horizontal"
android:layout_width="fill_parent" android:layout_height="wrap_content">
<TextView android:layout_width="wrap_content"
android:layout_height="fill_parent" android:padding="2dip"
android:text="#" />

```

```
<TextView android:id="@+id/id" android:layout_width="wrap_content"
android:layout_height="fill_parent" android:padding="2dip"
android:paddingRight="10dip"/>
```

```
<TextView android:layout_width="wrap_content"
android:layout_height="fill_parent" android:padding="2dip"
android:paddingRight="10dip" android:text="-" />
```

```
<TextView android:id="@+id/text1" android:layout_width="fill_parent"
android:layout_height="fill_parent" android:padding="2dip"/>
```

```
</LinearLayout>
```

```
<TextView android:id="@+id/text2" android:layout_width="fill_parent"
android:layout_height="wrap_content" android:padding="2dip"/>
```

```
</LinearLayout>
```

Kerakli layout larni yaratib olingach, asosiy kod yoziladi. Activity faylimizni ochib, main.xml faylida ko'rsatgan editTextlar ,Buttonlar va listView larni activity faylimizga bog'laymiz.

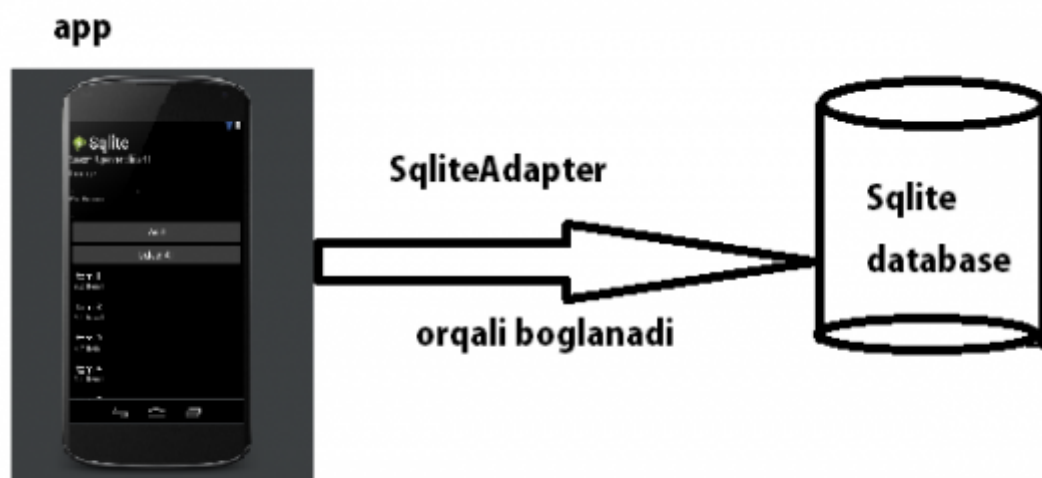
```
ism = (EditText)findViewById(R.id.ism);
familya= (EditText)findViewById(R.id.familya);
buttonAdd = (Button)findViewById(R.id.add);
buttonDeleteAll = (Button)findViewById(R.id.deleteall);
opennetlist = (ListView)findViewById(R.id.opennetlist);
```

Endi biz dasturimizni malumotlar bazasi bilan bog'laymiz. Biz SQLiteAdapter nomli klass yaratamiz, aynan shunday nom bo'lishi shart emas, o'ziz hohlagan nom berishiz mumkin.

Albatta Constructor ni yaratamiz.

```
public SQLiteAdapter(Context c)
```

```
{ context = c;
}
```



Ushbu klassimizni ichida yana bitta klass yaratamiz. Bu klassning nomini SQLiteHelper deb nomladik, va SQLiteOpenHelper klassidan extend olgan. SQLiteOpenHelper — bu bizga database yaratib, undagi amallar bajarishga yordam beruvchi klass.

Bu klassning 2 ta metodi mavjud:

onCreate — database yaratish uchun

onUpgrade — yaratilgan database ni yangilash uchun

onCreate metodini ichiga quydagilarni yoziladi:

```
@Override public void onCreate(SQLiteDatabase db)
{ // TODO Auto-generated method stub
db.execSQL(SCRIPT_CREATE_DATABASE); }
```

Endi quyidagi o'zgaruvchilarni yaratamiz, bu o'zgaruvchilar database bilan ishlash uchun har doim kerak bo'ladigan o'zgaruvchilar bo'lganligi uchun hususiyatini public qilamiz

```
public static final String MYDATABASE_NAME = "opennet";
public static final String MYDATABASE_TABLE = "opennetchilar";
public static final int MYDATABASE_VERSION = 1;
```

```

public static final String KEY_ID = "_id";
public static final String ism = "ism";
public static final String familya = "familya";
//create table MY_DATABASE (ID integer primary key, Content text not null);
private static final String SCRIPT_CREATE_DATABASE = "create table "
+ MYDATABASE_TABLE
+ " (" + KEY_ID + " integer primary key autoincrement, " + ism + " text not
null, " + familya + " text not null);";
private SQLiteHelper sqLiteHelper;
    private SQLiteDatabase sqLiteDatabase;
    private Context context;

```

Endi database yaratish funksiyasiga kelsak, quyidagikodni yozamiz

```

sqLiteHelper = new SQLiteHelper(context, MYDATABASE_NAME, null,
MYDATABASE_VERSION);

```

SQLiteHelper klassiga murojaat qiladi va onCreate metodi ishga tushadi, va bizning databaseimizni yaratib beradi.

Eslatma:database ni 1 marta yaratadi,agar shunday nomli database bor bo'lsa yaratmaydi.

SqliteAdapter klassimizni ichida quyidagi funksiyalarni yaratamiz:

```

openToRead() — nomidan ham ma'lumki, databaseni o'qishga ruxsat olish
uchun      public      SQLiteAdapter      openToRead()      throws
android.database.SQLException { sqLiteHelper = new SQLiteHelper(context,
MYDATABASE_NAME, null, MYDATABASE_VERSION); sqLiteDatabase
= sqLiteHelper.getReadableDatabase(); return this; }

```

openToWrite() — nomidan ham ma'lumki, databaseni tahrirlash uchun ruxsat olish

```

public SQLiteAdapter openToWrite() throws android.database.SQLException {
sqLiteHelper = new SQLiteHelper(context, MYDATABASE_NAME, null,
MYDATABASE_VERSION);

```

```
sqliteDatabase = sqliteHelper.getWritableDatabase();  
return this; }
```

Va nihoyat databaseni ishlatib bo'lganimizdan so'ng, uni yopish.

```
public void close(){ sqliteHelper.close();  
}
```

Database ga ma'lumot qo'shish funksiyasi

```
public long insert(String content1, String content2){ ContentValues  
contentValues = new ContentValues(); contentValues.put(ism, content1);  
contentValues.put(familya, content2);  
return  
sqliteDatabase.insert(MYDATABASE_TABLE, null, contentValues); }
```

Database dagi barcha ma'lumotlarni o'chirish funksiyasi

```
public int deleteAll(){return sqliteDatabase.delete(MYDATABASE_TABLE,  
null, null); }
```

Databasidagi barcha ma'lumotlarni olish uchun ishlatiladigan funksiya

```
public Cursor queueAll(){String[] columns = new String[]{KEY_ID, ism,  
familya};  
Cursor cursor = sqliteDatabase.query(MYDATABASE_TABLE, columns,  
null, null, null, null, null);  
return cursor; }
```

Endi SQLiteAdapter imiz dan yangi object yaratamiz va database ga yozish uchun ruxsat beruvchi funksiyani ishga tushiramiz:

```
mySQLiteAdapter = new SQLiteAdapter(this);  
mySQLiteAdapter.openToWrite();
```

Database dagi barcha ma'lumotlarni oluvchi funksiyani ishga tushiramiz:

```
cursor = mySQLiteAdapter.queueAll(); SimpleCursorAdapter — bu Adapter  
bizga databasedagi malumotlarni ro'yhatda ko'rsatishga yordam beradi.  
Databasedan barcha ma'lumotlarni olganimizdan keyin uni ListView ga  
joylaymiz:
```

```
String[] from = new String[]{SQLiteAdapter.KEY_ID, SQLiteAdapter.ism,
SQLiteAdapter.familiya}; int[] to = new int[]{R.id.id, R.id.text1, R.id.text2};
cursorAdapter = new SimpleCursorAdapter(this, R.layout.row, cursor, from,
to); opennetchilar.setAdapter(cursorAdapter);
```

Endi Button tugmalari bilan ishlash qoldi yani yangi ma'lumot qo'shish tugmasi va barcha ma'lumotlarni o'chirish tugmasi:

```
Button.OnClickListener buttonAddOnClickListener = new
Button.OnClickListener(){ @Override public void onClick(View arg0)
{ // TODO Auto-generated method stub String data1 =
ism.getText().toString();
```

```
String data2 = familiya.getText().toString();
```

```
mySQLiteAdapter.insert(data1, data2);
```

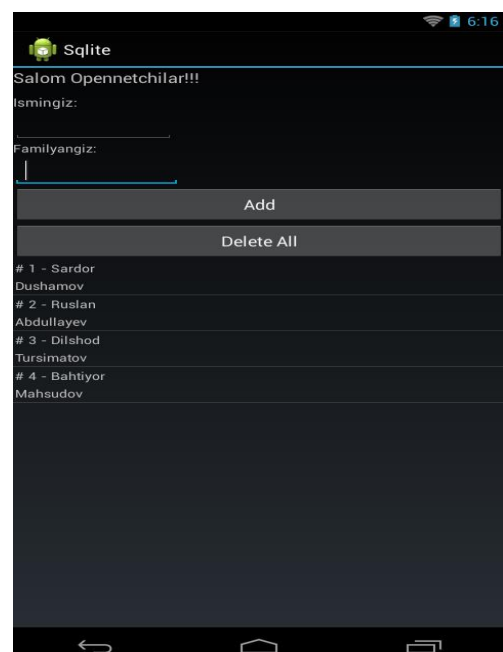
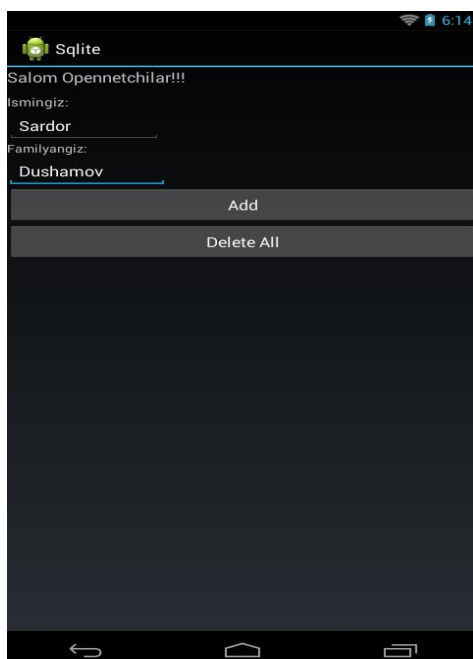
```
updateList(); }
```

```
};
```

```
Button.OnClickListener buttonDeleteAllOnClickListener = new
Button.OnClickListener(){ @Override public void onClick(View arg0)
{ // TODO Auto-generated method stub mySQLiteAdapter.deleteAll();
updateList(); }
```

