



РОССИЙСКАЯ ФЕДЕРАЦИЯ
МИНИСТЕРСТВО ОБРАЗОВАНИЯ
ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ

И. Н. ГЛУХИХ, Д. В. СЕМИХИН

КОМПЬЮТЕРНЫЕ ПРОДУКТЫ

Учебное пособие



Издательство
Тюменского государственного университета
2004

УДК 044.4(075)

ББК 32.973я73

Г 554

И. Н. ГЛУХИХ, Д. В. СЕМИХИН. КОМПЬЮТЕРНЫЕ ПРОДУКТЫ: Учебное пособие для дистанционного обучения. 2-е издание. Тюмень: Издательство Тюменского государственного университета, 2004. 108 с.

В состав учебного пособия по дисциплине «Компьютерные продукты» входят: методические указания с программой изучения дисциплины, курс лекций, вопросы для самопроверки, задания и методические указания для выполнения письменных практических работ, вопросы теста для итогового контроля, глоссарий, список литературы. Пособие предназначено для студентов, обучающихся по специальностям «Мировая экономика», «Финансы и кредит», «Бухгалтерский учет».

Рекомендовано к изданию Учебно-методической комиссией Международного института финансов, управления и бизнеса.

Рецензенты: **А. А. Ступников**, кандидат ф.-м. наук,
доцент каф. программного обеспечения
ТюмГУ

А. М. Андриянов, кандидат технических наук,
доцент кафедры управления, информатики и математики Тюменского
юридического института МВД РФ

© Тюменский государственный университет, 2004

© Институт дистанционного образования Тюменского государственного университета, 2004

© И. Н. Глухих, Д. В. Семихин, 2004

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ

1. Учебные цели

В результате изучения дисциплины «Компьютерные продукты» студент должен

знать:

- основные виды компьютерных продуктов в организационно-экономической сфере;
- компоненты архитектуры современного персонального компьютера, их основные характеристики и функциональные возможности;
- этапы, основные процессы и подходы к разработке программных продуктов;

уметь:

- проводить анализ и выбирать компоненты аппаратного обеспечения для решения профессиональных задач;
- применять программные средства для решения экономических задач;
- использовать современные технологии разработки приложений Visual Basic for Application для автоматизации работы с офисными программами;

иметь представление:

- об истории и перспективах развития рынка компьютерных продуктов;
- об основных понятиях объектно-ориентированного подхода к созданию программных продуктов;
- о характеристиках основных программных продуктов бухгалтерского учета;

2. Связь с другими дисциплинами

Дисциплина «Компьютерные продукты» продолжает линейку дисциплин по компьютерным наукам — «Информатика» и «Компьютерные технологии». Назначение упомянутых курсов — получение общих, универсальных знаний, навыков и умений работы на компь-

ютере. Цель дисциплины «Компьютерные продукты» — подготовка специалистов, которые могут использовать компьютерные системы для решения профессиональных задач, участвовать в проектах, связанных с выбором аппаратно-технического обеспечения, разработкой программных продуктов и оказания информационных услуг.

Данный курс так же создает необходимую базу для последующего изучения дисциплины «Информационные системы», рассматривая такие общие понятия, как «проектирование программного обеспечения», «жизненный цикл программного продукта» и некоторые другие.

3. Программа дисциплины

Содержание лекций

Введение. История появления персонального компьютера. Понятие компьютерного продукта.

Раздел 1. Аппаратное обеспечение

Т.1. Классификация персональных компьютеров. Спецификация РС99.

Т.2. Блоки обработки информации. Системная плата. Разъемы. Процессоры — история и характеристики. Оперативная память. Модули ОЗУ. Кэш память.

Т.3. Блоки хранения информации. Винчестер — происхождение, состав, характеристики, интерфейсы. Съёмные накопители — обзор, подключение.

Т.4. Подключение внешних устройств. Технология PnP. ISA и PCI. Порты, шины и слоты.

Раздел 2. Программное обеспечение

Т.5. Классификация и определение программных продуктов. Понятия «программное обеспечение» и «программный продукт». Классификация программного обеспечения.

Т.6. Стратегия разработки ПО. Программная инженерия. Жизненный цикл программного обеспечения. Процессы и модели ЖЦ. CASE-средства разработки приложений.

Т.7. Технология быстрой разработки приложений — RAD. Описание технологии. Применение к офисному программированию.

Т.8. Элементы языка VBA. Понятие, типы и описание переменной. Константы и операторы. Структурные операторы — ветвление и циклы.

Т.9. Практика использования VBA — объекты MS Excel. Основные понятия теории объектно-ориентированного программирования — объект, класс, метод, свойство. Избранные свойства MS Excel.

Т.10. Программное обеспечение для решения финансово-экономических задач. Обзор программных продуктов финансово-экономического назначения на Российском рынке: функциональные возможности, производители.

Содержание практических занятий:

— разбор вопросов (заданий) для самопроверки по темам лекций с разработкой примеров;

— выполнение практического задания «Поиск оптимальной ставки налога. Использование VBA».

Вопросы для самостоятельной работы:

— ознакомление с отечественным рынком программных продуктов для решения финансово-хозяйственных, аналитических задач и задач обучения в экономической сфере.

4. График изучения дисциплины и форма контроля

Дисциплина изучается в течение семестра.

Промежуточный контроль — по результатам выполнения практических заданий.

Форма итогового контроля: зачет в конце семестра по вопросам теста.

Требования к студенту и система оценки

В течение семестра студенту необходимо выполнить три практических задания (см. «Темы и рекомендации по выполнению практических заданий»). Каждое задание оценивается по двухбалльной шкале «зачтено — незачтено». Успешное выполнение всех заданий является условием допуска к итоговому зачету по дисциплине.

Для сдачи зачета по дисциплине от студента требуется:

а) выполнить и сдать практические задания;

б) правильно ответить на 80% вопросов теста для итогового контроля.

5. Рекомендации по изучению дисциплины

Дисциплину следует изучать в том порядке тем, которые приведены в программе дисциплины.

Для изучения дисциплины рекомендуется следующая последовательность.

1) Ознакомиться с названием и кратким содержанием темы по программе дисциплины.

2) Внимательно прочитать лекцию к выбранной теме.

3) Ответить на вопросы для самопроверки. При ответе на вопросы контролировать себя, насколько полно и правильно удастся ответить на вопрос без использования конспекта лекций.

4) Ознакомиться с основной литературой и источниками Internet, которые рекомендуются к данной теме.

5) Выполнить практические задания.

7) После изучения всех тем обратиться к вопросам теста для итогового контроля. Тест составлен таким образом, что ответы на его вопросы имеются в содержании соответствующей лекции. Среди перечисленных ответов к вопросам найти правильные с помощью конспекта лекций, глоссария.

8) Поддерживать уровень знаний по изученным темам. Для этого — использовать краткое содержание тем в программе дисциплины, а также глоссарий.

При возникновении вопросов по терминологии, при необходимости «быстро» вспомнить материал, а также для запоминания определений — использовать глоссарий дисциплины.

9) Для подготовки к итоговому контролю после изучения всех тем — проверить себя, ответив на вопросы теста для итогового контроля. При этом необходимо ответить на все вопросы теста без обращения к конспекту лекций, глоссарию или другому материалу.

Для получения более обширного материала по дисциплине можно воспользоваться перечнем дополнительной литературы.

КОНСПЕКТ ЛЕКЦИЙ

ВВЕДЕНИЕ

Развитие науки и техники XX в. ознаменовалось многими выдающимися достижениями: массовой автомобилизацией, развитием авиации, радио, телевидения, использованием энергии атома, полетом человека в космос, появлением Интернета и многим другим. Среди этих событий появление и последующее стремительное развитие компьютерной индустрии в последние десятилетия 20 столетия занимает весьма заметное, если не первое, место.

Начало было положено в августе 1981 г., когда фирма IBM представила конструкцию персонального компьютера (ПК), названную IBM PC XT (PC — это аббревиатура от английского Personal Computer). Появилась новая отрасль промышленности, темпы развития которой оказались невиданными: если телевизор морально устаревает лет за 10, то для компьютера этот срок составляет полтора-два года. Объем производства в этой отрасли очень быстро стал составлять в ценовом выражении десятки миллиардов долларов. Выяснилось, что появление ПК очень многое меняет в таких областях, как образование и наука. Использование ПК существенно повышает производительность труда инженеров, бухгалтеров, экономистов. Компьютер быстро внедрился в офисы и параллельно стал домашним компьютером, образовав весьма мощный сектор рынка SOHO (Small Office/Home Office).

В 70-х гг. идея персональной ЭВМ — машины, принадлежащей одному человеку, а не организации — витала в воздухе. Возможно, главным побудительным мотивом к этому была та парадоксальная смесь восхищения и досады, которую вызывали у людей большие вычислительные машины первых поколений. Эти электронные гиганты могли творить чудеса, но из-за своей огромной стоимости и технической сложности работали лишь в строго определенных условиях и были доступны лишь немногим избранным.

Взаимодействие с ЭВМ строилось через посредников — операторов. Положение не изменилось и после появления и широкого распространения мини-ЭВМ, которые нельзя отнести к персональным компьютерам из-за их стоимости, — они были по карману только очень богатым персонам.

Американцы считают первым персональным компьютером машину под названием «Альтаир-8800». Ее разработала в 1974 г. небольшая фирма Micro Instrumentation and Telemetry Systems (MITS), расположенная в городе Альбукерке, которую возглавлял отставной офицер ВВС США, энтузиаст электроники Эдвард Роберте. По современным понятиям это была довольно примитивная машина. Собранная на процессоре Intel 8080, с объемом памяти 256 байт, она не имела ни клавиатуры, ни экрана. Данные вводились в двоичном коде при помощи выключателей, результаты (в двоичном коде) считывались по мигающим лампочкам. Несмотря на это, машина «Альтаир-8800» имела ошеломляющий коммерческий успех, — настолько людям хотелось иметь собственную вычислительную машину.

В 1977 г. фирма Apple Computers начала массовое производство ПК Apple-2. Конструкция этого компьютера, построенного на основе процессора фирмы Motorola, оказалась весьма удачной. В том же году выпуск ПК наладили фирмы Commodore и TRS (Tandy Radio Shack).

Что касается Европы, то можно вспомнить ПК ZX-Spectrum на процессоре Z-80, появившийся в 1981 г. Эта конструкция быстро завоевала всю Западную Европу. Считается, что для ZX-Spectrum написано более сотни тысяч программ. Автор конструкции британец Клайв Синклер, основатель и владелец Sinclair Research Ltd, получил дворянское звание за вклад в компьютеризацию Соединенного Королевства. ZX-Spectrum получил второе дыхание на рубеже 80-х и 90-х гг. В эти годы к процессу компьютеризации подключилась Восточная Европа, включая Россию. Было разработано огромное количество разных вариантов Sinclair-совместимых ПК на местной элементной базе. Через эту машину многие тысячи молодых людей приобщились к вычислительной технике.

В 1981 г. случилось воистину эпохальное событие — на рынок ПК вышла фирма IBM. Рассеялись последние сомнения в том, что персональные компьютеры вошли в жизнь общества всерьез и на-

долго,— всем было известно, что эта фирма не занимается пустяками и все делает основательно. Выход на арену IBM означал конец «самодельности», неформального, любительского стиля работы, характерного для первого этапа развития индустрии ПК. Предложенная фирмой конструкция персонального компьютера IBM PC XT имела ряд черт, определивших ее ошеломляющий успех.