```
};
```

Endi Dasturni ishga tushiramiz va ma'lumot kiritib Add tugmasini bosamiz:



Nazorat savollari

1. Flutter: Android Studioda ma'lumotlar bazasi bilan ishlash haqida umumiy ma'lumot bering?
2. SQLite, MySQL MBBT lar bilan ulanish qayt taribda amalga oshiriladi?

3.2. Flutter: Android Studioda Telegram bot bilan ishlovchi mobil ilova misolida API bilan ishlashni o'rganish

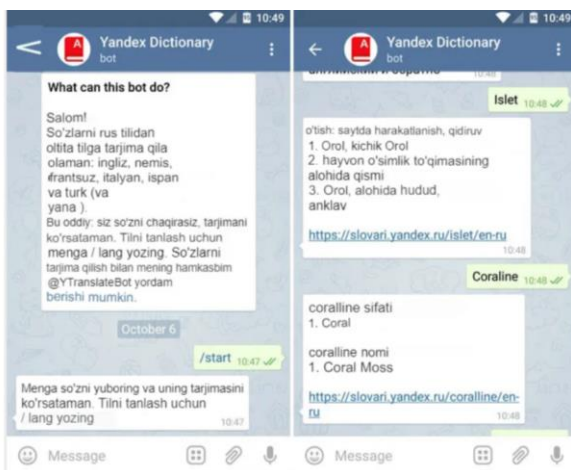
Reja:

1. Telegram bot bilan ishlovchi mobil ilovalar haqida umumiy ma'lumot
2. Shaxsiy telegram bot yaratish ketma-ketligi

Telegram botlarning eng foydali hususiyati, botlar bizning o'rnimizga foydalanuvchilar bilan bemalol suhbat qurishi mumkin. Natijada hizmat ko'rsatishi lozim bolgan inson, minglab foydalanuvchilar tomonidan har doim beriladigan bir xil savollarga qayta — qayta javob berishi shart emas, foydalanuvchilar ham o'z savollariga javob olish uchun navbat kutib vaqt o'tkazmaydi.

Telegram-bu boshqa foydalanuvchilar bilan muloqot qilish, ma'lumot almashish va bir-biriga fayllarni yuborishning oddiy usuli emas, balki juda ko'p foydali vazifalarni bajarishga imkon beruvchi ajoyib vosita.

Bundan tashqari, messenjer orqali yangiliklarni o'rganish, valyutalardan xabardor bo'lish, matnlarni tarjima qilish, o'yinlar o'ynash va boshqa ko'p narsalar qilish mumkin. Va bu harakatlar **maxsus robot – telegrammalar botlariga yordam beradi.**



"Bot" nomi qisqartirilgan "robot" so'zidan kelib chiqadi. Telegram-bot avtomatik rejimda yaratilgan maxsus hisob bo'lib, foydalanuvchilarga messenjerning o'zi orqali turli harakatlar qilish imkonini beradi.

Robotlar "boshlash" tugmasini bosgandan so'ng, "savol-javob" tamoyiliga ko'ra, foydalanuvchi matn buyruqlar bo'yicha harakatlarni amalga oshiradi. Misol uchun, bugungi kunda ob-havo haqida bilib olishingiz mumkin:



Telegram-botlarda ko'plab aniq afzalliklar mavjud:

- 24/7 mavjud;
- Foydalanuvchiga darhol javob;
- Foydalanish qulayligi, "savol-javob" tamoyili bo'yicha muloqot va matnli vazifalar hatto juda tajribasiz messenjer foydalanuvchisiga ham berilishi mumkin;
- Qo'shimcha dasturlar, ilovalar va boshqalarni o'rnatishni talab qilmaydi. bot bilan barcha aloqa to'g'ridan-to'g'ri messenjer orqali amalga oshiriladi;

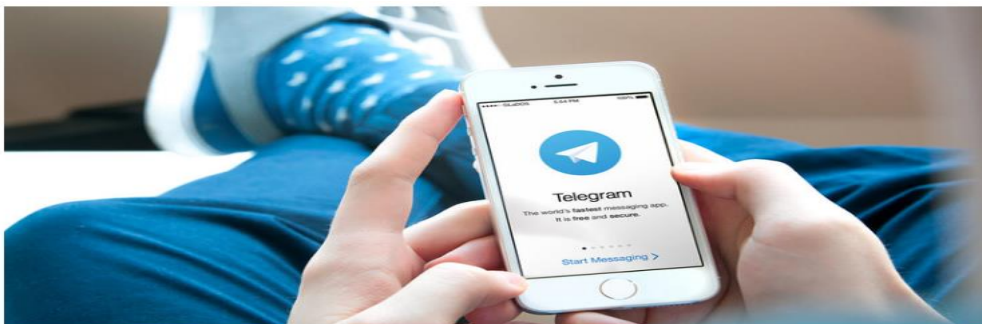
- Shaxsiy ma'lumotlar xavfsizligi-Bots faqat belgilangan buyruqlar bo'yicha ishlaydi.
- Cheksiz imkoniyatlar – virtual yordamchi yangiliklarni yuborish, anekdotlar haqida hikoya qilish, muhim ma'lumotlarni eslatish, muassasalarni qidirish, restoranda stollarni bron qilish, chiptalarni buyurtma qilish va h. k.

Telegram bot vazifalari

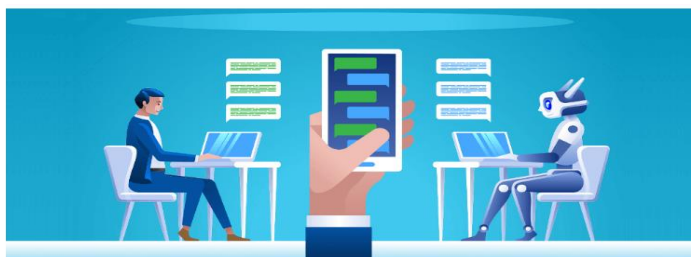
1. O'yin-kulgi. Bots kulgili memlar, rasmlar, latifalar yuborishi mumkin, filmni tanlashga yordam beradi, ovozli xabar orqali qo'shiqni topadi va boshqalar.
2. Fayllarni qidirish va almashish. Bot turli manbalardan fayllarni yuborish va saqlashga, torrentlarni, elektron kitoblarni va boshqalarni topishga yordam beradi.
3. Yangiliklar, muhim ma'lumotlar. Bot yangiliklar, ob-havo, valyuta kurslari va boshqalarni yoritadi.
4. Kommunal xizmatlar va asboblari. Robot matnlarni tarjima qilishga yordam beradi, muhim voqealarni eslatadi va hokazo.
5. Boshqa xizmatlar bilan integratsiya. Robot xabarnomalarni yuborishi, aqlli uyni boshqarishi va hokazo.
6. Joylarni qidirish. Telegram-bot mehmonxonalar, kinoteatrlar, restoranlar va boshqa muassasalarni qidirishga yordam beradi.
7. Operatsiyalar. Robot sizga chiptalarni bron qilish, buyurtma berish, taksiga qo'ng'iroq qilish va h. k.

Botlardan foydalanib, foydalanuvchilarga buyurtma berish, ularni to'lash, buyurtmaning holatini o'rganish, etkazib berishni kuzatish va h.k. robotlar xaridorlarni chegirmalar, maxsus bonus dasturlari haqida xabardor qilish va hatto maslahat berishga yordam beradi.

Botlar – bu ma'lum funktsiyalarni avtomatlashtirish uchun mo'ljallangan maxsus dasturlar.



So'ngi yillarda Telegram deya atalmish mo'jizakor tizim bizning hayotimizda kerakli va qadrli buyumga aylanib ulgurdi. Telegram bepul, tejamkor, tezkor va foydalanish juda oson bo'lganligi sabab O'zbekistonda eng ko'p foydalanuvchilarga ega messenjer nomiga sazavor bo'ldi. Ushbu tizim qachonlardur shunchaki habarlar almashinuvi uchun yaratilgandi,. Ushbu sehrli tarmoq orqali istalgan yaqinimizga, do'stlarimizga va hamkasblarimizga hatlar, musiqalar, vediolar, hujjatlar hattoki joyashuv nuqtamizni ham bir zumda yuborishimiz mumkin.



Telegram botlarning eng foydali hususiyati, botlar bizning o'rnimizga foydalanuvchilar bian bemaol suhbat qurishi mumkin. Natijada hizmat ko'rsatishi lozim bolgan inson, minglab foydalanuvchilar tomonidan har doim beriladigan bir xil savollarga qayta — qayta javob berishi shart emas, foydalanuvchilar ham o'z savollariga javob olish uchun navbat kutib vaqt o'tkazmaydi. Tan olish kerak bu juda qoyilmaqom texnologiya, Ahir botga doim beriladigan savollar va javoblarni o'rgatib qo'yish kifoya qolganini bot o'zi amallaydi.

Sizing shaxsiy yordamchi botingiz.

Ha bilaman siz ham telegramdan harkuni foydalanasiz, har kuni do'stlaringiz, tanishlar va begonalar sizdan maslahatlar so'rashadi yoki yoki siz tarmoqdan tashqarida bolganingizda sizga yozishadi, Balki shu zhoti javob

ololmaganlar hafa ham bo'lishi mumkin. Bugun siz bilan, sizning shaxsiy chat botingizni yaratamiz va kiyingi dasrlarda asta sekinlik bilan uni rivojlantirib boramiz. Ushbu botni shaxsiy yoki biznes maqsadda ishlatish sizning istagingizga bo'g'liq holos.

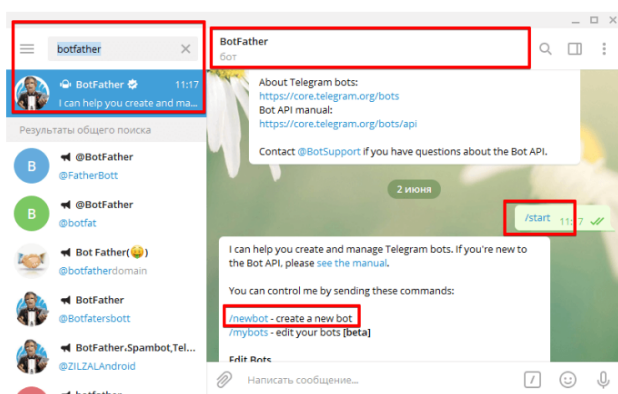


Telegramda shaxsiy botimizni ro'yxatdan o'tkazishdan boshlaymiz.

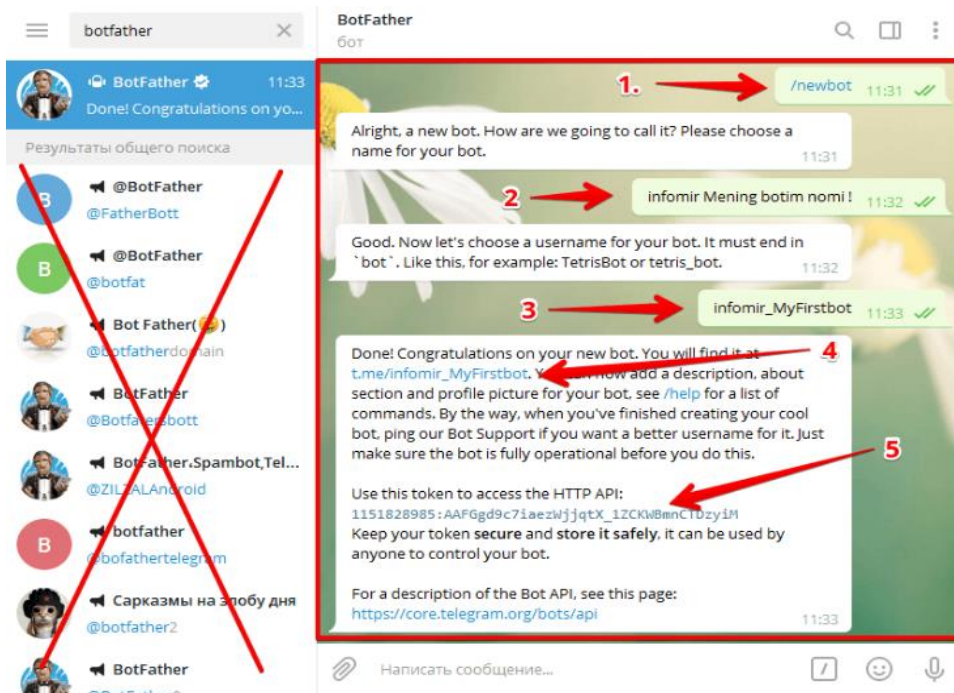
Botni Ro'yxatdan o'tkazamiz:

Ishni Telegramdagi barcha botlarning otasi bo'lmish **Botfather**da bot yaratishdan boshlaymiz:

BotFatherni ishga tushurib, «/newbot» buyrug'ini yuboramiz.

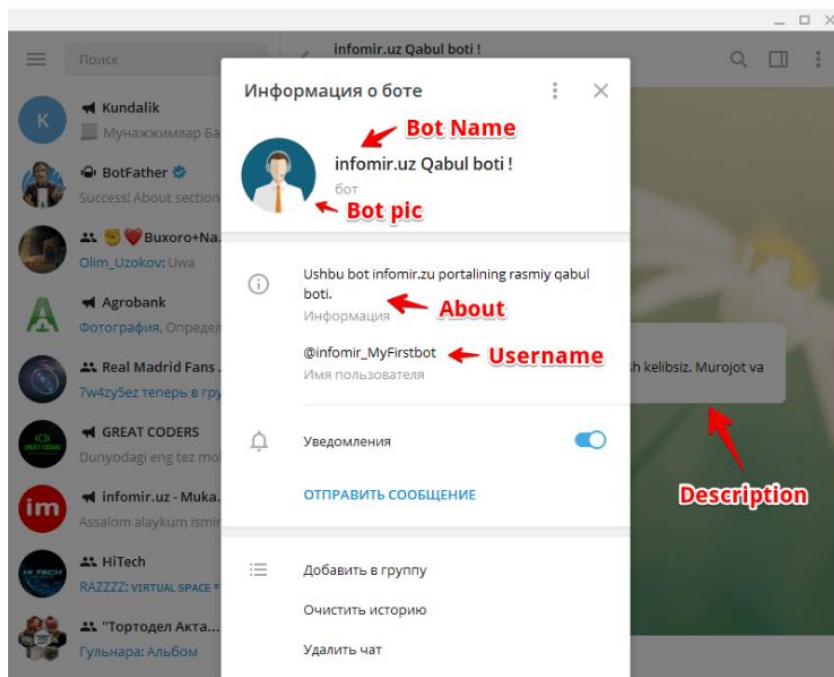


Avvaliga bot nomi kiyin bot uchun username kiritamiz. Bot nomi istalagan shaklda bo'lishi mumkin, Ammo username «**bot**» jumlasini bilan tugallanishi shart !



Botimizga surat va qo'shimcha tavsif qo'shishimiz uchun avvaliga **/mybots** buyrug'ini yuborib, yaratilgan botimizni tanlaymiz.

«**Edit bot**» tugmasini bosamiz va navbat bilan botimizni tahrirlaymiz.



Bot uchun dasturlashni boshlaymiz.

Telegram bot dasturiy qismini ishga tushurishimiz uchun bizga web hosting + ssl sertifikat bilan domen kerak boladi.

Demak dasturlashni boshladik:

Saytimizda istalgan nom bilan PHP fayl yaratamiz, Masalan «chat_bot.php» va fayl ichida bot dasturini boshlaymiz.

BotFather taqdim etgan, bot tokeni yani kaliti uchun o'zgarmas constant yaratib olamiz. Constant nomini «API_KEY» deb nomladim.