Прежде всего — это применение нового, перспективного и весьма дорогого по тем временам процессора Intel 8088. Фирма IBM не пошла на то, чтобы использовать популярные на тот момент восьмидесятибитные процессоры (таков процессор Z-80), которые были уже близки к своему закату. Процессор Intel 8088 имел внутренние регистры, рассчитанные на 16 разрядов двоичного числа (это уже шестнадцатидесятибитная архитектура).

В-вторых — это открытая архитектура компьютера IBM PC. Под открытой архитектурой понимается возможность расширять конструкцию компьютера путем подключения дополнительных блоков (карт расширения) через специальное устройство подключения, называемое шиной расширения. В IBM PC использовалась очень удачная шина расширения ISA (Industrial Standard Architecture), которая «жива» до сих пор (удивительное долголетие для компьютерного мира). Сразу же появилось немало фирм, производящих карты расширения. Руководители этих фирм были заинтересованы в сохранении и развитии именно этой архитектуры.

В-третьих — это то обстоятельство, что IBM никак не закрепила свои авторские права на разработанную архитектуру. Были запатентованы только коды BIOS (Basic Input Output System, по-русски ее называют «базовая система ввода-вывода»). С этой программы начинается запуск ПК. Поэтому у других фирм появилась возможность подключиться к производству ПК с такой архитектурой, требовалось лишь написать свои коды BIOS. Так появилось понятие IBM-совместимый ПК.

Таким образом, к настоящему времени сложилась развитая индустрия производства компьютерных продуктов для профессионального использования в различных сферах общественной жизни, в том числе, в финансово-хозяйственной деятельности. Под **компьютерными продуктами** понимаются аппаратные и программные средства компьютера как объекты производства и коммерческой деятельности.

Так, например, по результатам опроса журнала «Мир ПК» в 2000-м г. лучшими компьютерными продуктами стали персональные компьютеры фирмы Compaq и бухгалтерская программа «1С:Бухгалтерия 7.7», 1-ГГц процессор Athlon от фирмы AMD и пакет офисных программ MS Office 2000, мониторы PF775 фирмы ViewSonic и система оптического распознавания символов FineReader 5.0.

В настоящем курсе лекций последовательно рассматриваются оба типа компьютерных продуктов — и аппаратное и программное обеспечение (ПО). В первом разделе рассмотрены классификация и основные компоненты архитектуры персонального компьютера, приведены их основные характеристики. Во втором разделе рассматриваются программные продукты. Даны возможные классификации программного обеспечения, описаны современные подходы к проектированию и разработке ПО. В рамках повествования о технологиях разработки будут изучены методы применения инструментальной среды программирования для создания приложений на примере Visual Basic for Applications. Также приведены оценки некоторых экономических программных продуктов, используемых для автоматизации бухгалтерского учета на различных аппаратных платформах. Для обеспечения усвоения материала ссылки на используемую литературу по тексту не приводятся. Используемые при составлении источники приведены в списке литературы.

Вопросы для самопроверки

1. Какой компьютер американцы называют первым ПК?
2. В каком году была представлена конструкция IBM PC XT?
3. Какие черты конструкции определили ее успех?
4. Какой процессор применялся в ПК этой конструкции?
5. Объясните термин «открытая архитектура компьютера IBM PC»?
6. Как перевести аббревиатуру BIOS?
7. Что понимается под компьютерными продуктами?
8. Расшифруйте аббревиатуру ПО.

ЧАСТЬ I. АППАРАТНОЕ ОБЕСПЕЧЕНИЕ ПК

1. Классификация персональных компьютеров

В этой главе выделим из всего разнообразия **персональных компьютеров** группы компьютерной аппаратуры, отличающейся

по своим функциональным возможностям. Ведь, как правило, и производители, и сборщики компьютеров предлагают не одну, а целую «линейку» марок машин, отвечающих разным запросам пользователей.

Существует несколько возможных способов классифицировать существующие модели — например, критерием может служить архитектура микропроцессора. Если говорить о классификации компьютеров вообще, то можно вспомнить об известном разделении на мобильные компьютеры, персональные компьютеры, а также многопроцессорные рабочие станции и серверы. Но в этом курсе подробно рассмотрим спецификацию РСхх, разработанную компаниями Intel и Microsoft для ПК, и предназначенную для аппаратуры, работающей под управлением операционной системы (ОС) Windows. Инициаторы внедрения РСхх декларировали цель — придать эволюции компьютерной аппаратуры, управляемой ОС Windows, планируемый и целенаправленный характер в направлении повышения надежности, совместимости и удобства. В РСхх персональные компьютеры разделяются на определенные группы, и по каждой группе подробно и методично описываются архитектура, наборы устройств и требования к ним, функции BIOS, внешние порты и прочее, и прочее. Устанавливается даже тип корпуса — вскрываемый или наглухо закрытый.

В РСхх содержатся требования к составу аппаратуры определенного временного этапа (метка «хх» в названии спецификации заменяется цифрами года принятия стандарта). Известны спецификации РС97, РС98. Сейчас в ходу — РС99. Эта спецификация — довольно объемный 512-страничный документ под названием РС99 System Design Guide. С нею можно ознакомиться в Интернете на Web-сайте фирмы Intel. Спецификация РС99 все персональные компьютеры разделяет на следующие группы:

— *Consumer PC* (пользовательский компьютер). Предназначен для работы вне локальной сети, но приспособлен для подключения к Интернету (комплектуется модемом). Такой компьютер при установке соответствующего программного обеспечения может использоваться как обучающий и как игровой. Также он подходит для поддержания малого (домашнего) офиса (вписывается в аппаратный сектор, называемый SOHO — Small Office/Home Office). Содержит графическую подсистему, поддерживающую солидные мультимедийные приложения.

тимедиа-приложения. Рекомендуется наличие Device Bay — устройства, позволяющего заменять периферию без вскрытия корпуса и перезагрузки компьютера.

— *Office PC* (офисный компьютер). Это сетевой компьютер (снабжается сетевым адаптером для включения в локальную сеть). Для него устанавливается минимальная общая цена владения, то есть минимизируется ТСО (Total Cost of Ownership — общая цена владения).

— *Workstation PC* (рабочая станция). Компьютеры данного типа также приспособлены для работы в сети, но превосходят по своим возможностям Office PC. Они предназначены для решения задач, требующих интенсивных вычислений (финансовые приложения, разработка программного обеспечения и т. п.).

— *Entertainment PC* (игровые или развлекательные компьютеры). Требования к наполнению этих компьютеров подчинены их назначению. В них устанавливают устройства высокопроизводительной графики, аудиосистемы Hi-Fi, сравнимые по качеству с хорошим музыкальным центром. Обеспечивается расширенная поддержка подключения внешних устройств (видеомагнитофонов, цифровых фотокамер и прочее).

— *Mobile PC* (мобильный компьютер). Здесь, в соответствии со спецификой компьютера, основные направления совершенствования — снижение веса и увеличение продолжительности работы от батарей. Набор функциональных возможностей сопоставим с пользовательским компьютером.

По каждой названной группе в PC99 подробно описаны требования к набору устройств компьютера и выполняемым функциям. Спецификация PC99 разделяет требования на обязательные (устройство или функция должны обязательно присутствовать и быть строго определенными), желательные (желательно присутствие и в определенном виде) и прочие (отвечающие требованиям спецификации, но не обязательные). Производители аппаратуры, сумевшие продемонстрировать, что их компьютер вписывается в рамки спецификации PC99, получают право на размещение на своих изделиях логотипа Windows. Сертификат на соответствие аппаратуры требованиям PC99 выдает тестовый центр Microsoft, называемый WHQL (Windows Hardware Quality Labs). Также — WHQL — называют и сам сертификат.

Вопросы для самопроверки

9. Какие фирмы являются разработчиками спецификации РСхх?
10. Для какой аппаратуры она была создана?
11. Где можно ознакомиться с содержанием этой спецификации?
12. Перечислите группы устройств, описанные в РС99.
13. Что общего между группами *Consumer PC* (пользовательский компьютер) и *Entertainment PC* (игровые или развлекательные компьютеры)?
14. Какая фирма выдает сертификат соответствия аппаратуры требованиям РС99?

2. Блоки обработки информации.

В этой главе рассмотрим блоки ПК, занимающие центральное место в его архитектуре и непосредственно используемые в обработке информации,— системную плату и процессор. Именно от характеристик этих устройств зависят и функциональные возможности и быстродействие ПК.

Системная плата

Компьютер подобен конструктору, позволяющему из разных устройств, выполняющих определенные функции, собрать вычислительную машину. Объединяющим элементом этого конструктора, на котором все собирается, является системная плата (рис. 1). Ее называют также главной платой (*mainboard*), материнская плата (*motherboard*), а в документации для обозначения *mainboard* используют аббревиатуру МБ. На системной плате обычно размещаются: процессор; оперативная память; сверхоперативное запоминающее устройство, называемое также кэш-памятью; микросхема системной BIOS с данными об аппаратных настройках и аккумулятор для ее питания; набор управляющих микросхем, вспомогательные микросхемы, контроллеры ввода/вывода или чипсет (*chipset* — набор микросхем); разъемы расширения или слоты (*slot*); разъемы для подключения интерфейсных кабелей жестких дисков, дисководов, последовательного и параллельного портов, универсальной последовательной шины USB; разъемы питания; разъем для подключения клавиатуры и ряд других компонентов. Электрические характеристики системной платы определяются набором управля-

ющих микросхем на плате (чипсетом — от английского chipset), а ее конструктивные особенности, габариты, совместимость с определенным корпусом определяются форм-фактором (например, ATX — применяется на большинстве современных компьютерах).

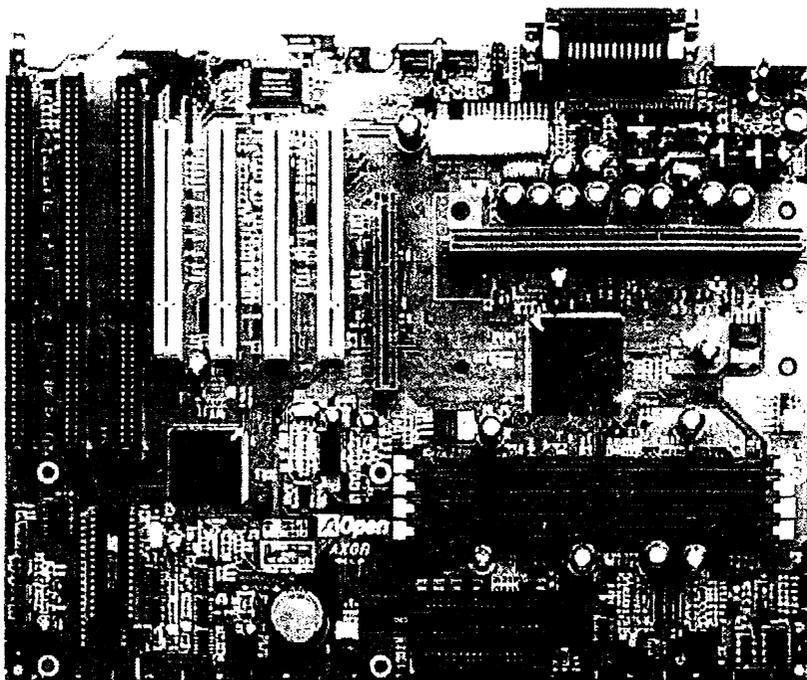


Рис. 1. Внешний вид системной платы форм-фактора ATX

Главное место на любой системной плате занимает процессорный разъем. Сложилось так, что контактные разъемы процессоров имеют разные конструкции, и появляются все новые и новые. Времена единого процессорного гнезда (таким одно время был Socket 7) ушли в прошлое. Конкурентная борьба, а также маркетинговая политика компьютерных гигантов приводят к тому, что уже сейчас практически каждая новая серия процессоров имеет свой, отличный от предшественников и тем более от конкурентов, тип разъема.

Имеются две основные разновидности процессорных разъемов: щелевой разъем (слот, slot) и прямоугольная панель с гнездами (сокет, socket). Разработаны и переходные разъемы, позволяющие

вставить в системную плату процессор с «неподходящим» разъемом. Последнее время отмечается тенденция к отходу от щелевых разъемов, причем она просматривается для процессоров, выпускаемых и Intel, и AMD. Это объяснимо, поскольку появление разъемов Slot 1 и Slot A было вызвано несовершенством технологического процесса изготовления процессоров Pentium II и Athlon. Технологию наладили и вернулись к панельному разъему типа Socket.

Таблица 1.

Разъемы и процессоры

Разъем	Число выводов	Типы процессоров
Slot I (SECC)	242	Pentium II/III/Pro, Celeron
Slot 2 (SECC2)	330	Pentium III, Xeon
Slot A	242	Athlon
Socket 7	321	Pentium, K5, K6, 6x86, C6
Socket 370 PPGA	370	Celeron, Cyrix III
Socket 370 FC-PGA	370	Pentium III, Celeron
Socket A	462	Athlon, Duron
Socket 423 FC-PGA	423	Pentium 4 (Willamette-423)
Socket 479 micro FC-PGA2	479	Prescott (Willamette-479)
Socket 370 FC-PGA2	370	Tualtin (развитие Coppermine)

Процессор

Наиболее значимым элементом ПК является центральное процессорное устройство (ЦПУ, CPU), или просто процессор. Его основными характеристиками являются:

— *Степень интеграции чипа* — сколько транзисторов может в нем уместиться (например, для Intel 80586 — это от 3 млн. транзисторов на 3,5 см²)

— *Внутренняя разрядность данных* — количество бит, которое может одновременно обрабатываться внутри ЦПУ.

— *Тактовая частота* — частота внутреннего тактового генератора.

— *Адресация памяти* — размер поддерживаемой памяти (1МБ для CPU 8086, 4ГБ для 80486).

В первых ПК IBM PC класса XT использовались процессоры Intel 8088 (в документации пишут — i8088).

Процессоры Intel 8088 относят к процессорам первого поколения. Внутренние регистры этих процессоров шестнадцатиряд-

ные. Данные передаются и принимаются по восьмиразрядной шине данных. Шина адреса содержит 20 разрядов, то есть процессор может использовать 2^{20} байт памяти (1 Мбайт). Это обстоятельство наложило свой отпечаток на всю архитектуру IBM PC-совместимых ПК — первый мегабайт памяти таких компьютеров (вплоть до Pentium III) имеет особое значение и устроен одинаковым образом.

За период с 1981 г. по 1995 г. сменилось четыре поколения процессоров: на смену i8088 пришел i80286, затем i80386 (первый процессор с 32-разрядными внутренними регистрами), и, наконец, процессор i80486, или, как его называют проще, — 486-й. Всю линию этих процессоров стали называть линией i80x86. (процессор i8088 был лишь упрощенным, более дешевым вариантом «праотца» Intel-процессоров — i8086, давшего название линии). Производительность процессоров удваивалась примерно каждые два года. Это связано с совершенствованием их внутренней архитектуры и с новыми технологическими решениями, позволяющими увеличивать параметр, задающий «темп» вычислений, — тактовую частоту. У процессоров i8088 она равнялась 4,77 МГц, а у 486DX4 внутренняя тактовая частота уже перевалила за 100 МГц.

Первые процессоры пятого поколения (80586) появились в марте 1993-го. Тогда Intel, чтобы не повторять ошибки с i80486 (суд отклонил иск Intel к AMD по поводу названия), решила дать своему детищу собственное имя — Pentium, которое впоследствии стало нарицательным.

В апреле 1998 г. фирма Intel выпустила Celeron — революционный (в смысле ориентации на широкого пользователя) процессор (Celeron произносится как «силерон», созвучно английскому слову celerity — скорость). Это целое семейство недорогих процессоров, изготавливаемых с кэшем второго уровня или без него. Имеются варианты моделей с разными контактными разъемами: под Socket 370 и под Slot 1.

Последняя разработка линейки Pentium — Willamete, или Pentium 4 — принципиально новый процессор Intel, рассчитанный на использование в обычных ПК. Кэш L2 (1 Мбайт) располагается на чипе процессора. Предпринят ряд шагов, направленных на увеличение производительности: добавлены исполнительные модули, декодеры, увеличен объем буферов и т. д. Превосходит по производительности процессоры Pentium III на 30-50 %. Производится по

технологии с шагом 0,18 микрон, тактовая частота — от 1 ГГц и выше. Формат разъема крепления — Socket 462.

Основным соперником Intel на рынке процессоров является компания AMD. K7-Athlon (рис. 2) — первый процессор от AMD, получивший собственное имя. Создавая его, разработчики AMD отказались от прямого копирования архитектурных принципов Intel и предложили рынку свой вариант платформ для IBM PC-совместимых компьютеров. В марте 2000 г. этим процессором был преодолен барьер тактовой частоты 1000 МГц (чуть раньше, чем это сделал Pentium III). Процессор Athlon имеет большой объем кэш-памяти первого уровня — 128 Кбайт (по 64 Кбайт для инструкций и данных). Объем кэша L2 — от 512 Кбайт до 8 Мбайт. Поддерживает набор инструкций MMX, а также расширенный набор инструкций 3DNow! Контактный разъем щелевого типа — Slot A. В зависимости от объема кэша L2 процессоры Athlon имеют дополнительную приставку в названии, например, — Athlon Value, Athlon Ultra и Athlon Select.

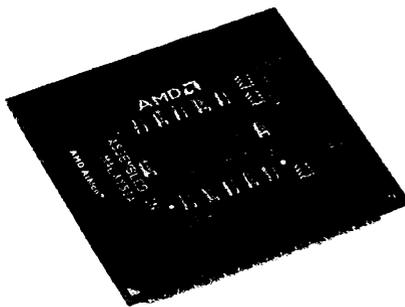


Рис. 2. Процессор K7-Athlon

Также можно упомянуть еще один продукт этой компании — K7-Duron. Выпуск этого процессора рассматривают как реакцию фирмы AMD на появление процессора Celeron. Основные козыри Duron — большой объем кэша L2 (192 Кбайт), расположенного на кристалле процессора. Частота FSB — до 200 МГц. Работает в материнских платах, оборудованных специальным 462-контактным процессорным разъемом Socket A.

По тепловыделению Duron несколько превосходит процессор Celeron, поэтому его нужно лучше охлаждать (Duron 650 выделяет примерно столько же тепла, сколько и процессор Pentium III 733).

Оперативная память

На производительность ПК сильно влияет объем памяти оперативного запоминающего устройства (сокращенно — ОЗУ). Английский термин для ОЗУ — RAM (Random Access Memory — память

с произвольным доступом). На системных платах первых IBM PC устанавливались отдельные микросхемы памяти. Сейчас микросхемы ОЗУ размещают блоками на специальных модулях памяти — небольших платах с многочисленными контактами, рассчитанными на установку в щелевидные разъемы (слоты) системной платы (рис. 3). Модули ОЗУ бывают разных типов:

— *SIMM (Single inline memory module)* — модуль памяти с односторонними контактами;

— *DIMM (Dual inline memory module)* — модуль памяти с двухсторонними контактами;

— *RIMM (Rambus inline memory module)* — 168-контактный модуль памяти RDRAM повышенной (1,6 Гбайт/с) пропускной способности (используя 16-битную шину, эта память передает информацию по обоим фронтам тактового импульса, то есть фактически удваивает частоту работы);

— *DDR RAM* — модуль повышенной (до 2,1 Гбайт/с) пропускной способности, имеющий 184-контактный разъем типа DIMM.



Рис. 3. Внешний вид модуля ОЗУ

Быстродействие ОЗУ увеличивать сложно, поэтому в современном ПК используют так называемую кэш-память. Ее еще называют просто кэш. Слово созвучно названию cache (запас), произносимому по-английски. Кэш располагается или на системной плате, или на самом процессоре и реализует быстродействующую буферную память между процессором и основной памятью. Кэш служит для частичной компенсации разницы в быстродействии процессора и ОЗУ, — в кэш-память попадают наиболее часто используемые данные. Когда процессор первый раз обращается к ячейке памяти, ее содержимое параллельно копируется в кэш. В случае близкого по времени повторного обращения к адресу той же ячейки это содержимое может быть с большой скоростью выбрано из кэш-памяти. При записи в память данные попадают в кэш и либо одновременно копируются в память (схема Write Through — прямая, или сквозная, запись), либо копируются через некоторое время (схема Write Back — отложенная, или обратная, запись). При обратной записи

данные копируются в память, если запись в кэш невозможна — отсутствует свободная область. И тогда в память вытесняется наименее используемая область кэш-памяти.

В процессоры линии 486 и выше встроен также внутренний (internal) кэш объемом 8-16 Кбайт. Его именуют Primary (первичный), или кэш L1 (от Level 1 — первый уровень). Есть еще внешний (external) кэш, расположенный на плате. Его именуют Secondary (вторичный), или кэш L2 (от Level 2 — второй уровень). В большинстве процессоров внутренний кэш работает по схеме с прямой записью, а в процессорах Pentium он может работать и с отложенной записью. У процессоров Pentium Pro кэш L2 (объемом 256 Кбайт или 512 Кбайт) находился на одном кристалле с процессором. Это приводило к большим трудностям при производстве и повышало стоимость Pentium Pro.

В Pentium II и Pentium III (а также в последних моделях процессоров Celeron) реализовано компромиссное решение: процессор и кэш расположены на разных кристаллах, которые компактно смонтированы на одной плате. Плата процессора с размещенными на ней обоими кристаллами вставляется в разъем системной платы.

Вопросы для самопроверки

15. Перечислите возможные названия системной платы ПК.
16. Какие элементы ПК размещаются на ней?
17. Что такое чипсет?
18. Что такое слот?
19. Какие параметры системной платы определяются ее форм-фактором?
20. Назовите виды процессорных разъемов, чем они отличаются?
21. Перечислите альтернативные названия процессора.
22. Какие основные характеристики процессоров вы знаете?
23. Как назывались процессоры первых ПК IBM PC XT?
24. Почему процессор Intel Pentium не стали называть просто 586-м?
25. Какая фирма выпускает процессоры Celeron?
26. Как еще называют процессор Pentium-IV?
27. Какая фирма является основным соперником Intel?
28. Назовите процессоры ее производства.

29. Назовите английский аналог русской аббревиатуры ОЗУ
30. Перечислите разновидности модулей ОЗУ
31. Где расположен и для чего используется кэш?

3. Блоки хранения информации

Рассмотрим основные устройства для записи, хранения и переноса информации и оценим их характеристики.

Винчестер

Винчестер — накопитель на жестких дисках (Hard Disk Drive — HDD) применяется для хранения и использования больших объемов информации. Само название «винчестер» появилось в 1973 г., когда фирма IBM разработала ЭВМ 3340. Ею тогда был выпущен пакет из двух дисков по 30 Мбайт с маркировкой «30/30», что напоминало пользователям



Рис. 4. Жесткий диск со снятой верхней крышкой калибр двустволки «Винчестер 30/30». Типичный жесткий диск состоит из гермоблока и платы электроники (рис. 4). В гермоблоке установлен шпиндель с одним или несколькими дисками. Диски изготовлены из алюминия (иногда из керамики или стекла) и покрыты тонким слоем окиси хрома. Сбоку шпинделя расположен поворотный позиционер со считывающими головками. Двигатель расположен под дисками. Скорость вращения дисков постоянна (обычно — 4500, 5400, 7200 или 10000 оборотов в минуту) и является одной из основных характеристик. Еще одной характеристикой винчестера является поддерживаемый им интерфейс, от которого

зависит способ подключения устройства и скорость обмена данными. Развитие интерфейсов винчестеров шло двумя параллельными путями: дешевым и дорогим.