```
<?php
```

```
define('API_KEY',
```

```
'1016572094:AAGrwwZhjmMHTUaC4VAHgforR3QlFkSbF8s');
```

Botni boshqaradigan insonning telegram id raqami va Ism Familyasini yoki kompaniya nomini o'zgaruvchiga saqlaymiz. Telegram id raqamini olish uchun ushbu botga «/start» buyrug'uni bering. Botga o'tish: http://t.me/infomiruz_idbot

```
$Manager = "150105698";
```

```
$compame = "infomir.uz";
```

bot dasturi barcha amallarini kerakli shaklda Telegram Bot API ga yuboradigan asosiy funksiyani yozamiz

```
function bot($method, $datas = []){
```

```
    $url = "https://api.telegram.org/bot".API_KEY."/". $method;
```

```
    $ch = curl_init();
```

```
    curl_setopt($ch, CURLOPT_URL, $url);
```

```
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
```

```
    curl_setopt($ch, CURLOPT_POSTFIELDS, $datas);
```

```
    $res = curl_exec($ch);
```

```
    curl_close($ch);
```

```
    if (!curl_error($ch)) return json_decode($res);
```

```
};
```

Foydalanuvchilarning Ism Familyasi yoki Yozgan murojati tarkibida uchrashi mumkin bo'lgan kodga a'loqador belgilardan himoya funksiyasini yozamiz

```
function html($text){  
    return str_replace(['<','>'],['&#60;','&#62;'],$text);  
};
```

Telegram Bot API dan kelgan JSON formatdagi ma'lumotni «json_decode» funksiyasi yordamida «stdClass Object» shaklida qabul qilamiz

```
$update = json_decode(file_get_contents('php://input'));
```

Telegram Bot API bizga yuborayotgan javobni ko'rishimiz uchun hozircha ushbu kodni qo'shib qo'yamiz, ushbu kod bot ishga tushganda javobni yangi «log.txt» faylga yozib qo'yadi.

```
//test log
```

```
file_put_contents("log.txt",file_get_contents('php://input'));
```

Botni ishga tushurish uchun Maxsus «setWebhook» funksiyasi orqali Telegram API ga botimiz kodlari qaysi manzilda joylashganligi haqida habar berishimiz lozim. Buning uchun avval Webhook murojat urlini tayyorlaymiz:

[https://api.telegram.org/botBuyerda Token/setWebHook?url=php fayl manzili](https://api.telegram.org/botBuyerda_Token/setWebHook?url=php_fayl_manzili)

Mening botim uchun mana bunday shaklda:

```
https://api.telegram.org/bot1151828985:AAFu_2fo/setWebHook?url=https://infomir.uz/chat_bot.php
```

Ushbu url so'rovni browserda ishga tushuramiz va agar hammasini to'g'ri qilgan bo'lsangiz javobi mana bunday ko'rinishda boladi:

```
{"ok":true,"result":true,"description":"Webhook was set"}
```

Botimiz ishga tushdi ammo hozircha bizga hech nima qaytarmaydi, Chunki dastur davomini yozishimiz lozim.



Ammo nima yozishni bilmayman deysizmi ? Ahir bizda endi log fayli bor shuni ochib ko'ramiz:

```
{
  "update_id":445311974,
  "message":{
    "message_id":39,
    "from":
    {
      "id":767265901,"is_bot":false,
      "first_name":"SaidAbbos",
      "last_name":"Khudoykulov",
      "username":"Mujik_s_Borodoy",
      "language_code":"ru"
    },
    "chat":
    {
      "id":767265901,
      "first_name":"SaidAbbos",
      "last_name":"Khudoykulov",
      "username":"Mujik_s_Borodoy",
```



```

    "type":"private"
  },
  "date":1591090681,
  "text":"/start",
  "entities":[{"offset":0,"length":6,"type":"bot_command"}]}
}

```

stdClass tarkibida kelgan ma'lumotlar ichidan o'zimizga kerakli ma'lumotlarni o'zgaruvchilarga himoya funksiyamiz orqali tozalab saqlaymiz

```

// message variables
$message = $update->message;
$text = html($message->text);
$chat_id = $message->chat->id;
$from_id = $message->from->id;
$message_id = $message->message_id;
$first_name = $message->from->first_name;
$last_name = $message->from->last_name;
$full_name = html($first_name . " " . $last_name);

// replymessage
$reply_to_message = $message->reply_to_message;
$reply_chat_id = $message->reply_to_message->forward_from->id;
$reply_text = $message->text;

Demak dastur davomida shunday yozamiz:
// Agar yozgan odam $Manager bo'lmasa ushbu kod qismiga kiramiz
if ($chat_id != $Manager) {
    // Agar yozilgan habar /start bolsa, yani yangi foydalanuvchi
    // botni ishga tushursa ushbu kod bajariladi
    if ($text == "/start") {

```

// Foydalanuvchiga Manager yoki kompaniya nomidan salom
yo'llaymiz.

```
$reply = "Assalom Alaykum <b>" . $full_name . "</b>", " . $compane .
```

```
"
```

```
Qabul Botiga Xush Kelibsiz !\nMurojat Yo'llashingiz Mumkin
```

```
👉";
```

```
bot('sendmessage', [ // maxsus bot funksiyamiz orqali sendmessage ga
```

```
'chat_id' => $chat_id, //foydalanuvchi id raqami va
```

```
'text' => $reply, // habar matnini
```

```
'parse_mode' => "HTML", //html formatda yuboramiz.    ]);
```

// Yangi foydalanuvchi malumotlarini manajerga aniq vaqt bilan
yuboramiz.

```
$reply = "Yangi mijoz:\n" . $full_name . "\n
```

```
👉 👉 <a href='tg://user?id=' . $from_id . "'>" . $from_id .
```

```
"</a>\n
```

```
" . date('Y-m-d H:i:s') . "";
```

```
bot('sendmessage', [ // maxsus bot funksiyamiz orqali sendmessage ga
```

```
'chat_id' => $Manager, //Manager id raqami va
```

```
'text' => $reply, // habar matnini
```

```
'parse_mode' => "HTML", //html formatda yuboramiz.    ]);
```

// Foydalanuvchidan kelgan ilk /start habarini javob

// bera olishi uchun managerga yuboramiz.