Дорогое решение заключалось в создании на плате самого винчестера отдельного интеллектуального контроллера, который бы брал на себя значительную часть работы по взаимодействию с винчестером. Результатом этого подхода явился интерфейс SCSI (Small Computers System Interface — системный интерфейс малых компьютеров, читается «скази»), который быстро завоевал популярность на рынке серверов. Одним из преимуществ этого подхода являлась возможность подключения к компьютеру значительного для того времени количества устройств, требующих для своей работы широкого канала передачи данных.

Простое и дешевое решение — переложить значительную часть операций по вводу-выводу на центральный процессор. У этого решения вполне очевидный недостаток: снижение общей вычислительной мощности системы, особенно заметное при многозадачной работе. А в те времена, когда процессоры не были такими мощными, это сильно ограничивало возможности, в частности, файловых серверов. Результатом воплощения в жизнь этого подхода явился широко распространенный интерфейс IDE.

Этот интерфейс был сравнительно дешев и, хотя не был самым производительным, полностью вытеснил другие интерфейсы с рынка дешевых и недорогих систем. Он постепенно развивался, и со временем появились стандарты UDMA, существенно ускоряющие работу винчестеров, интерфейсы IDE стали более интеллектуальными. А так как производительность процессоров росла быстрее производительности винчестеров, то ограничения интерфейса IDE играли все меньшую роль.

Тем самым на сегодня мы имеем два типа винчестеров: высокопроизводительные SCSI и «ширпотреб» — IDE. Принципиальных различий в устройстве самих винчестеров SCSI и IDE нет, но исторически сложилось, что интерфейс SCSI рассчитан на сегмент дорогих серверных решений, поэтому, в среднем, винчестера с этим интерфейсом быстрее и, как следствие, существенно дороже.

Пропускная скорость SCSI значительно выше IDE, целых 160 Мб/с. А IDE работает со скоростью 33, 66 и 100 Мб/с. Соответствующие стандарты называются ATA/33, ATA/66 и ATA/100.

У стандартного ПК имеется два канала IDE, к каждому из которых можно подключить по два HDD. К SCSI-интерфейсу (имеется в виду наиболее распространенный — SCSI-2) можно подключить до семи устройств.

Говоря об интерфейсах для подключения винчестеров, стоит вспомнить и о переносных винчестерах. В настоящее время существует несколько решений для подключения внешних устройств. Во-первых, есть винчестеры, подключающиеся к USB-порту. Они используются в основном для обмена данными с цифровыми камерами и прочими мобильными устройствами. В силу невысокой пропускной способности этой шины подобные диски, конечно, не смогут сравниться в производительности с внутренними устройствами.

Все большее распространение получает новый интерфейс IEEE1394, который может использоваться не только для подключения жестких дисков, но и других устройств, работающих с большими массивами данных, например, видеокамер. Контроллеры этого интерфейса иногда даже встраиваются в материнские платы. Его производительности хватает, например, для проигрывания видео высокого качества — заявленная пропускная способность интерфейса достигает 50 Мб/с. Напомним, что еще пару лет назад такой скоростью не мог похвастаться интерфейс IDE. Основными задачами производителей всегда было увеличение объема хранящейся на дисках информации и скорости работы с этой информацией. Как увеличить **объем диска**? Наиболее очевидным решением является увеличение количества пластин в корпусе жесткого диска. Подобным образом обычно различаются модели в пределах одного модельного ряда. Этот способ является наиболее простым и позволяет на одной и той же элементной базе получать диски различной емкости.

Но у этого способа существуют естественные ограничения: количество дисков не может быть бесконечным. Увеличивается нагрузка на мотор, ухудшаются температурные и шумовые характеристики диска, вероятность брака растет пропорционально количеству пластин, а значит, труднее обеспечить надежность. Среди промышленно производимых дисков наибольшим количеством пластин обладает SCSI диск Seagate Barracuda 180 — у этого винчестера 12 пластин! Есть и рекордсмены в области упрощения устройства дисков — это, например, рассмотренный нами далее Maxtor 513DX и 541DX, у которого один диск, используемый только с одной стороны.

Таблица 2.

**Некоторые технические характеристики некоторых
IDE-винчестеров форм-фактора 3,5 дюйма со скоростью
вращения 5400 об/ мин и 40-гигабайтными пластинами**

Серия, Модель	Емкость, Гбайт	Число пластин/головок	Время поиска, мс Мин./среднее/макс. (Типовые)	Уровень шума, дБ (типовой) вращение/поиск	Потребляемая мощность, Вт вращение/поиск/чтение/ раскрутка
Maxtor DiamondMax D540X-K (Galaxy)					
4K080H4	80	2/4		30/31	3,75 /
4K060H3	60	2/3	чтение:	30/31	4,8 /
4K040H2	40	1/2	2/12/24	30/31	5,2 /
4K020H1	20	1/1		30/31	23,7
Seagate U Series 6					
ST380020A	80	2/4	чтение:	30/35	5 /
ST360020A	60	2/3	1,2/8,9/22	30/35	7,5 /
ST340810A	40	1/2	запись:	29/34	7,5 /
ST320410A	20	1/1	2/10,9/24	29/34	34
Western Digital Performer					
WD1200AW	120	3/6		28/30	
WD800AW	80	2/4	чтение:	28/30	—
WD600AW	60	2/3	2/13,9/32	28/30	
WD400AW	40	1/2		28/30	
Samsung SpinPoint V 40					
SV 8004H	80	2/4		29/30	4,8 /
SV 6003H	60	2/3	чтение:	29/30	6,0 /
SV 4002H	40	1/2	0,8/8,9/16,9	29/30	4,5 /
SV 2001H	20	1/1		29/30	28

Технологически более сложный (и более перспективный) метод увеличения объема — увеличение плотности записи информации. Тут возникает целый ряд технологических проблем. Современные пластины изготавливаются из алюминия или даже из стекла (некоторые модели IBM). Магнитное покрытие имеет сложную многослойную структуру и покрыто сверху специальным защитным слоем. Размеры частиц магнитного покрытия уменьшаются, а чувстви-

ность их возрастает. Помимо улучшения параметров самих пластин, существенным усовершенствованиям должна подвергнуться система считывания информации. Необходимо уменьшить зазор между головкой и поверхностью пластины, повысить чувствительность головки. Но и тут законы физики накладывают свои естественные ограничения на предел применения подобных технологий. Ведь размеры магнитных частиц не могут уменьшаться бесконечно.

Самый простой способ увеличить **скорость считывания** — увеличить скорость вращения пластин. По этому пути и пошли конструкторы. Если пластины вращаются с большей скоростью, то за единицу времени под считывающей головкой проходит больше информации. На увеличение скорости считывания влияет также и рассмотренное выше увеличение плотности записи информации. Именно по этой причине SCSI диски, как правило, обладают большей скоростью вращения. Однако на такой скорости сложнее точно позиционировать головку считывания, поэтому плотность записи там меньше, чем на некоторых IDE дисках, а стоят такие диски больше.

Так как головка при поиске информации перемещается только поперек диска, она вынуждена «ждать», пока диск повернется, и сектор с запрашиваемыми данными окажется доступным для чтения. Это время зависит только от скорости вращения диска и называется **временем ожидания информации (latency)**. Но необходимо понимать, что общее **время доступа к информации** определяется временем поиска нужной дорожки на диске и временем позиционирования внутри этой дорожки. Увеличение скорости вращения диска уменьшает лишь последнее значение. Для уменьшения времени поиска нужной дорожки совершенствуют привод считывающей головки и... уменьшают диаметр пластин диска. Почти все современные винчестеры выпускаются с пластинами диаметром 2,5 дюйма.

Позиционирование головки вообще является отдельной весьма нетривиальной проблемой. Достаточно сказать, что при современной плотности записи приходится учитывать даже тепловое расширение! Таким образом, увеличение скорости вращения диска существенно затрудняет точное позиционирование головки. И в попытках увеличить быстродействие диска иногда приходится жертвовать

объемом, используя пластины с меньшей плотностью записи. Неудивительно, что наиболее дорогие и быстрые винчестеры, отличающиеся более высокой скоростью вращения, не используют максимальной технологически доступной на данный момент плотности записи. За скорость приходится платить.

Так какому диску отдать предпочтение? При одинаковом объеме большего внимания заслуживают модели с большей плотностью записи по сравнению с моделями с большим количеством дисков, хотя бы потому, что у них выше линейная скорость чтения/записи (большие файлы читаются быстрее). Скорость доступа к информации напрямую зависит от скорости вращения пластин (быстрее работа с большим количеством мелких файлов). Но увеличение скорости приводит к удорожанию изделий, а иногда приходится жертвовать и плотностью записи.

Съемные накопители

Спектр устройств, применяемых для переноса информации с компьютера на компьютер, достаточно широк. Помимо обычных 3,5" дискет это и дисковод CD-ROM (в том числе записывающий — CD-R или CD-RW), магнитные накопители ZIP, магнитооптические накопители, ленточные стримеры, флэш-память, мобильные HDD и т. д. Некоторые из перечисленных устройств подключаются к компьютеру как внешние устройства (см. следующую главу). Но пока чаще используются устройства, располагающиеся непосредственно в корпусе системного блока и подключенные к системе через контроллеры.

Накопитель на гибких магнитных дисках (Floppy Disk Drive — FDD) подключается к системе через контроллер FDD (контроллер — устройство управления). В современных ПК этот контроллер находится на системной плате. Через контроллер FDD к компьютеру можно подключать некоторые типы стримеров — накопителей на магнитной ленте. Стандартный ПК может иметь два дисковода FDD (системой они именуется как дисководы А и В). К контроллеру дисководы подключаются с помощью ленточного кабеля.

Съемные накопители могут быть подключены и к каналам IDE. Как уже отмечалось, к каждому каналу IDE можно подключить два устройства. Кроме HDD, это могут быть: дисковод CD-ROM (в том числе записывающий — CD-R или CD-RW), магнитные нако-

пители ZIP, LS-120, магнитооптические накопители. К SCSI-интерфейсу также могут, кроме HDD, подключаться устройства долговременной памяти и дополнительная внешняя аппаратура, то есть аппаратура, находящаяся вне корпуса.

Вопросы для самопроверки

32. Назовите альтернативные названия винчестера.
33. Откуда появилось такое название — «винчестер»?
34. Опишите устройство винчестера.
35. Какие параметры его характеризуют?
36. Какие интерфейсы используются для подключения винчестера?
37. Чем они отличаются?
38. Сколько пластин (дисков) может быть у винчестера?
39. Из чего их изготавливают?
40. Перечислите устройства, относящиеся к съемным накопителям.
41. Куда подключается FDD?

4. Подключение внешних устройств

Здесь необходимо сказать о способах расширения функциональных возможностей ПК за счет подключения дополнительного оборудования. Подключить к компьютеру дополнительное устройство можно через специальную плату расширения, имеющую контактную группу определенного стандарта. Такая плата (ее называют также картой) вставляется в соответствующий разъем (слот) шины расширения системной платы. Тем самым обеспечивается включение размещенного на карте устройства в состав компьютерной аппаратуры. Посредством технологии PnP (короткое название для Plug and Play или Plug&Play — «подключи и играй») операционная система самостоятельно выполнит необходимые настройки, и устройство впишется в ансамбль ПК. Если технология PnP не поддерживается подключаемым устройством, то настройки выполняют вручную.