```
bot('forwardMessage', [ // maxsus bot funksiyamiz orqali  
forwardMessage ga
```

```
'chat_id' => $Manager, //Manager id raqami va
```

```
'from_chat_id' => $chat_id, // foydalanuvchi bilan bot o'rtasidagi  
chat id raqami
```

```
'message_id' => $message_id, // va foydalanuvchi yuborgan habar id  
raqamini yuboramiz.
```

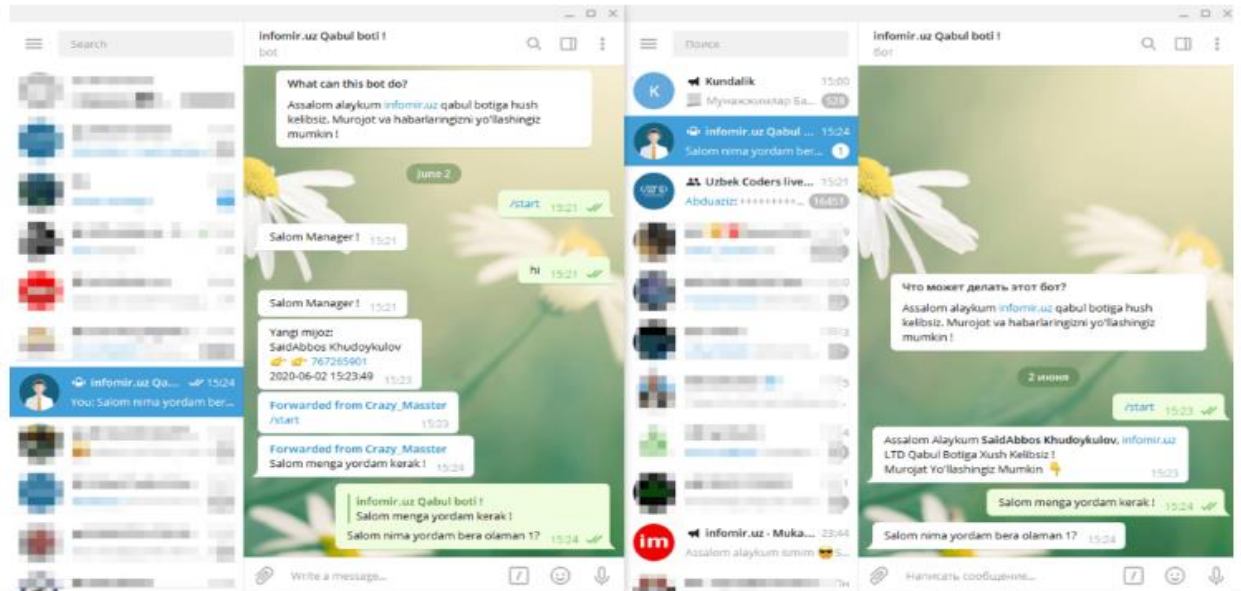
```

    ]);
    // Tekshiramiz foydalanuvchi /start komandasidan boshqa narsa yozgan
bo'lsa
    }else if ($text != "/start"){
        // Foydalanuvchidan kelgan habarni javob bera olishi uchun managerga
yuboramiz.
        bot('forwardMessage', [ // maxsus bot funksiyamiz orqali
forwardMessage ga
            'chat_id' => $Manager, //Manager id raqami va
            'from_chat_id' => $chat_id, // foydalanuvchi bilan bot o'rtasidagi
chat id raqami
            'message_id' => $message_id, // va foydalanuvchi yuborgan habar id
raqamini yuboramiz.
        ]); }
    // Yoki agar $Manager yozgan bo'lsa ushbu kod qismiga kiramiz
}else if($chat_id == $Manager){
    // Agar manager bot qayta yuborgan hatga javob berish orqali habar
yuborsa,
    if(isset($reply_to_message)){
        // Manager habarini bot qayta yuborgan habar egasiga bot nomidan
yuboramiz
        bot('sendmessage', [ // maxsus bot funksiyamiz orqali sendmessage ga
            'chat_id' => $reply_chat_id, // bot qayta yuborgan habar id raqami va
            'text' => $reply_text, // manager yuborgan habarni
            'parse_mode' => "HTML", //html formatda yuboramiz.
        ]); }
    // Manager profilidan botni tekshirib ko'rish uchun botdan managerga
salom !
    if($text == "hi" or $text == "/start"){

```

```
bot('sendmessage', [
    'chat_id' => $Manager,
    'text' => "Salom Manager !",    ]);  }}
```

Natija:



Nazorat savollari

1. Telegram bot bilan ishlovchi mobil ilovalar haqida umumiy ma'lumot bering?
2. Shaxsiy telegram bot yaratish ketma-ketligi qay tartibda amalga oshirilishini tushuntiring.

3.3. Flutter: Android Studioda to'laqonli mobil ilova: oddiy messenger yaratish

Reja:

1. Android Studioda oddiy messenger yaratish

Messenger **iPC (Inter-Process Communication) bilan boshqa jarayonda xabarlarni yuborish uchun ishlatiladi** va shu bilan mijoz va server o'rtasidagi aloqani ta'minlaydi. Messenger ishlovchiga (handler) ulanadi, shuning uchun barcha ishlar bitta ishlovchining oqimida bo'ladi.

Xizmatni bog'lash uchun Messenger exmeplyar-ga bog'lanishni usulga onBind() o'tkazish va Messenger misolini yaratishda ishlovchining havolasini yuborish kerak. Keling, ishlovchiga ulangan Messengerni ishlatib, boshqa jarayonda ishlaydigan xizmatni yaratishning kichik misolini ko'rib chiqaylik:

```
1 package com.javadevblog.messengerexampleapp;
2 import android.app.Service;
3 import android.content.Intent;
4 import android.os.Bundle;
5 import android.os.Handler;
6 import android.os.IBinder;
7 import android.os.Message;
8 import android.os.Messenger;
9 import android.os.RemoteException;
10 import android.support.annotation.Nullable;
11 import android.widget.Toast;
12 public class SimpleServiceIPC extends Service {
13     public static final int TASK_1 = 1;
14     public static final int TASK_RESPONSE_1 = 2;
15     Messenger messenger = new Messenger(new IncomingHandler());
16     @Nullable
17     @Override
18     public IBinder onBind(Intent intent) {
19         return messenger.getBinder();
20     }
```

```

21 class IncomingHandler extends Handler {
22     @Override
23     public void handleMessage(Message msg) {
24         Message message;
25         Bundle bundle = new Bundle();
26         String messageText;
27         switch (msg.what) {
28             case TASK_1:
29                 messageText = msg.getData().getString("message");
30                 message = Message.obtain(null, TASK_RESPONSE_1);
31                 Toast.ushtubusaytga tashrif buyurib , siz qonuniy yoshga
32 To'lganligingizni tasdiqlaysiz va ushbu veb-saytga tashrif buyurganingiz
33 mahalliy qonunchilikning buzilishi emasLENGTH_SHORT).show();
34                 bundle.putString("message_res", messageText);
35                 message.setData(bundle);
36                 Messenger activityMessenger = msg.replyTo;
37                 try {
38                     activityMessenger.send(message);
39                 } catch (RemoteException e) {
40                     e.printStackTrace();
41                 }
42                 break;
43             default:
44                 super.handleMessage(msg);
45         }
46     }
}

```

Koddan ko'rinib turibdiki, xizmat Activity bilan muloqot qilish uchun Messengerni ishlatadi. Service va Acitivity o'rtasidagi barcha ma'lumotlar Bundle ob'ektlarida

uzatiladi, bu esa har qanday ma'lumotni uzatish uchun juda qulaydir. Xizmatning o'zi `incominghandler` sinfidan foydalanadi — qabul qilingan xabarni qayta ishlash uchun `Handler` merosxo'ri.

Keling, ushbu xizmatni serverga xabar yuborish va `Messenger` ishlovchisini ishlatib, serverga javob berish uchun faollashtirish yarataylik:

```
1 package com.javadevblog.messengerexampleapp;
2 import android.content.ComponentName;
3 import android.content.Context;
4 import android.content.Intent;
5 import android.content.ServiceConnection;
6 import android.os.Bundle;
7 import android.os.Handler;
8 import android.os.IBinder;
9 import android.os.Message;
10 import android.os.Messenger;
11 import android.os.RemoteException;
12 import android.support.annotation.Nullable;
13 import android.support.v7.app.AppCompatActivity;
14 import android.view.View;
15 import android.widget.Button;
16 import android.widget.EditText;
17 import android.widget.Toast;
18 public class MainActivity extends AppCompatActivity {
19     private final Messenger mActivityMessenger = new Messenger(new
20 ResponseHandler(this));
21     private Button mButtonSend;
22     private EditText mEditTextMessage;
23     private Messenger mMessenger;
24     private boolean isBound;
```

```

25     ServiceConnection serviceConnection = new ServiceConnection() {
26         @Override
27         public void onServiceConnected(ComponentName name, IBinder
28 service) {
29             mMessenger = new Messenger(service);
30             isBound = true;
31         }
32         @Override
33         public void onServiceDisconnected(ComponentName name) {
34             isBound = false;
35             mMessenger = null;
36         }
37     };
38     @Override
39     protected void onCreate(@Nullable Bundle savedInstanceState) {
40         super.onCreate(savedInstanceState);
41         setContentView(R.layout.activity_main);
42         mEditTextMessage = (EditText) findViewById(R.id.et_message);
43         mButtonSend = (Button) findViewById(R.id.btn_send);
44         mButtonSend.setOnClickListener(new View.OnClickListener() {
45             @Override
46             public void onClick(View v) {
47                 String messageText = mEditTextMessage.getText().toString();
48                 if (messageText.isEmpty()) {
49                     Toast.makeText(MainActivity.bu, " xabar kiriting!",
50 Toast.LENGTH_LONG).show();
51                 } else {
52                     Message message = Message.obtain(null,
53 SimpleServiceIPC.TASK_1);

```



```

54         Bundle bundle = new Bundle();
55         bundle.putString("message", messageText);
56         message.setData(bundle);
57         message.replyTo = mActivityMessenger;
58         try {
59             mMessenger.send(message);
60         } catch (RemoteException e) {
61             e.printStackTrace();
62         }
63     }
64 }
65 });
66 }
67
68 @Override
69 protected void onStart() {
70     super.onStart();
71     if (!isBound) {
72         // start service here
73         Intent intent = new Intent(MainActivity.this, SimpleServiceIPC.class);
74         bindService(intent, serviceConnection,
75 Context.BIND_AUTO_CREATE);
76     }
77 }
78 @Override
79 protected void onStop() {
80     super.onStop();
81     isBound = false;
82     mMessenger = null;

```

```

83     }
84     private static class ResponseHandler extends Handler {
85         private Context mContext;
86         ResponseHandler(Context context) {
87             mContext = context;
88         }
89         @Override
90         public void handleMessage(Message msg) {
91             switch (msg.what) {
92                 case SimpleServiceIPC.TASK_RESPONSE_1:
93                     String result = msg.getData().getString("message_res");
94                     Toast.makeText(mContext, "iPC xizmatidan keldi:" + result,
95 Toast.LENGTH_LONG).show();
96                     break;
97                 default:
98                     super.handleMessage(msg);
99             }
100         }
101     }
    }

```

Ushbu ilovada biz ham bo'lishi kerak ServiceConnection mijoz va server o'rtasidagi aloqani yaratish uchun xizmatga.ResponseHandler-Handler vorisi Messenger'a javobini qayta ishlaydi va foydalanuvchi tajribasini yangilaydi.

Endi ilova bo'limda bizning xizmat qo'shishAndroidManifest.xml:

```

1 <service android:name=".SimpleServiceIPC" android:process=":remote"/>

```

Layout fayliga faqat kirish maydonini va xabarni yuborish uchun tugmani qo'shing:

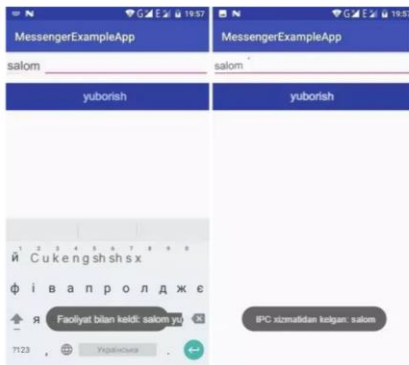
```

1 <?xml version="1.0" encoding="utf-8"?>

```

```
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:orientation="vertical"
7   tools:context="com.javadevblog.messengerexampleapp.MainActivity">
8   <EditText
9     android:id="@+id/et_message"
10    android:layout_width="match_parent"
11    android:layout_height="wrap_content"
12    android:hint="xabarni kiriting..." />
13  <Button
14    android:id="@+id/btn_send"
15    android:layout_width="match_parent"
16    android:layout_height="wrap_content"
17    android:layout_marginTop="10dp"
18    android:background="@color/colorPrimaryDark"
19    android:matn="yuborish"
20    android:textColor="#fff"/>
21
22 </LinearLayout>
23
24
```

Endi dasturimizni ishga tushiramiz va natijani ko'rib chiqamiz:



Ko'rib turganingizdek, Messenger server va mijoz o'rtasida muloqot qilishning ajoyib usuli.

Nazorat savollari

1. Messenjer deganda nimani tushunasiz?
2. Android Studioda oddiy messenjer yaratish ketma-ketligini tushuntiring?


3.4. Flutterda yozilgan Android ilovani iOS ga o'girish, ilovani Play Market va Apple Store ga joylash


Reja:

1. Flutterda yozilgan Android ilovani iOS ga o'girish,
2. ilovani Play Market va Apple Store ga joylash

Ilovani Google Play-ga joylashtirish

1. [Google Play-ga ruxsat](#) berish. Agar ilovani sotishni rejalashtirmoqchi bo'lsangiz, har bir yangi dastur uchun hisob yaratish lozim bo'ladi.
2. [Google Play konsolida ro'yxatdan](#) o'tish(Apple Developer dasturida bo'lgani kabi - ma'lumotlarni to'ldirish va boshqalar).
3. To'lov. Bu yerda ro'yxatga olish narxi bir marta \$ 25.

 Google Play Console

 ivanov.ivan1977@gmail.com

Google Play-ga ruxsat berish

Ushbu ishlab chiquvchi hisob tanlangan Google hisobiga bog'liq bo'ladi. Agar mavjud ishlab chiquvchi hisobiga qo'shilishga harakat qilsangiz, taklifnomani so'rang boshqaruv.

Tashkilot uchun ishlab chiquvchi hisobini yaratishda shaxsiy hisobdan foydalanishni tavsiya etmaymiz.
 Siz har qanday mavjud elektron pochta manzili bilan Google hisobini yaratishingiz mumkin. [Batafsil](#)

Hisobni yaratish uchun 25 AQSH dollari miqdorida bir martaik ro'yxatdan o'tish badalini to'leshingiz kerak. Ro'yxatga olishni yakunlash uchun sizdan haqiqiy hisobga olish kartasini taqdim etishni so'rashimiz mumkin. Agar sizning shaxsingizni tasdiqlay olmasak, ro'yxatdan o'tish to'lovi qaytarilmaydi.

Ommaviy ishlab chiquvchi nomi *

Ushbu nom Google Play-da foydalanuvchilarga ko'rinadi. 0/50

Qo'shimcha elektron pochta manzili *

Ushbu elektron pochta manzilidan Google Hisobingizning asosiy manziliga qo'shimcha ravishda siz bilan bog'lanish uchun foydalanishimiz mumkin. Google Play-da foydalanuvchilarga ko'rinmaydi.

Telefon raqami *

Raqamni ortiqcha belgisi, mamlakat va mintaqa kodi bilan belgilang. Bu sizning hisob haqida ma'lumot yuborish uchun ishlatiladi. Google Play foydalanuvchilari uni ko'rmaydilar.

Ishlab chiquvchi shartnomasi shartnomasini qabul qilaman dasturiy *

va men Google Play mahsulotlardan foydalanish shartlari, men ular bilan Google hisobingizni bog'lashga rozilik bildiraman va 18 yoshga kirganimni tasdiqlayman

Men Google Play konsolidan foydalanish shartlarini qabul qilaman va ular bilan Google hisobingizni bog'lashga rozilik bildiraman

* Yulduzcha majburiy maydonlarni belgilaydi.

[Hisob yaratish va to'lash](#)

Ilovani App Store-ga joylashtirish

1. [Apple ID-ga kirish kerak.](#)
2. Keyin [Apple Developer dasturiga ro'yxatdan o'tishingiz kerak.](#) Sozlamalarda ikki faktorli autentifikatsiyani yoqish kerak.
3. Shaxsiy ma'lumotlarni Apple Developer-ga to'ldiring.
4. Siz kimligizni tanlang: kompaniya, yakka tartibdagi tadbirkor va boshqalar.
5. Foydalanuvchi shartnomasini qabul qiling.
6. To'lov qiling. Endi ishlab chiquvchiga obuna narxi yiliga \$ 99. Obuna uzaytirilmasa, sizning ilovangiz AppStore-da yuklab olish uchun mavjud bo'lmaydi.

Ikki kun ichida hisobingiz yaratiladi va ilovani yuklab olish mumkin

Apple ID uchun ikki faktorli autentifikatsiya

Ikki faktorli autentifikatsiya qilish sizning hisobingizga kira oladigan yagona shaxs bo'lishingizni ta'minlash uchun mo'ljallangan Apple ID uchun qo'shimcha xavfsizlik qatlami agar kimdir sizning parolingizni bilsa ham, yozuvlar.



bo'ladi.

Ro'yxatdan O'tish Uchun Nima Kerak

✔ Shaxs sifatida ro'yxatdan o'tish

Agar siz jismoniy shaxs yoki yakka tartibdagi tadbirkor bo'lsangiz/biznes bir kishi uchun, ikkita faktorli autentifikatsiya bilan Apple ID identifikatoringiz bilan kirishni boshlang. Sizing asosiy shaxsiy ma'lumotlaringizni, shu jumladan sizning yuridik nomi va manzili.

✔ Tashkilot sifatida ro'yxatdan o'tish

Agar siz tashkilotingizni ro'yxatdan o'tkazsangiz, avval siz ikkita faktorli autentifikatsiya bilan birga Apple ID-ga ega bo'lishingiz kerak:

D-U-N-S° raqami

sizning tashkilotingiz D-U-N-S° raqamiga ega bo'lishi kerak, shunda biz sizning tashkilotingiz kimligini va yuridik shaxsning maqomini tasdiqlashimiz mumkin. Ushbu noyob to'g'iz xonali raqamlar Dun & Bradstreet tomonidan tayinlanadi va standart biznes identifikatorlari sifatida keng qo'llaniladi. Sizing tashkilotingiz allaqachon D-U-N-S° raqamiga ega ekanligini tekshirishingiz mumkin va agar kerak bo'lsa, uni so'rashingiz mumkin. Ular bepul ko'pgina yurisdiksiyalarda. Ko'proq bilib oling >

D-U-N-S° raqami

sizning tashkilotingiz D-U-N-S° raqamiga ega bo'lishi kerak, shunda biz sizning tashkilotingiz kimligini va yuridik shaxsning maqomini tasdiqlashimiz mumkin. Ushbu noyob to'g'iz xonali raqamlar Dun & Bradstreet tomonidan tayinlanadi va standart biznes identifikatorlari sifatida keng qo'llaniladi. Sizing tashkilotingiz allaqachon D-U-N-S° raqamiga ega ekanligini tekshirishingiz mumkin va agar kerak bo'lsa, uni so'rashingiz mumkin. Ular bepul ko'pgina yurisdiksiyalarda. Ko'proq bilib oling >

Yuridik shaxsning maqomi

Sizing tashkilotingiz yuridik shaxs bo'lishi kerak, shunda u Apple bilan shartnoma tuzishi mumkin. Biz ma'lumotlar bazalarini, soxta korxonalami, savdo-sotiqni qabul qilmaymiz ismlar yoki filiallar.

Yuridik jihatdan majburiy organ

sizning tashkilotingizni Apple tuzuvchi dasturida ro'yxatdan o'tkazgan shaxs sifatida sizning tashkilotingizni qonuniy bitimlarga bog'lash uchun qonuniy vakolatlarga ega bo'lishingiz kerak. Siz tashkilotingiz egasi/asoschisi, ijro etuvchi jamoa a'zosi, katta loyiha rahbari yoki yuqori lavozimli xodim tomonidan sizga berilgan qonuniy vakolatga ega bo'lgan xodim bo'lishingiz kerak.

Veb-sayt

Tashkilotingizning veb-sayti ommaviy bo'lishi kerak va domen nomi bo'lishi kerak tashkilotingiz bilan bog'lanish.

[Ro'yxatdan O'tishni Boshlang](#)

To'lovdan so'ng darhol Google Play konsoliga kirish mumkin va ilovani yuklab olishingiz mumkin.

Ilova tavsifi

Tavsifdagi birinchi jumlar ilovaning asosiy g'oyasini maksimal darajada tasvirlab berishi va Foydalanuvchining muammolarini qanday hal qilishini aytib berishi kerak. Ushbu paragrafning maksimal uzunligi 255 ta belgidir.

Matn foydalanuvchi uchun foydali bo'lishi uchun monosyllabic jummalarni ishlatish kerak — bu ma'noni osonroq etkazishga yordam beradi. U dasturni o'rnatish orqali nima olishini yoki tan olishini tushunishi kerak. Uni harakatga chaqiring. Raqobatbardosh ustunlikni va mahsulotni bozorda joylashtirishni tavsiflang.

Tavsifda nimani ko'rsatishim mumkin?

1. Maxsus xizmatlar;
2. Xususiyatlari va tafsilotlari (asosiy matn 2-3 paragraf);
3. Ularning tavsifi bilan funktsiyalar ro'yxati. Ro'yxatda 3 dan 7 funktsiyalariga ega bo'lish tavsiya etiladi va ularning barchasi nom va qisqa tavsifga ega bo'lishi kerak. Ba'zan nomi nomi nomi bilan atalgan, matn bilan bir jumla tomonidan ta'qib;
4. Yangi nima? Qanday xatolarni tuzatganingizni va qanday xususiyatlarni qo'shganingizni ayting.

Maslahatlar:

1. Kalit so'zlar faqat nomlar emas, balki butun Tavsifning kontekstiga kiritilishi kerak.
2. Ta'rif ikkinchi shaxsdan bo'lishi kerak. Foydalanuvchi nuqtai nazaridan ilova haqida bizga xabar berishi lozim.

ASOdesk tadqiqotlari

Matn rangi kontrastli fon bo'lishi kerak. Rasmdagi matn katta va matn ko'p emas. Screenshotlar tavsifida fe'llarni ishlatish eng yaxshisidir. Ular foydalanuvchini harakatga yo'naltiradi va dasturni yuklab olish orqali qanday foyda olishini aytib berishadi. [Ekran tasvirlari uchun tavsif bo'lishi kerak:](#) . Aniq;

Fon rangi va shriftni qanday tanlash mumkin?

ASOdesk tadqiqotlari [shuni](#) ko'rsatdiki, fon rangi sifatida oq rang App Store do'konidagi mashhur bepul ilovalarning 38% va Google play-da 42% dan foydalanadi. Bundan tashqari, ko'pincha ko'k va yashil quyruq ranglar ishlatiladi. Onlayn do'konlar uchun vaziyat teskari: bu erda yorqin, to'yingan ranglar tez-tez ishlatiladi.

Matn rangi kontrastli fon bo'lishi kerak. Rasmdagi matn katta va matn ko'p emas. Screenshotlar tavsifida fe'llarni ishlatish eng yaxshisidir. Ular foydalanuvchini harakatga yo'naltiradi va dasturni yuklab olish orqali qanday foyda olishini aytib berishadi.

Ekran tasvirlari uchun tavsif bo'lishi kerak:

1. Aniq;
2. Axborot;
3. Qisqa.