Наиболее распространены шины расширения ISA и PCI. Соответственно и платы расширения (рис. 5) должны иметь контактные группы, отвечающие приемным гнездам слотов шин ISA или PCI.

В любом случае подобный способ соединения предполагает «вскрытие» системного блока и манипуляции с разъемами систем-

ной платы. У портативных компьютеров слот шины расширения (PC Card) выведен наружу, и подключение к такой шине не требует «вскрытия» компьютера.

Для связи с внешним миром служат также коммуникационные, или последовательные, порты, их обычно два: COM1 и COM2. Они используются для связи двух ПК, подключения внешних модемов, мыши и для некоторых других целей. Внутренний модем (подключаемый через шину расширения) воспринимается системой как еще один COM-порт. В архитектуре IBM PC через BIOS можно управлять четырьмя последовательными портами.

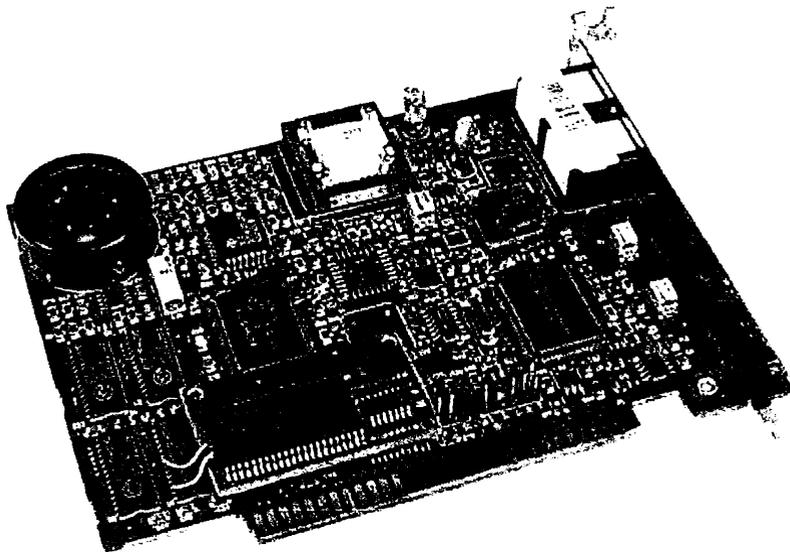


Рис. 5. Пример платы расширения для шины ISA — внутренний модем

Для подключения принтеров служит параллельный порт: LPT. В ПК он обычно один (LPT1), но можно установить еще два: LPT2, LPT3.

В середине девяностых годов у ПК появился новый интерфейс USB (Universal Serial Bus — универсальная последовательная шина). Это весьма быстрый интерфейс, его скорость (12 Мбит/сек) позволяет обеспечить эффективное взаимодействие с лазерным принтером, сканером, цифровой фотокамерой. К шине USB можно под-

ключить до 127 устройств, она допускает «горячее» включение и отключение того или иного устройства, при этом операционная система сама загружает или отгружает соответствующие драйверы.

До недавнего времени шина USB почти не использовалась, поэтому шутники называли ее Unused Serial Bus — неиспользуемая последовательная шина. Однако сейчас положение меняется. Интерфейс USB становится обязательным для любого современного ПК. С другой стороны, многие периферийные компьютерные устройства (сканеры, принтеры, цифровые фотокамеры и даже мониторы) оборудуют разъемами и устройствами, обеспечивающими связь с системным блоком интерфейса USB. Считают, что потенциально USB может в будущем полностью вытеснить последовательные и параллельные порты.

Другая последовательная шина, которая имеет большие перспективы, — это FireWire (буквальный перевод с английского: огненный провод). Официальное название этого устройства — высокопроизводительная последовательная шина IEEE 1394. Эта шина обеспечивает скорость обмена 100 мегабит в секунду и более. Она позволяет связать периферийные устройства не только с ПК, но и между собой. Шина обеспечивает одновременное соединение многих устройств (до 63) без применения дополнительной аппаратуры. Это могут быть цифровые фото- и видеокамеры, видеокамеры для конференций, видеомagniфоны, сканеры. На базе FireWire можно строить небольшие локальные сети. Для оснащения ПК шиной FireWire нужна соответствующая карта расширения.

Если системная плата — интегрированная, то она снабжена контактными разъемами для подключения звуковой аппаратуры и других устройств. Или подобные контакты имеются на соответствующих платах расширения (см. рис. 5).

На рис. 6 показано, как могут выглядеть разъемы, предназначенные для подключения к компьютеру дополнительных устройств. Соединение внешних устройств с компьютерами — та сфера, где особенно весомо ощущается значение стандартизации. Из приведенных выше сведений видно, что способов подключения внешних устройств немало, и немудрено, что их разработчики ожидают от разработчиков компьютеров четкой позиции относительно перспектив развития тех или иных линий интерфейсов подключения внешней аппаратуры. И работа в этом направлении ведется.

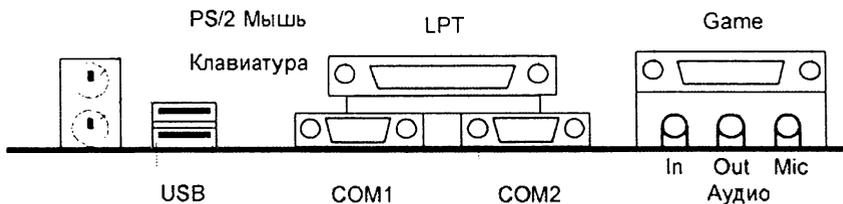


Рис. 6. Так может выглядеть контактная группа для подключения внешних устройств

В спецификации PC99, где сформулированы требования к составу аппаратуры компьютера, большое внимание уделяется освобождению от морально устаревшей шины ISA, а также COM и LPT портов. По этому поводу в PC99 приведены, в частности, следующие рекомендации:

- системы, отвечающие спецификации PC99, должны разрабатываться и поставляться без ISA-карт расширения, а также без ISA-слотов. Принтерам «разрешено» использовать порты COM и LPT, но настоятельно рекомендуется переводить их на интерфейсы USB или IEEE 1394;

- модемы, сканеры и другие устройства ввода/вывода не должны использовать устаревшие шины. Для модемов рекомендуется USB, а для всех остальных — SCSI или IEEE 1394;

- для звуковых устройств рекомендуется PCI или USB;

- устройства видеосистем (видеоадаптеры) не должны выпускаться для установки на шине ISA. Рекомендуемые решения — PCI и AGP (шина подключения видеоадаптеров).

Вопросы для самопроверки

42. Переведите на русский аббревиатуру PnP.
43. Как называют коммуникационные порты?
44. Для чего они используются?
45. Как называется порт принтера?
46. Что такое USB?
47. Расскажите о шине FireWire.

ЧАСТЬ II. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПК

5. Определение и классификация программных продуктов

Понятие «*программное обеспечение*» (*software*) вошло в жизнь с развитием компьютерной индустрии. Без программного обеспечения компьютер — всего лишь электронное устройство, которое не управляется и потому не может приносить пользы. Но это понятие несколько шире, чем комплекс (набор, совокупность) программ. Кроме собственно программ на носителях информации (дискетах или компакт-дисках) оно включает упаковку, эксплуатационную документацию и лицензионное соглашение, когда речь идет о программном продукте, который тиражируется.

Программный продукт — это совокупность отдельных программных средств, их документации, гарантии качества, рекламных материалов, мер по обучению пользователей, распространению и сопровождению готового программного обеспечения.

Основная классификация программного обеспечения, родившаяся еще в 60-е годы, делит его по **сфере применения** на два класса: системное и прикладное. Впервые такое разделение было проведено, видимо, в знаменитом проекте IBM System/360. Согласно ему прикладные программы предназначены для решения определенных задач и получения конкретных результатов, а системные программы — для обеспечения работы прикладных программ. К классу системного ПО традиционно относят операционные системы и разнообразные утилиты по обслуживанию системы. В классе прикладного ПО можно выделить группы офисных пакетов программ, разнообразные программы для обработки графической информации, программы для Интернета т. д. Промежуточное место занимает группа инструментальных средств разработки программ, такие, как например, Visual C++, Delphi и прочие.

В последнее время стали употребляться выражения «вертикальные» и «горизонтальные» приложения. Эти термины заимствованы из модной экономической фразеологии, и в обыденной жизни заменяются неформальными определениями «заказное» и «коробочное» приложение. Их можно считать примером классификации по **источнику финансирования**. В первом случае под-

разумеается приложение, которое сделано по конкретному индивидуальному заказу, а во втором — приложение, рассчитанное на массовый спрос, и которое можно купить в магазине. Разработка горизонтальных приложений финансируется производителем в расчете на последующую продажу, а разработка вертикальных приложений с самого начала работ финансируется заказчиком. Из сказанного видно, что вертикальные приложения обладают одним важным преимуществом — при их разработке можно более полно и точно учесть ожидания пользователей и этого оказывается достаточно для оправдания разработки вертикальных приложений. Однако свои преимущества имеют и тиражируемые коробочные приложения. Они:

- используются многими заказчиками, которые могут обмениваться между собой едиными форматами данных, формами представления информации;

- прошли неоднократную апробацию, проверку другими заказчиками;

- как правило, менее дороги, чем заказные продукты;

- могут быть внедрены в более короткие сроки проверенными технологиями внедрения.

Так же важной характеристикой ПО является его масштаб. **Масштаб** приложения — это качественная оценка размера приложения с точки зрения его использования. Обычно различают следующие градации размеров приложений:

- Настольные (desktop)

- Групповые (groupware)

- Масштаба предприятия (enterprise)

Это разделение чисто качественное, точных количественных характеристик, отличающих приложения разных классов, указать нельзя.

Настольное приложение, как правило, имеет одного пользователя, может выполняться на одном компьютере, и работать автономно, без сети.

Групповое приложение имеет нескольких пользователей, которые могут работать с ним порознь или одновременно, но все принадлежат к одной группе (или категории) пользователей. Как правило, групповые приложения используют локальную сеть или иной способ связи компьютеров.

Приложение *масштаба предприятия* имеет значительное количество пользователей, принадлежащих к различным категориям, выполняет несколько различных, связанных между собой функций и работает в локальной или глобальной сети.

Вопросы для самопроверки

48. Что такое программный продукт?
49. Опишите классификацию программного обеспечения по сфере применения.
50. Назовите примеры прикладных программ.
51. Что означают термины «вертикальное» и «горизонтальное» приложение?
52. Что такое «масштаб приложения»?
53. Дайте описания следующим группам: настольное приложение, групповое приложение, приложение масштаба предприятия?

6. Стратегия разработки ПО

Самым существенным и неотъемлемым свойством программных систем является их сложность. Благодаря уникальности и несхожести своих составных частей программные системы принципиально отличаются от технических систем (например, компьютеров), в которых преобладают повторяющиеся элементы.

Сами компьютеры сложнее, чем большинство продуктов человеческой деятельности. Количество их возможных состояний очень велико, поэтому их так трудно понимать, описывать и тестировать. У программных систем количество возможных состояний на порядок величин превышает количество состояний компьютеров.

Потребность контролировать процесс разработки ПО, прогнозировать и гарантировать ее стоимость, сроки и качество результатов привела в конце 70-х гг. к необходимости перехода от кустарных к индустриальным способам создания ПО и появлению совокупности инженерных методов и средств создания ПО, объединенных общим названием **«программная инженерия»** (software engineering). Впервые этот термин был использован как тема конференции, проводившейся под эгидой НАТО в 1968 г. Спустя семь лет, в 1975 г., в Вашингтоне была проведена первая международная конференция, посвященная программной инженерии. Тогда же появилось первое издание, посвященное программной инженерии, — IEEE Transactions on Software Engineering.