Screenshotlar sizning mahsulotingiz uslubining umumiy tushunchasiga mos kelishi kerakligini yodda tutish kerak bo'ladi.

Video ishlatish qanday amalga oshiriladi?

Agar ilovada namoyish etilishi kerak bo'lgan ko'plab xususiyatlar mavjud bo'lsa, videoni ishlatish mantiqan to'g'ridir. Video o'yinlar va foto-video muharrirlari uchun ko'proq mos keladi. Sizning mahsulotingiz uchun videodan foydalanishning maqsadga muvofiqligini aniqlash uchun A/b testini o'tkazish mumkin.

Odatda video yo'nalishini tanlash dastur formatiga tushadi. Ammo, [Google](#) dastur vertikal bo'lsa ham gorizontal video qilish tavsiya qiladi. Gorizontal videoda siz ko'proq matnga mos kelishingiz mumkin va u yanada foydali ko'rinadi. Bundan tashqari, Google Play uchun videolar avval YouTube-ga yuklanadi va u erda landshaft orientatsiyasi videosi yaxshiroq ko'rinadi.

Google play-da 2 daqiqagacha videolarni yuklashingiz mumkin bo'lsa-da, video 30 soniyadan oshmasligi kerak. Chunki u avtomatik ravishda yarim daqiqa davomida ijro etiladi, undan keyin foydalanuvchi YouTube-ga o'tishi kerak.

Aksariyat foydalanuvchilar videoni ovozsiz tomosha qilishadi.

Dasturni joylashtirish. O'rnatish va Rev yuborish

Google Play-da moderatsiya qilish

Revyu uch kungacha davom etishi mumkin. Nashrdan oldin, ilova Google qoidalariga mos kelishiga ishonch hosil qilishingiz kerak. Buzilishlarni oldini olish uchun siz:

1. Metadata to'g'ri kiritilganligini tekshiring - to'liq va qisqa tavsifda keraksiz tafsilotlar va foydalanuvchi sharhlari bo'lmasligi kerak;
2. Faqat mualliflik huquqi bo'lgan tasvirlardan foydalaning-ekran tasvirlari va videolarni targ'ib qilish noyob mahsulot bo'lishi kerak;
3. Google vakillari sizga murojaat qilishlari uchun aloqa ma'lumotlarini yangilang;
4. Agar yosh chegarasi so'rovnomasining savollariga to'g'ri javob berganiga ishonch komil bo'lmasa, so'rovga qaytish va uni qayta ko'rib chiqish lozim bo'ladi

Google Play App Store'dan ko'ra yaxshiroqdir. Bu yerda ishlab chiquvchi super-noyob ilovani talab qilmaydi.

Nima uchun Google ilovalarni rad etishi mumkin?

Google Play qoidalari bilan mos kelmasligi. Dastur dasturiy mahsulotlarni tarqatish to'g'risidagi shartnomani buzadi.

Ilovaning tavsifiga mos kelmasligi. Ilovadagi ma'lumot va uning haqiqiy funksiyalari bilan mos kelmasligi natijasida ilovalar rad etiladi.

Intellektual mulk huquqlarini buzish. Agar Google sizning mahsulotingizda kontentni nusxalash yoki mualliflik huquqining buzilishiga olib keladigan bo'lsa, unda sizning arizangizni nashr qilish rad etiladi.

App Store do'konida moderatsiya qilish

Apple, ilovalarning 40% ni rad etadi. Har qanday dastur yoki yangilanish qo'lda boshqariladi. Sharhlovchilar dasturni xatolar, noto'g'ri ishlash va kontentning qonuniyligi uchun tekshirishadi.

App Store do'konida rad etilishining asosiy sabablari ro'yxati:

Xatolar. Barcha foydalanuvchi bilan hamkorlik stsenariylarini dastur bilan tekshirishingiz kerak;

Noto'g'ri ekran tasvirlari. Screenshotlar dizaynining to'g'riligini tekshiring, ular ilovaning qadriyatlari va funksiyalarini aniq ko'rsatishlari kerak;

Ro'yxatdan o'tish. Agar dastur foydalanuvchini ro'yxatga olishni nazarda tutsa, sharhlovchilar uchun ariza tavsiflari bo'limida Login va parolni kiritishingiz kerak;

Noyob dastur, andozalar va saytlar emas. Ilova noyob bo'lishi kerak. Saytning mobil versiyasi ilovada ochilgan holatlar mavjud-bunday App Store darhol rad etadi;

Apple App Store gaidline bilan mos kelmasligi. [Havolani ko'rishingiz mumkin.](#)

Shunga o'xshash ilovalarni qayta topshirish. Agar siz bir xil funksiyalarga ega bo'lgan bir nechta ilovalarni ko'rib chiqsangiz, ehtimol, ular hamma narsani rad etadi.

Fichering bu bizning ishinmizning natijasidir. Google Play va App Store-da maxsus to'plamlarga kirish ko'proq yuklab olish imkonini beradi.

Fichering dasturiga kirish uchun quyidagilar kerak:

1. Foydalanuvchini noyob xususiyatlar bilan ta'minlash;
2. Gaidlinsga muvofiq rasmiylashtirilishi kerak;
3. Vizual ravishda umumiy fonga qarshi turish.

Nazorat savollari

1. Flutterda yozilgan Android ilovani iOS ga o'girish ketma-ketligini tushuntiring
2. ilovani Play Market va Apple Store ga joylash qay tartibda amalga oshiriladi.

3.5. Mobil ilovalarni ishlab chiqish va ulardan foydalanishda xavfsizlik parametrlari, ruxsatlar, ma'lumotlarni shifrlash

Reja:

1. Mobil ilovalarni ishlab chiqish va ulardan foydalanishda xavfsizlik parametrlari haqida umumiy ma'lumotlar
2. Ruxsatlar, ma'lumotlarni shifrlash

So'nggi yillarda mobil qurilmalardan foydalanish veb-ga katta qiziqish uyg'otdi, operatsion tizimlar rivojlanmoqda, biroq zararli kod va aniqlangan zaifliklar soni yildan-yilga deyarli chidamli ravishda o'sib bormoqda. Shu munosabat bilan axborotni ochiq aloqa kanallari orqali saqlash va uzatish jarayonida ruxsatsiz kirishdan himoya qilish uchun mo'ljallangan dasturiy vositalarni ishlab chiqish zarur. Bu muammoni hal qilish uchun turli usullar mavjud, masalan, steganografiya usullaridan foydalanish va ma'lum bir konteynerda himoyalangan ma'lumotlarni yashirish mumkin, ammo bu usul muayyan cheklovlarga ega, shuning uchun eng maqbul usul-maqbul himoya darajasini ta'minlash uchun kriptografik transformatsiyani qo'llashdir. Ishda axborotni nosimmetrik va assimetrik metodologiyadan foydalangan holda aylantirishning kriptografik usullari o'rganildi. Nosimmetrik metodologiya usullari orasida nosimmetrik kriptosistemalarning ikkita asosiy guruhi - blok va oqim

shifrlari ko'rib chiqiladi. Kalitlardan foydalangan holda metodologiya usullarining qiyosiy tahlillari o'tkazildi, ushbu usullarni amalga oshirishda apparat resurslarini iste'mol qilish bo'yicha tadqiqotlar o'tkazildi. Olingan natijalar asosida amalga oshirish uchun kriptografik usullar tanlangan.

Dastur Android mobil operatsion tizimi bilan ishlaydigan Java tilida amalga oshiriladi. Ilovaning arxitekturasi muhim vaqt sarflamasdan yangi kriptografik algoritmlarni qo'shish imkonini beruvchi tarzda ishlab chiqilgan. Dastur sizga kalitlarni ishlab chiqarish, foydalanuvchi ma'lumotlarini shifrlash va parolini ochish, shuningdek olingan natijalarni Google Drive bulutli saqlashga imkon beradi.

Turli xil kirish ma'lumotlarini ishlatganda amalga oshirilgan dasturiy ta'minot sinovdan o'tkazildi. 4096 bitgacha kalitlarni ishlatganda dasturiy ta'minotni tekshirish amalga oshirildi. Tekshiruv natijasida dasturiy ta'minot belgilangan vaqt ichida barcha funksiyalarni muntazam ravishda amalga oshirishi aniqlandi.

Android fayl va konteyner darajasida shifrlashni qo'llab-quvvatlaydi (to'liq diskli shifrlash). Ilovalar uchun platforma sifatida u uchinchi tomon shifrlash usullarini, masalan, xavfsiz papkalarni yoki messenjerlarda va pochta orqali shifrlangan yozishmalarni qo'llab-quvvatlashi mumkin. Bu shuni anglatadiki, qurilma chipseti tezda ma'lumotlarni shifrlash va parolini hal qilish uchun o'rnatilgan komponentni o'z ichiga oladi. Faylni dekodlash uchun tegishli kalit qurilmada saqlanadi va har bir foydalanuvchi harakati parol, barmoq izi, ishonchli qurilma va boshqalar. - shifrlangan ma'lumotlarga kirish uchun ishlatiladigan, xavfsiz elementga (plastik kartalarda ishlatiladiganlar kabi alohida mikroprotsessorga) murojaat qiladi. Android 6.0 Marshmallow-dan boshlab, barcha shifrlash funksiyalari ushbu xavfsiz Element va ma'lumotlarni shifrlash va parolini hal qilish uchun qo'llaniladigan maxsus kalit, belgilar (bir martalik yoki qayta ishlatiladigan elektron kalitlar) yordamida amalga oshirilishi mumkin. Bu

shuni anglatadiki, protsessorni joriy Tokenni taqdim etmasdan, ma'lumotlar shifrlangan bo'lib qoladi.

Android sozlamalarda parol kiritilmaguncha, tizim telefonni har safar yuklab olishda shifrlangan bo'lib qolishi mumkin. Telefondagi ma'lumotlar shifrlangan bo'lsa, u allaqachon xavfsizlikni ta'minlaydi, lekin parol kiritilgunga qadar yuklash jarayonini to'xtatib turish fayllarga kirishni oldini oladi va qo'shimcha himoya qatlami sifatida xizmat qiladi. Har qanday holatda, parol (yoki PIN kodi yoki grafik kaliti yoki barmoq izi) Secure Element orqali ma'lumotlarga murojaat qiladi va maxsus shifrlash kalitini olish imkoniyati yo'q – ma'lumotlar bilan nima qilinganligini va ularni qanday qilib eski ko'rinishga qaytarish kerakligini bilish uchun kerak bo'lgan yagona narsa.

Brauzeringizdagi yozishmalar va harakatlar ham shifrlanishi mumkin. Ehtimol, brauzerda ko'plab saytlar http o'rniga HTTPS bilan boshlangan manzilga ega ekanligiga e'tibor qaratilgandir. HTTP Hypertext Transfer Protocol (matn uzatish protokoli) degan ma'noni anglatadi va bu protokol internetda ma'lumotlarni yuborish va qabul qilish uchun ishlatiladi. O'z navbatida, HTTPS "SSL orqali HTTP" degan ma'noni anglatadi (Secure Sockets Layer, xavfsiz soket darajasi), protokolga shifrlash standartini qo'shadi. Brauzerda kiritgan barcha ma'lumotlar saytdan yuklab olgan ochiq kalit yordamida o'zgartiriladi, unga kiradi va veb-serverda mavjud bo'lgan maxsus kalit ularni parolini hal qilishi mumkin.

Qaytib keladigan ma'lumotlar o'zgartiriladi, shuning uchun ularni faqat noyob davlat kalit yordamida tushunish mumkin. Https bilan boshlangan xavfsiz saytlarga tashrif buyurishdan boshqa hech narsa qilish shart emas. Smartfon server aslida sertifikat bilan ko'rinadigan narsa ekanligini tekshiradi va brauzer ilovasi orqali ma'lumotlarni mustaqil ravishda shifrlaydi.

Shifrlangan yozishmalarni saqlash uchun odatda Google Play-dan ilovani yuklab olish lozim bo'ladi. Signal yoki WhatsApp kabi ilovalar uchidan uchigacha bo'lgan shifrlashni taklif qiladi, ya'ni ilova kalitlarni alohida kontaktlarga yoki guruhlarga tayinlaydi va xabar faqat unga yuborilgan shaxsni o'qishi

mumkin. Ko'pchilik BlackBerry Messenger-ni xavfsiz deb hisoblaydi, ammo barcha BlackBerry qurilmalarida mavjud bo'lgan bitta umumiy kalit mavjud bo'lgani uchun, bu xavfsizlik biroz tortishuvlarga sabab bo'ladi. BBM Protected yuqori darajadagi shifrlash yoki uchidan uchgacha shifrlashni talab qiluvchi guruhlar uchun mavjud. Apple iMessage ham uchidan uchgacha shifrlashga ega, ammo faqatgina barcha yozishmalar iPhone egalari uchundir.

Foydalanadigan ushbu ilovalar, boshqa har qanday messenjerdan foydalanganidek: kontakt qo'shiladi va u bilan xabar almashiladi. Faqatgina farq shundaki, bu xabarlar shifrlangan bo'lishi mumkin, shuning uchun faqat ikki tomon – yozishmalar ishtirokchilari – ularni o'qishlari mumkin.

Nazorat savollari

1. Mobil ilovalarni ishlab chiqish va ulardan foydalanishda xavfsizlik parametrlari haqida umumiy ma'lumotlar bering
2. Ruxsatlar, ma'lumotlarni shifrlash deganda nimani tushunasiz

XULOSA

Mazkur o'quv qo'llanma Mobil ilovalarini ishlab chiqish jarayonini tushuntirishga bag'ishlangan. Mobil ilovalarning rivojlanish tarixi. Dastlabki mobil qurilmalar va mobil ilovalar, zamonaviy mobil OT lar, mobil ilovalarni yaratish tillari va texnologiyalari: iOS va Android uchun dasturlash, emulyatorlar, mobil ilovalarni joylash uchun platformalar: Play Market, Apple Store internet do'konlari haqidagi ma'lumotlar berilgan. Mobil ilova ishlab chiqish tamoyillari, talablarni o'rganish va texnik topshiriqni shakllantirish, platformani tanlash, mobil ilova dizaynini ishlab chiqish, UI/UX dizaynlar, Kross pltaformali dasturlash texnologiyalari: DART => Flutter texnologiyasi, Android Studioni o'rnatish va sozlash, Flutter: Android Studio interfeysi, ilk dasturni yozish va uni emulyatorda ishlatish, komponentalar bilan tanishish, Flutter: Android Studioda asosiy komponentalar bilan ishlash: Layout, Table, ListView, Grid, List va boshqalar bilan ishlash haqida umumiy ma'lumotlar keltirib o'tilgan.

“Mobil ilovalarini ishlab chiqish” qo'llanmasi qo'lyozmasi universitetning qo'lyozmasi universitetning 5330500 - Kompyuter injiniringi (“Kompyuter injiniring”, “AT-Servis”, “Multimedia texnologiyalari”) bakalavriat talabalari uchun mo'ljallangan.

Mazkur qo'llanmadan mobil ilovalar, mobil ilovalarning asosiy tushunchalari, xususiyatlari va Android studiodan foydalangan holda mobil ilova yaratishni o'rganishni xohlovchilar foydalanishlari mumkin.

Glossariylar

Kamerofon - kamera va / yoki videokamera vazifasini bajaradigan mobil telefonlar.

Multimediya telefoni (pleer telefoni, musiqa telefoni, musiqiy telefon) - audio va videofayllarni ijro etish imkoniyatlari yaxshilangan ixtisoslashgan mobil telefonlar va boshqalar raqamli fonogrammalarni ijro etishi mumkin va xotira kartasi uyasi bilan jihozlangan.

Smartfon - to'liq operatsion tizimga ega mobil telefon (Symbian OS, Windows Mobile, Palm OS, GNU / Linux, Android, Apple iOS, MeeGo va boshqalar).

Kommunikator - mobil telefon funktsiyalariga ega bo'lgan shaxsiy kompyuter (PDA).

Ish telefoni - korporativ foydalanuvchilar uchun maxsus funktsiyalarga ega telefon.

Tasviriy telefon - telefonlari, ularning asosiy xususiyati zamonaviy ko'rinish va yoqimli funktsionallik (xususan, avtomatlashtirilgan ochilish).

Bir martali ishlatiladigan telefon - asosiy funktsional imkoniyatlarga ega telefon.

Keksalar uchun telefon (buvi, bobo) - asosiy funktsional imkoniyatlarga ega telefon, SOS tugmasi, katta tugmalar, displeyda katta bosma (ko'pincha monoxrom).

Mobil telefon, qo'l telefon — simsiz telefon turi.

Mobil ta'lim (M-Learning) – bu vaqt va makondan qat'iy nazar mobil qurilmalar orqali o'qish imkoniyatidir.

Mobil ta'lim – bu joylashgan o'rindan yoki o'quvchilarning joylashuvining o'zgarishidan cheklanmay mobil qurilmalar orqali elektron o'qishdir.

Ixchamlilik – mobil ta'lib taribiy qismlari aloqadagi ehtimolli uzilishlarni inobatga olgan holda davomiylik bo'yicha qisqa bo'lishi lozim.

Mikroergonomikaning yuqori darajadaliği – kichik ekranda ovoz va rasmlarning kichik hajmda va yuqori sifatli bo'lishi.

Har joydalik – mobil o'qitish tarkibidagi materillarni joylashgan joyidan qat'iy nazar hohlagan joyda olish mumkin.

Talab bo'yicha foydalanishga ruxsat – tabiatan mobil qurilmalar o'quvchilar uchun talab bo'yicha zarur paytda o'zining maksimal yetkazib berish imkoniyatini ishga solib zaruriy tarkibdan foydalanishni ta'minlaydi.

Mobil operatsion tizimlar – bu mobil qurilmalar (smartfonlar, tabletlar, PDA (Personal digital assistant) va boshqa mobil qurilmalar) ni boshqarish uchun mo'ljallangan tizimlar hisoblanadi.

Java yuqori darajada himoyalangan va obyektga yo'naltirilgan dashturlash tilidir.

Unity-kompyuter o'yinlarini ishlab chiqishning o'zaro platformali muhitidir.

C/C++ C tillari yuqori ko'rsatkichga ega bo'lib, ayniqsa, mobil 3D o'yinlari kabi og'ir dasturlash imkonini beruvchi til

Dart - veb ilovalar va mobil ilovalar kabi mijozlarni rivojlantirish uchun mo'ljallangan dasturlash tilidir

C# yordamida Androidda dasturlash *Xamarin* platformasini nazarda tutadi.

Standalone applications – Linux, Mac yoki Windowsga o'rnatib, kundalik ishlatadigan dasturlarimiz: mp3 player, ofis, antivirus kabilar. Ular AWT, Swing yoki JavaFX texnologiyalari orqali tuziladi.

Web Applications – tarmoq orqali ishlovchi ixtiyoriy dasturlar. Eslatma: veb dasturlar ikki qismdan, server tomon hamda klient tomon (brauzer) dan iborat.

Enterprise Applications – bu dasturlar yirik salmoqga ega bo'lib, odatda katta jabhalarda ishlatiladi. Banklar, tashkilotlar yoki astronomiya kabi sohalarda.

Mobile Applications – Mobil qurilmalarga mos dasturlarni ham Javada yozish mumkin.

Swift - bu kompilyatsiya qilingan dasturlash tili.

Emulyator - bu boshqa kompyuter yoki dasturga taqlid qiladigan dastur.

Bochs -Android uchun barcha Windows emulyatori- bu operatsion tizimning to'liq versiyasini joylashtirish uchun eng yaxshi variant

Limbo PC Emulator Windows-ni ham o'z ichiga olgan 32-bitli operatsion tizimlarning katta qismini boshqarishi mumkin

Limbo PC Emulator Windows-ni ham o'z ichiga olgan 32-bitli operatsion tizimlarning katta qismini boshqarishi mumkin.

Tayyorgarlik. Fikr tug'ilganidan keyingi qadam loyiha ustida ish rejasini tuzishdir.

Monetizatsiya. Ilovadan xaridlar, obuna yoki premium versiyalar uchun to'lovlar, reklama joylashtirish, ma'lumotlarni sotish-ilovani bepul tarqatish sarf xarajatlar monetizatsiya qilishning usullari hisoblanadi

Texnik topshiriq-loyihaning kelajagi bog'liq bo'lgan rivojlanishning birinchi bosqichi.

[Appy Pie](#) - bu hech qanday kodlashni talab qilmasdan o'zingizning ilovalaringizni ishlab chiqishga imkon beruvchi dasturiy ta'minot platformasi.

[Zoho Creator](#) - bu yangi dastur yaratishga yoki tayyor dasturdan foydalanishga imkon beruvchi ko'p platformali dastur tuzuvchisi.

[AppSheet](#) - bu sizning biznesingiz uchun buyurtma qilingan dasturlarni yaratishga imkon beradigan, boshqa kodlar bilan yozish yoki ishlab chiqishga imkon [bermaydigan](#) boshqa platforma.

[Appian](#) "past kodli" dasturlarning rivojlanishiga ko'maklashadi, ular o'zlarining reklama g'oyasini ishlab chiqish va dasturni to'ldirish orasida sakkiz hafta ichida reklama qiladilar.

[Appery](#) - bu dasturlarni ishlab chiqaruvchi, ishlab chiqaruvchilar uchun o'z dasturlarini yaratish uchun platformalar taklif qiladigan dastur ishlab chiqaruvchisi. UX (angl. user experience) so'zma-so'z "foydalanuvchi tajribasi" degan ma'noni anglatadi.

UX-dizayner (eng. user experience designer)-foydalanuvchilarning ehtiyojlarini o'rganadigan, interfeysning mantiqiy sxemalarini tuzadigan, maqsadli auditoriyada prototiplarni sinovdan o'tkazadigan va UI dizaynerlari uchun texnik topshiriqni tuzadigan dizayner.

UI (*angl. user interface*) "foydalanuvchi interfeysi" deb tarjima qilinadi.

UI dizayni-foydalanuvchi tajribasi va maqsadli auditoriyani o'rganish asosida ishlab chiqilgan prototipni ko'rish jarayoni.

UI-dizayn. Uslublar-bu mobil ilovaning "ichki qismi" ni tugatish va uning mavjudligini oshirish uchun qurilish materiallari

Kross platformali ilova - bir vaqtning o'zida bir nechta operatsion tizimlarda ishlaydigan ilovalardir.

Xamarin—Microsoft kompaniyasi tomonidan yaratilgan ramka

React Native-o'zaro kross platformali ilovalarni ishlab chiqish uchun ishlatiladigan eng mashhur fremvorklardan biri.