В процессе становления и развития программной инженерии можно выделить два этапа: 70-е и 80-е гг. — систематизация и стандартизация процессов создания ПО и 90-е гг. — начало перехода к сборочному, индустриальному способу создания ПО.

В основе программной инженерии лежит одна фундаментальная идея: проектирование ПО является формальным процессом, который можно изучать и совершенствовать. Для успешной реализации проекта должны быть построены полные и непротиворечивые модели архитектуры ПО.

Под **моделью** понимается полное описание системы ПО с определенной точки зрения. Модели представляют собой средства для визуализации, описания, проектирования и документирования архитектуры системы. Модели строятся для того, чтобы понять и осмыслить структуру и поведение будущей системы, облегчить управление процессом ее создания и уменьшить возможный риск, а также документировать принимаемые проектные решения. В 70-80-х годах при разработке ПО достаточно широко применялись структурные методы, базирующиеся на строгих формализованных методах описания ПО и принимаемых технических решений. В настоящее время такое же распространение получают объектно-ориентированные методы. Эти методы основаны на использовании наглядных графических моделей для описания архитектуры ПО, в различных аспектах используются схемы и диаграммы. Наглядность и строгость методов структурного и объектно-ориентированного анализа позволяют разработчикам и будущим пользователям системы с самого начала неформально участвовать в ее создании, обсуждать и закреплять понимание основных технических решений. Перечисленные проблемы породили потребность в программно-технологических средствах специального класса — CASE-средствах, о которых будет рассказано ниже.

Жизненный цикл

Понятие **жизненного цикла** программного обеспечения (ЖЦ ПО) является одним из базовых в программной инженерии. Жизненный цикл программного обеспечения определяется как период времени, который начинается с момента принятия решения о необходимости создания ПО и заканчивается в момент его полного изъятия из эксплуатации.

Основным нормативным документом, регламентирующим состав процессов ЖЦ ПО, является международный стандарт ISO/IEC 12207: 1995 «Information Technology — Software Life Cycle Processes» (ISO — International Organization for Standardization — Международная организация по стандартизации, IEC — International Electrotechnical Commission — Международная комиссия по электротехнике). Структура ЖЦ по стандарту ISO/IEC 12207 базируется на трех группах процессов:

— основные процессы ЖЦ ПО (приобретение, поставка, разработка, эксплуатация, сопровождение);

— вспомогательные процессы, обеспечивающие выполнение основных процессов (документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, оценка, аудит, решение проблем);

— организационные процессы (управление проектами, создание инфраструктуры проекта, определение, оценка и улучшение самого ЖЦ, обучение).

Некоторые процессы ЖЦ разберем подробнее.

Разработка включает в себя работы по созданию ПО и его компонент в соответствии с заданными требованиями, включая оформление проектной и эксплуатационной документации, подготовку материалов, необходимых для проверки работоспособности и соответствующего качества программных продуктов, материалов, необходимых для организации обучения персонала и т. д. Разработка ПО включает в себя, как правило, анализ, проектирование и реализацию (программирование).

Эксплуатация включает в себя работы по внедрению компонентов ПО в эксплуатацию, в том числе конфигурирование базы данных и рабочих мест пользователей, обеспечение эксплуатационной документацией, проведение обучения персонала и т. д., и непосредственно эксплуатацию, в том числе локализацию проблем и устранение причин их возникновения, модификацию ПО в рамках установленного регламента, подготовку предложений по совершенствованию, развитию и модернизации системы.

Процесс **сопровождения** предусматривает действия и задачи, выполняемые сопровождающей организацией (службой сопровождения). Данный процесс активизируется при изменениях (модификациях) программного продукта и соответствующей

документации, вызванных возникшими проблемами или потребностями в модернизации либо адаптации ПО. В соответствии со стандартом IEEE-90 под сопровождением понимается внесение изменений в ПО в целях исправления ошибок, повышения производительности или адаптации к изменившимся условиям работы или требованиям.

Среди вспомогательных процессов особо выделим процесс **верификации** — определения того, отвечает ли текущее состояние разработки, достигнутое на данном этапе, требованиям этого этапа. Проверка позволяет оценить соответствие параметров разработки с исходными требованиями. Проверка частично совпадает с тестированием, которое связано с идентификацией различий между действительными и ожидаемыми результатами и оценкой соответствия характеристик ПО исходным требованиям. **Управление конфигурацией** позволяет организовывать, систематически учитывать и контролировать внесение изменений в ПО на всех стадиях ЖЦ. Общие принципы и рекомендации конфигурационного учета, планирования и управления конфигурациями ПО отражены в проекте стандарта ISO 12207-2. Процесс **аудита** представляет собой определение соответствия требованиям, планам и условиям договора. Аудит может выполняться двумя любыми сторонами, участвующими в договоре, когда одна сторона проверяет другую.

Каждый процесс характеризуется определенными задачами и методами их решения, исходными данными, полученными на предыдущем этапе, и результатами. Результатами анализа, в частности, являются функциональные модели, информационные модели и соответствующие им диаграммы. ЖЦ ПО носит итерационный характер: результаты очередного этапа часто вызывают изменения в проектных решениях, выработанных на более ранних этапах.

Модели жизненного цикла

Стандарт ISO/IEC 12207 не предлагает конкретную модель ЖЦ и методы разработки ПО (под моделью ЖЦ понимается структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач, выполняемых на протяжении ЖЦ). Стандарт ISO/IEC 12207 описывает структуру процессов ЖЦ ПО, но не конкретизирует в деталях, как реализовать или выполнить действия и задачи, включенные в эти процессы.

К настоящему времени наибольшее распространение получили следующие две основные модели ЖЦ:

- каскадная модель (70-85 г.г.);
- спиральная модель (86-90 г.г.).

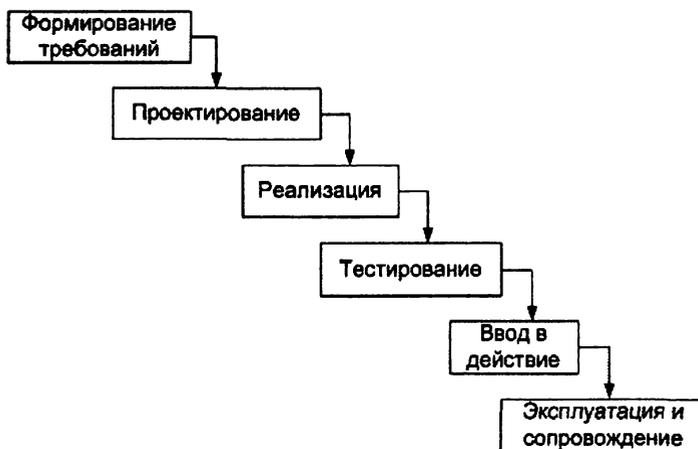


Рис. 7. Каскадная схема разработки ПО

В каскадной модели основной характеристикой является разбиение всей разработки на этапы, причем переход с одного этапа на следующий происходит только после того, как будет полностью завершена работа на текущем (рис. 7). Каждый этап завершается выпуском полного комплекта документации, достаточной для того, чтобы разработка могла быть продолжена другой командой разработчиков.

Положительные стороны применения каскадного подхода заключаются в следующем:

- на каждом этапе формируется законченный набор проектной документации, отвечающий критериям полноты и согласованности;
- выполняемые в логичной последовательности этапы работ позволяют планировать сроки завершения всех работ и соответствующие затраты. Однако в процессе использования этого подхода обнаружился ряд его недостатков, вызванных, прежде всего тем, что реальный процесс создания ПО никогда полностью не укладывался в такую жесткую схему. В процессе создания ПО постоянно возникала потребность в возврате к предыдущим этапам и уточне-

нии или пересмотре ранее принятых решений. В результате реальный процесс создания ПО принимал вид представленный на рис. 8.

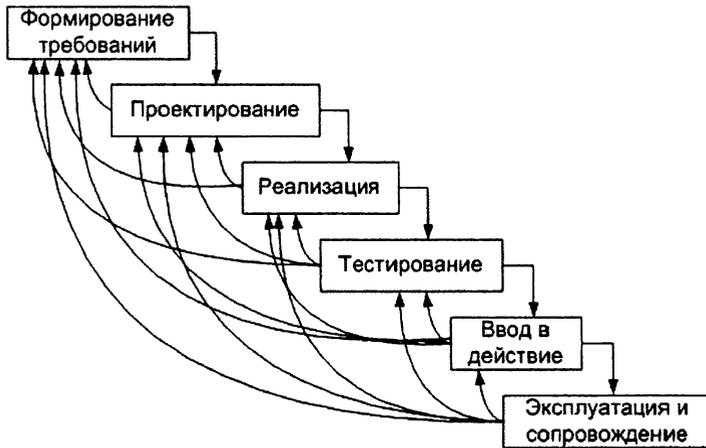


Рис. 8. Реальный процесс разработки ПО по каскадной схеме

Основным недостатком каскадного подхода является существенное запаздывание с получением результатов. Согласование результатов с пользователями производится только в точках, планируемых после завершения каждого этапа работ, требования к ПО «заморожены» в виде технического задания на все время его создания. Таким образом, пользователи могут внести свои замечания только после того, как работа над системой будет полностью завершена. В случае неточного изложения требований или их изменения в течение длительного периода создания ПО, пользователи получают систему, не удовлетворяющую их потребностям. Модели (как функциональные, так и информационные) автоматизируемого объекта могут устареть одновременно с их утверждением.

Для преодоления перечисленных проблем была предложена спиральная модель ЖЦ (рис. 9), делающая упор на начальные этапы ЖЦ: анализ и проектирование. На этих этапах реализуемость технических решений проверяется путем создания прототипов. Каждый виток спирали соответствует созданию фрагмента или версии ПО, на нем уточняются цели и характеристики проекта, определяется его качество и планируются работы следующего витка спирали.



Рис. 9. Спиральная модель ЖЦ

Таким образом, углубляются и последовательно конкретизируются детали проекта, и в результате выбирается обоснованный вариант, который доводится до реализации.

Разработка итерациями отражает объективно существующий спиральный цикл создания системы. Неполное завершение работ на каждом этапе позволяет переходить на следующий этап, не дожидаясь полного завершения работы на текущем. При итерационном способе разработки недостающую работу можно будет выполнить на следующей итерации. Главная же задача — как можно быстрее показать пользователям системы работоспособный продукт, тем самым активизируя процесс уточнения и дополнения требований. В силу этого будущий пользователь программного продукта в данной модели становится полноправным участником процессов разработки ПО (в первую очередь на этапах анализа и проектирования). Это предъявляет соответствующие требования и к заказчикам ПО (в том числе специалистам экономического профиля) — они должны владеть основными сведениями в области инженерии программирования, которые бы позволили им продуктивно общаться с разработчиками.

Основная проблема спирального цикла — определение момента перехода на следующий этап. Для ее решения необходимо ввес-

ти временные ограничения на каждый из этапов жизненного цикла. Переход осуществляется в соответствии с планом, даже если не вся запланированная работа закончена. План составляется на основе статистических данных, полученных в предыдущих проектах, и личного опыта разработчиков.

CASE-средства разработки приложений

Термин CASE (Computer Aided System Engineering) имеет весьма широкое толкование. В рамках программной инженерии CASE-средства представляют собой основную технологию, используемую для создания и эксплуатации систем ПО. Под **CASE-средством** понимается программное средство, поддерживающее процессы жизненного цикла ПО, включая анализ требований к системе, проектирование прикладного ПО и баз данных, генерацию кода, тестирование, документирование, обеспечение качества, управление конфигурацией ПО и управление проектом, а также другие процессы. CASE-средства вместе с системным ПО и техническими средствами образуют среду разработки ПО.