Flutter o'z dasturlash tili Dartdan foydalanadi. Ishlab chiquvchilar ushbu fremwork bilan ishlash uchun uni o'rganishlari kerak bo'lsa-da, u rivojlanish uchun keng imkoniyatlar ochadi.

RadioGroup—radio tugmalari to'plami uchun bir nechta istisno doirasini yaratish uchun ishlatiladi

toggleButton foydalanuvchiga ikkita holat orasidagi sozlamani o'zgartirishga imkon beradi

Switch - bu ikkita variantni tanlashi mumkin bo'lgan ikkita holatni o'zgartirish tugmachasi vidjeti. Tanlangan variantni tanlash uchun imkoniyatini beradi.

TextView-eng ko'p ishlatiladigan komponentlardan biri. Uning yordami bilan foydalanuvchi dasturda harakat qilish uchun qulayroqdir.

id - bu matn ko'rinishini aniqlash uchun ishlatiladigan atribut

textSize atributi matn ko'rinishini matni hajmini belgilash uchun ishlatiladi.

textStyle atributi matn ko'rinishini matn uslubini o'rnatish uchun ishlatiladi.

background: fon atributi matn ko'rinishini fonini o'rnatish uchun ishlatiladi. Matn ko'rinishining foniga rang yoki naqsh o'rnatishimiz mumkin.

Edit text—elektron pochta xabarnomasida foydalanish uchun matnli shaklda taqdim etilgan ma'lumotlar formati

Password. Bu [autentifikatsiya](#) jarayonida foydalanuvchi identifikatorini tekshirish uchun ishlatiladigan belgilar qatori.

E-mail. Elektron pochta (e-mail) — ma'lumotlarni uzatish tarmog'i orqali axborotlarni bir foydalanuvchi elektron qutisidan boshqasinigiga jo'natish, qabul

qilish va ma'lum vaqtgacha saqlanishini ta'minlovchi dasturiy texnik vositalar to'plami.

Number signed. Bunda faqat 0 dan 9 gacha raqamlarni kiritish mumkin.

Android-da autoCompleteTextView - bu EditText-ga o'xshash ko'rinish, faqat foydalanuvchi matn kiritishi bilan avtomatik ravishda tugatish takliflari ro'yxatini ko'rsatadi.

numColumns-ustunlar sonini aniqlaydi

verticalSpacing-jadval hujayralari orasidagi bo'sh joy hajmini belgilaydi

android:columnWidth-ustunlar kengligini o'rnatadi

stretchMode-bo'sh joy qoldig'ini belgilangan Android qiymati bilan jadval uchun ajratilganligini ko'rsatadi

RelativeLayout - ekranda boshqa elementlar qanday stacked yoki ko'rsatilishini aniqlaydi.

TextView - element matnni ko'rsatish uchun ishlatiladi..

EditText - TextView boshqaruviga o'xshash yagona farq shundaki, foydalanuvchilar matnni tahrirlashlari mumkin

Telegram-bu boshqa foydalanuvchilar bilan muloqot qilish, ma'lumot almashish va bir-biriga fayllarni yuborishning oddiy usuli emas, balki juda ko'p foydali vazifalarni bajarishga imkon beruvchi ajoyib vosita

Foydalanilgan adabiyotlar

1. Beginning App Development with Flutter: Create Cross-Platform Mobile Apps. Rap Payne. APress. 2019.
2. Google Flutter Mobile Development Quick Start Guide: Get up and running with iOS and Android mobile app development. Prajyot Mainkar, Salvatore Giordano. Packt Publishing. 2019.
3. UX-дизайн. Практическое руководство по проектированию опыта взаимодействия. Расс Унгер и Кэролайн Чендлер. Издательство Символ-Плюс. 2011.
4. UX-стратегия. Чего хотят пользователи и как им это дать. Джейми Леви. Издательство «Питер», 2017.

Qo‘shimcha adabiyotlar

5. «Mobil ilovalarni ishlab chiqish» fanidan tajriba ishlarini bajarish bo‘yicha uslubiy ko‘rsatma. Xo‘jayev O., Quryazov D. Toshkent axborot texnologiyalari universiteti Urganch filiali. 2016.
6. Fundamentals of database systems sixth edition. Ramez Elmasri. Department of Computer Science and Engineering The University of Texas at Arlington. 2011. – 261 s.

Axborot manbalari

7. www.intuit.ru
8. www.library.tuit.uz
9. www.w3schools.com
10. <https://flutter.dev/docs>
<https://developer.android.com/studio/intro>

MUNDARIJA		Sahifa
KIRISH		3
I BOB	MOBIL ILOVALARNING ASOSIY TUSHUNCHALARI	6
1.1.	Mobil ilovalarning rivojlanish tarixi. Dastlabki mobil qurilmalar va mobil ilovalar. Zamonaviy mobil OT lar.	6
1.2.	Mobil ilovalarni yaratish tillari va texnologiyalari: iOS va Android uchun dasturlash, emulyatorlar	29
1.3.	Mobil ilovalarni joylash uchun platformalar: Play Market, Apple Store internet do'konlari	51
1.4.	Mobil ilova ishlab chiqish tamoyillari, talablarni o'rganish va texnik topshiriqni shakllantirish, platformani tanlash	58
1.5.	Mobil ilova dizaynini ishlab chiqish, UI/UX dizayn	70
II BOB	KROSS PLTAFORMALI DASTURLASH TEXNOLOGIYALARI	73
2.1.	Kross pltaformali dasturlash texnologiyalari: DART => Flutter texnologiyasi, Android Studioni o'rnatish va sozlash	73
2.2.	Flutter: Android Studio interfeysi, ilk dasturni yozish va uni emulyatorida ishlatish, komponentalar bilan tanishish	83
2.3.	Flutter: Android Studioda asosiy komponentalar bilan ishlash: Layout, Table, ListView, Grid, List va boshqalar	102
2.4.	Flutter: Android Studioda oddiy kalkulyator dasturini yaratish, uni emulyatorida sinovdan o'tkazish va mobil telefonga o'rnatib ishlatish.	113
III BOB	FLUTTER: ANDROID STUDIODA MA'LUMOTLAR BAZASI BILAN ISHLASH	123
3.1.	Flutter: Android Studioda ma'lumotlar bazasi bilan ishlash,	123

	SQLite, MySQL MBBT lar bilan ulanish	
3.2.	Flutter: Android Studioda Telegram bot bilan ishlovchi mobil ilova misolida API bilan ishlashni o'rganish.	136
3.3.	Flutter: Android Studioda to'laqonli mobil ilova: oddiy messenjer yaratish	148
3.4.	Flutterda yozilgan Android ilovani iOS ga o'girish, ilovani Play Market va Apple Store ga joylash	156
3.5.	Mobil ilovalarni ishlab chiqish va ulardan foydalanishda xavfsizlik parametrlari, ruxsatlar, ma'lumotlarni shifrlash	163
XULOSA		167
GLOSSARIY		168
ADABIYOTLAR RO'YXATI		173

ОГЛАВЛЕНИЕ		Стр.
ВВЕДЕНИЕ		
ГЛАВА I.	MOBIL ILOVALARNING ASOSIY TUSHUNCHALARI	
1.1.	Mobil ilovalarning rivojlanish tarixi. Dastlabki mobil qurilmalar va mobil ilovalar. Zamonaviy mobil OT lar.	
1.2.	Mobil ilovalarni yaratish tillari va texnologiyalari: iOS va Android uchun dasturlash, emulyatorlar	
1.3.	Mobil ilovalarni joylash uchun platformalar: Play Market, Apple Store internet do'konlari	
1.4.	Mobil ilova ishlab chiqish tamoyillari, talablarni o'rganish va texnik topshiriqni shakllantirish, platformani tanlash	
1.5.	Mobil ilova dizaynini ishlab chiqish, UI/UX dizayn	
ГЛАВА II.	KROSS PLTAFORMALI DASTURLASH TEXNOLOGIYALARI	
2.1.	Kross pltaformali dasturlash texnologiyalari: DART => Flutter texnologiyasi, Android Studioni o'rnatish va sozlash	
2.2.	Flutter: Android Studio interfeysi, ilk dasturni yozish va uni emulyatorida ishlatish, komponentalar bilan tanishish	
2.3.	Flutter: Android Studioda asosiy komponentalar bilan ishlash: Layout, Table, ListView, Grid, List va boshqalar	
2.4.	Flutter: Android Studioda oddiy kalkulyator dasturini yaratish, uni emulyatorida sinovdan o'tkazish va mobil telefonga o'rnatib ishlatish.	
ГЛАВА III.	FLUTTER: ANDROID STUDIODA MA'LUMOTLAR BAZASI BILAN ISHLASH	
3.1.	Flutter: Android Studioda ma'lumotlar bazasi bilan ishlash,	

	SQLite, MySQL MBBT lar bilan ulanish	
3.2.	Flutter: Android Studioda Telegram bot bilan ishlovchi mobil ilova misolida API bilan ishlashni o'rganish.	
3.3.	Flutter: Android Studioda to'laqonli mobil ilova: oddiy messenjer yaratish	
3.4.	Flutterda yozilgan Android ilovani iOS ga o'girish, ilovani Play Market va Apple Store ga joylash	
3.5.	Mobil ilovalarni ishlab chiqish va ulardan foydalanishda xavfsizlik parametrlari, ruxsatlar, ma'lumotlarni shifrlash	
ЗАКЛЮЧЕНИЕ		
ГЛОССАРИЙ		
СПИСОК ЛИТЕРАТУРЫ		

CONTENTS		Page
INTRODUCTION.....		
CHAPTER I.	MOBIL ILOVALARNING ASOSIY TUSHUNCHALARI	
1.1.	Mobil ilovalarning rivojlanish tarixi. Dastlabki mobil qurilmalar va mobil ilovalar. Zamonaviy mobil OT lar.	
1.2.	Mobil ilovalarni yaratish tillari va texnologiyalari: iOS va Android uchun dasturlash, emulyatorlar	
1.3.	Mobil ilovalarni joylash uchun platformalar: Play Market, Apple Store internet do'konlari	
1.4.	Mobil ilova ishlab chiqish tamoyillari, talablarni o'rganish va texnik topshiriqni shakllantirish, platformani tanlash	
1.5.	Mobil ilova dizaynini ishlab chiqish, UI/UX dizayn	
CHAPTER II.	KROSS PLTAFORMALI DASTURLASH TEXNOLOGIYALARI	
2.1.	Kross pltaformali dasturlash texnologiyalari: DART => Flutter texnologiyasi, Android Studioni o'rnatish va sozlash	
2.2.	Flutter: Android Studio interfeysi, ilk dasturni yozish va uni emulyatorda ishlatish, komponentalar bilan tanishish	
2.3.	Flutter: Android Studioda asosiy komponentalar bilan ishlash: Layout, Table, ListView, Grid, List va boshqalar	
2.4.	Flutter: Android Studioda oddiy kalkulyator dasturini yaratish, uni emulyatorda sinovdan o'tkazish va mobil telefonga o'rnatib ishlatish.	
CHAPTER III.	FLUTTER: ANDROID STUDIODA MA'LUMOTLAR BAZASI BILAN ISHLASH	
3.1.	Flutter: Android Studioda ma'lumotlar bazasi bilan ishlash,	

	SQLite, MySQL MBBT lar bilan ulanish	
3.2.	Flutter: Android Studioda Telegram bot bilan ishlovchi mobil ilova misolida API bilan ishlashni o'rganish.	
3.3.	Flutter: Android Studioda to'laqonli mobil ilova: oddiy messenjer yaratish	
3.4.	Flutterda yozilgan Android ilovani iOS ga o'girish, ilovani Play Market va Apple Store ga joylash	
3.5.	Mobil ilovalarni ishlab chiqish va ulardan foydalanishda xavfsizlik parametrlari, ruxsatlar, ma'lumotlarni shifrlash	
CONCLUSION...		
GLOSSARY....		
BIBLIOGRAPHY		