Наиболее трудоемкими стадиями разработки ПО являются стадии формирования требований и проектирования, в процессе которых CASE-средства обеспечивают качество принимаемых технических решений и подготовку проектной документации. При этом большую роль играют методы визуального представления информации. Это предполагает построение разнообразных графических моделей (диаграмм), использование многообразной цветовой палитры, сквозную проверку синтаксических правил.

В разряд CASE-средств попадают как относительно дешевые системы для персональных компьютеров с весьма ограниченными возможностями, так и дорогостоящие системы для неоднородных вычислительных платформ и операционных сред. Так, современный рынок программных средств насчитывает около 300 различных CASE-средств, наиболее мощные из которых, так или иначе, используются практически всеми ведущими западными фирмами.

CASE-средствам присущи следующие основные особенности:

- наличие мощных графических средств для описания и документирования системы, обеспечивающих удобный интерфейс с разработчиком и развивающих его творческие возможности;
- интеграция отдельных компонентов CASE-средств, обеспечивающая управляемость процессом разработки ПО;

— использование специальным образом организованного хранилища проектных метаданных (репозитория).

Интегрированное CASE-средство (комплекс средств, поддерживающих полный ЖЦ ПО) содержит следующие компоненты:

— репозиторий, являющийся основой CASE-средства. Он должен обеспечивать хранение версий проекта и его отдельных компонентов, синхронизацию поступления информации от различных разработчиков при групповой разработке, контроль метаданных на полноту и непротиворечивость;

— графические средства анализа и проектирования, обеспечивающие создание и редактирование комплекса взаимосвязанных диаграмм, образующих модели деятельности организации и системы ПО;

— средства разработки приложений, включая языки 4GL (Fourth Generation Language — язык 4-го поколения) и генераторы кодов;

— средства управления требованиями;

— средства управления конфигурацией ПО;

— средства документирования;

— средства тестирования;

— средства управления проектом;

Основные функции средств организации и поддержки репозитория — хранение, доступ, обновление, анализ и визуализация всей информации по проекту ПО. Содержимое репозитория включает не только информационные объекты различных типов, но и отношения между их компонентами, а также правила использования или обработки этих компонентов. Репозиторий может хранить свыше 100 типов объектов, примерами которых являются диаграммы, определения экранов и меню, проекты отчетов, описания данных, исходные коды и прочее.

Вопросы для самопроверки

54. Что такое программная инженерия?
55. Когда появился этот термин?
56. Определите понятие «модель»?
57. Для чего необходима модель?
58. Выделите этапы становления программной инженерии.
59. Что такое жизненный цикл ПО?
60. Назовите базовые группы процессов ЖЦ.
61. Какие работы включает в себя процесс верификации?
62. Что представляет собой процесс аудита?

- 63. Что понимается под моделью ЖЦ?
- 64. Назовите наиболее распространенные модели ЖЦ.
- 65. Чем они отличаются?
- 66. Что такое CASE-средства?

7. Технология быстрой разработки приложений — RAD

Давайте рассмотрим наиболее популярные на сегодняшний день подходы к разработке программного обеспечения, и некоторые примеры их использования.

Описание технологии

В современной инженерии программирования существуют различные технологические методы и приемы разработки программных продуктов. Выбор той или иной технологии разработки обуславливает разные временные и трудовые ресурсы, а также различную квалификацию разработчиков. Потенциальному потребителю программного продукта нужно иметь представление об особенностях и возможностях различных технологий. Наиболее функционально развитой на сегодня является технология быстрой разработки приложений — RAD. Подход RAD (Rapid Application Development) предусматривает наличие следующих составляющих:

- развитых средств визуального программирования, основанных на использовании готовых фрагментов программ;
- небольших групп разработчиков (от 3 до 7 человек), выполняющих работы по проектированию отдельных подсистем ПО. Это обусловлено требованием максимальной управляемости коллектива;
- короткого, но тщательно проработанного производственного графика (до 3 месяцев);
- повторяющегося цикла, при котором разработчики по мере того, как приложение начинает обретать форму, запрашивают и реализуют в продукте требования, полученные в результате взаимодействия с заказчиком.

Команда разработчиков должна представлять собой группу профессионалов, имеющих опыт в проектировании, программировании и тестировании ПО, способных хорошо взаимодействовать с конечными пользователями (менеджерами проекта, экономистами) и трансформировать их предложения в рабочие прототипы. В основе технологии RAD лежит использование современных средств визу-

ального программирования, в которых высокая скорость разработки приложений связана с возможностью создавать и настраивать различные интерфейсные элементы без написания кода программы.

В настоящее время для разработки программного обеспечения активно используется целый спектр инструментальных сред. Это продукты компании Microsoft — Visual C++ и Visual Basic, популярный продукт компании Borland — среда программирования Delphi, построенная на основе Pascal и др. В зависимости от решаемой задачи разработчик вправе выбрать тот или иной инструмент.

Применение к офисному программированию

Крупнейший производитель программного обеспечения — корпорация Microsoft, создала и активно продвигает как удобный инструмент разработки офисных приложений интегрированный в пакет MS Office язык визуального программирования VBA (Visual Basic for Applications). Для специалистов он интересен прежде всего тем, что позволяет естественным образом автоматизировать решение своих повседневных задач.

Чтобы начать знакомство со средой программирования VBA, необходимо выполнить следующие действия. Запустить приложение MS Office (здесь и далее — MS Excel). В меню программы выбрать команду *Сервис — Макрос — Редактор Visual Basic* (или комбинацию Alt+F11). В результате откроется интегрированная среда разработки VBA.

Она имеет стандартный для всех Windows-приложений вид: строка меню, панель инструментов и два окна — *Project* и *Properties*.

Рассмотрим для начала окно Project. В нем отображается список форм и модулей, входящих в создаваемый проект. Модуль, упрощенно говоря, это лист (не путать с рабочим листом), в котором набирается код программы. Двойным щелчком на значке модуля в окне Project можно открыть соответствующий модуль. В VBA каждый рабочий лист, как и рабочая книга, имеет собственный модуль. Для того чтобы добавить стандартный модуль в проект, необходимо в меню редактора выбрать *Insert — Module* (вставка модуля).

Рассмотрим пример создания простейшей программы для вывода диалогового окна с надписью «Hello, world!». Для этого необходимо в модуле ввести следующий код:

```
Sub Hello ()  
    MsgBox "Hello, world!"
```

После нажатия Enter, текст автоматически дополнится еще одной строкой (*End Sub*), и программа готова. Запустить ее можно выбрав в меню *Run — Sub/UserForm* (или нажать F5). Результат работы и окно редактора представлены на рисунке 10.

В только что созданной программе первая и третья строки — это первая и последняя инструкции **процедуры**, написанной на языке VBA. Процедура — это минимальная семантически законченная программная конструкция, допускающая выполнение. Процедура имеет следующий синтаксис:

```
Sub name (arglist)
    [statements]
End Sub
```

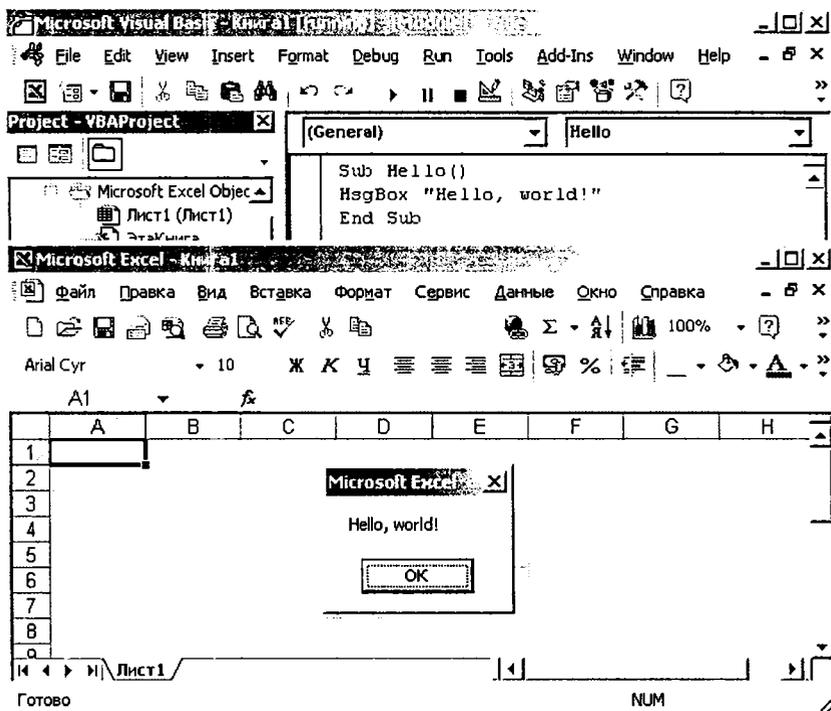


Рис. 10. Результат работы программы с окном редактора VBA

Здесь *name* — имя процедуры; *arglist* — список аргументов, который может быть и пустым; *statements* — это последовательность операторов (инструкций). Оператор *MsgBox* имеет следующий синтаксис:

`MsgBox Text`

В результате его действия на экране отображается диалоговое окно с кнопкой ОК и сообщением, текст которого задается параметром *Text*.

Созданную таким образом процедуру, можно запустить не только из инструментальной среды VBA но и с помощью кнопки панели инструментов Excel. Например, чтобы создать такую кнопку, необходимо выполнить следующее:

1. В MS Excel используя команду *Сервис* — *Настройка* или команду *Вид* — *Панели инструментов* — *Настройка*, вызвать окно диалога *Настройка*.

2. Выбрать вкладку *Команды*, затем нажать на категории *Макросы*.

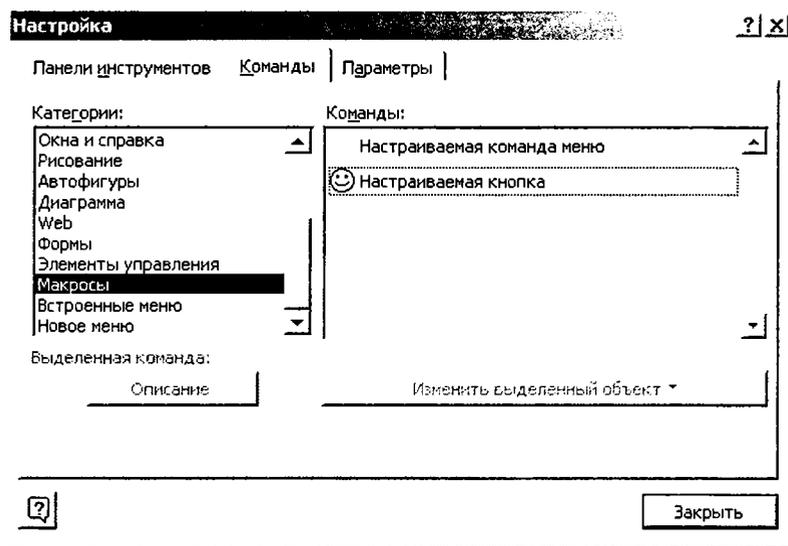


Рис. 11. Окно настройки панелей инструментов

3. Выберите *Настраиваемая кнопка* и с помощью мыши перетащите ее на предварительно открытую панель инструментов. Это приведет к установке кнопки на встроенной панели инструментов. При первом нажатии на нее на экране появится окно, предлагаю-

щее указать имя макроса (процедуры) из списка, который будет вызываться нажатием на нее. В нашем примере выбираем *Hello* и нажимаем ОК.

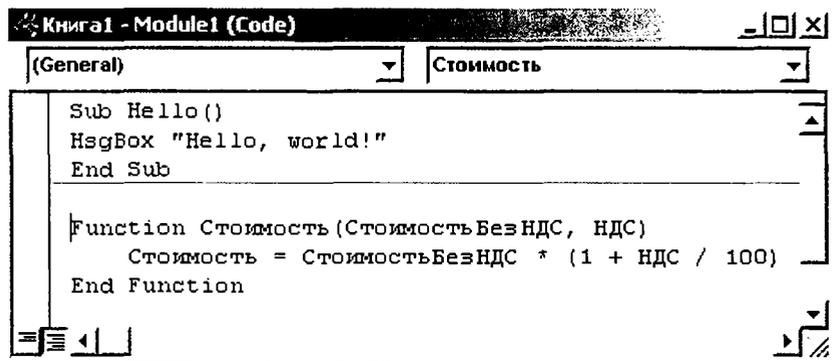
Другой пример использования программирования в MS Office — создание пользовательских функций. Функция отличается от процедуры тем, что помимо выполнения операторов ею возвращается некоторое значение. Собственно процедуру (*sub*) называют также процедура-подпрограмма, а функцию (*function*) — процедура-функция. Чтобы в существующий модуль с примером *Hello* добавить функцию, необходимо в меню VBA выбрать *Insert — Procedure*, вписать имя, и указать тип создаваемой *процедуры* — функция, после его нажать ОК. В код модуля будет добавлено описание функции. Синтаксис описания функции:

```
Function name (arglist) [as type]
    [statements]
End Function
```

Назначение элементов описания функции аналогично описанию процедуры, но есть и отличия. Так, после указания имени функции и описания аргументов, следует указание типа возвращаемого результата (блок *as type*). Также в коде функции должен присутствовать оператор

```
name = expression,
```

где *expression* — выражение, определяющее значение функции.



```
Sub Hello ()
    MsgBox "Hello, world!"
End Sub

Function Стоимость (СтоимостьБезНДС, НДС)
    Стоимость = СтоимостьБезНДС * (1 + НДС / 100)
End Function
```

Рис. 12. Окно модуля с текстом примеров

Для закрепления сказанного можно построить простую пользовательскую функцию для расчета стоимости товара по его стоимости без НДС и значению ставки НДС. Для этого необходимо занести в код модуля следующий текст:

```
Function Стоимость(СтоимостьБезНДС, НДС)  
Стоимость = СтоимостьБезНДС * (1+НДС/100)  
End Function
```

Описанная таким образом функция попадает в категорию *Определенные пользователем* списка *Категория* окна *Мастер функций*. А значит, предполагая, что в ячейке A2 хранится стоимость без НДС, а в ячейке B2 — НДС, можно рассчитать стоимость товара, занеся в ячейку C2, например, следующую формулу:

= Стоимость (A2;B2)

Вопросы для самопроверки

67. Назовите составляющие RAD.
68. Использование каких средств лежит в основе технологии RAD?
69. Какие инструментальные среды используются для разработки ПО?
70. Что такое VBA?
71. Как запустить редактор VBA?
72. Приведите пример программы для вывода диалогового окна с надписью «Привет!». Прокомментируйте приведенный пример.
73. Определите понятие «процедура».
74. Синтаксис процедуры.
75. Что такое функция?
76. Синтаксис описания функции.
77. Чем функция отличается от процедуры?
78. Приведите пример создания пользовательской функции.

8. Элементы языка VBA

Как правило, списка аргументов процедуры недостаточно для вычислений. Тогда описание подпрограммы дополняют описанием переменных. **Переменной** называется имя, определяющее область памяти для хранения величины, которая может изменяться во время работы программы. В каждом из языков программирования определены правила, определяющие, какие имена можно давать пере-

менным, процедурам и константам. В VBA имена должны удовлетворять следующим правилам:

- первым символом имени обязательно должна быть буква;
- нельзя использовать символы «.», «!», «@», «&», «\$», «#» и пробел;
- имя может содержать буквы, цифры и знак подчеркивания;
- имя не может содержать более 255 символов;
- имя не должно совпадать с ключевыми словами VBA;
- имена не должны повторяться на одном уровне определения, то есть в одной процедуре не могут быть объявлены две переменные с одним и тем же именем.

Тип данных	Описание
Byte	Байт
Boolean	Логический
Integer	Целый
Long	Длинный целый
Single	С плавающей точкой обычной точности
Double	С плавающей точкой двойной точности
Currency	Денежный
Decimal	Масштабируемое целое
Date	Даты и время
Object	Ссылка на объект
String	Строка
Variant	Хранит переменную любого типа

Перед использованием переменной в программе ее необходимо сначала объявить, то есть задать ее тип и область видимости (область использования переменной). Тип переменной определяет множество допустимых значений, которые может принимать переменная указанного типа, а также объем, который она занимает в памяти, и операции, применимые к ней.

Синтаксис объявления переменных имеет вид:

Dim VariableName [As] [Type], где

Dim — зарезервированное слово, с которого начинается описание переменных;

VariableName — имя переменной;

As Type — указание **типа** переменной (например, As Integer или As String).

VBA не требует обязательного объявления переменных. В случае если переменная не была объявлена, ей автоматически присваивается тип Variant. Этот тип является универсальным и может содержать данные различных подтипов: Integer, Long, String и т. п. Так в примере с функцией *Стоимость* было пропущено указание типа как самой функции, так и ее аргументов. Но следует помнить, что использование универсального типа затрудняет отладку программы. Чтобы избежать этого, рекомендуется всегда объявлять переменные и присваивать им необходимый тип. Задание переменной типа, отличного от Variant, позволяет также экономить память, так как переменные соответствующих подтипов типа Variant занимают больший объем памяти. Ниже приведен пример блока процедуры с описанием переменных.

```
Sub Test ()
```

```
Dim Count As Integer, Cash As Long
```

```
Dim Results As String
```

Для изменения типа переменной можно использовать имеющиеся в VBA функции преобразования типа. Так, функция *CStr(expression)* возвращает значение выражения (expression) в строковом виде. А функции *Cdbl(expression)* и *CSng(expression)* могут преобразовать строковую запись числа в переменную типа Double и Single, соответственно. Приведем пример использования этих функций.

```
Dim sTXRT As String
```

```
Dim dTXRT As Double
```

```
sTXRT = "13,0"
```

```
dTXRT = cDbl(sTXRT)
```

В этом примере переменной sTXRT присваивается строковое значение "13,0", которое потом после преобразования функцией Cdbl к числовому вещественному типу присваивается переменной dTXRT.

Массив (array) — это упорядоченная группа однотипных переменных, которые объединены одним общим именем. Их можно считать одной из разновидностей переменных. Отличие состоит в том, что в массивах можно хранить не одно, а несколько значений. Доступ к заданному элементу массива осуществляется с помощью индекса. Синтаксис объявления массива отличается от синтаксиса объявления переменной тем, что здесь требуется указать также

размерность массива и границы изменения индексов. Ниже представлен пример объявления массива:

```
Dim B(3,3) As Single
Dim A(11) As Integer
```

Первая инструкция объявляет двумерный массив 3x3, состоящий из действительных чисел. Вторая инструкция объявляет одномерный массив из 12 целых чисел, причем по умолчанию первый элемент массива будет A(0), а последний — A(11). Массив в программе определяется поэлементно. Например, так можно поэлементно определить массив B(2,2):

```
Dim B(2,2) As Single
B(1,1)=2: B(1,2)=4
B(2,1)=5: B(2,2)=1
```

Для повышения наглядности при написании программ вместо использования какого-либо постоянного значения часто используют константы. Применение констант облегчает восприятие текста программы, а также значительно упрощает отладку программы.

В VBA существуют константы двух типов:

— **Встроенные константы.** Список этих констант можно просмотреть в окне просмотра объектов. Их количество определяется используемым приложением. Например, к таким константам относятся True, False, Null.

— **Пользовательские константы** объявляются с помощью инструкции Const.

В момент объявления пользовательским константам присваиваются значения. Приведем пример объявления и использования пользовательской константы:

```
Sub TestSub()
    Const sngЧислоПи As Single = 3.14
    Dim sngРадиусКруга As Single
    Dim sngПлощадьКруга As Single
    sngРадиусКруга = 5
    sngПлощадьКруга = эдЧислоПи * sngРадиусКруга
End Sub
```

В программах VBA можно использовать стандартный набор операций над данными, записываемых при помощи **операторов**.

Например, оператор «+» выполняет операцию сложения двух чисел или выражений, являющихся операндами. Большинство операторов VBA требуют наличия двух операндов. Все простые операторы можно разделить на несколько групп:

— Арифметические операторы. Они используются для выполнения основных арифметических операций над операндами, представляющими числовые выражения. Определены следующие арифметические операторы: сложение (+), вычитание (-), умножение (*), деление (/), возведение в степень (^) и целая часть от деления двух операндов (\).

— Логические операторы предназначены для выполнения логических операций, таких, как логическое сложение (or), логическое умножение (and) и логическое отрицание (not).

— Операторы сравнения (<,>=,Like).

Особо нужно выделить так называемые структурные операторы (или операторы управления). Первый из них — оператор выбора (условного перехода). Оператор выбора задает выполнение определенных групп инструкций в зависимости от значения выражения.

Синтаксис:

```
If condition Then [statements] [Else elsestatements]
```

Если условие *condition* принимает значение True, то выполняются инструкции *statements*, иначе — инструкции *elsestatements*. Ветвь *Else* является необязательной. Например, если скидка (скажем 5%) применяется только к суммам больше 1000, то в VBA это можно записать так:

```
If Money>1000 Then Discount = 0.05 Else Discount = 0
```

Допускается также использование блочной формы синтаксиса, которая часто упрощает восприятие оператора условного перехода:

```
If condition Then  
    [statements]  
[Else]  
    [elsestatements]  
End If
```

Дерево условий может оказаться гораздо более сложными, чем просто проверка одного условия. В этом случае используется оператор If...Then...ElseIf, который позволяет проверять множественные

условия. Следующий пример показывает порядок проверки условий.

Предположим, что менеджеру по продажам необходимо разработать функцию, позволяющую рассчитать комиссионные. Процент комиссионных зависит от объема продаж и начисляется по следующему правилу:

от 0 до 999	— 8%
от 1000 до 1999	— 10%
от 2000 до 3999	— 12%
свыше 4000	— 14%

Построим для расчета комиссионных следующую функцию:

```
Function Комиссионные (Продажи)
    If Продажи<=999 Then
        Комиссионные = Продажи*0.08
    ElseIf Продажи<=1999 Then
        Комиссионные = Продажи*0.1
    ElseIf Продажи<=3999 Then
        Комиссионные = Продажи*0.12
    Else
        Комиссионные = Продажи*0.14
    End If
End Function
```

Вторая группа структурных операторов — циклы. Эти операторы предназначены для многократного выполнения группы инструкций. В VBA для организации циклов используются: For...Next, While...Wend, Do...Loop и For Each...Next.

Оператор For...Next

Этот оператор повторяет выполнение группы инструкций указанное число раз. Синтаксис:

```
For counter = start To end [Step step]
    [statements]
[Exit For]
    [statements]
Next [counter]
```

Оператор повторяет выполнение группы инструкций *statements* пока счетчик *counter* изменяется от начального *start* до конечного

end значения с указанным шагом *step*. Если шаг не задан, он полагается равным 1.

Оператор For Each...Next

Этот оператор повторяет выполнение группы инструкций *statements* для каждого элемента *element* массива *group*. Синтаксис:

```
For Each element In group
    [statements]
[Exit For]
    [statements]
Next [element]
```

Оператор While...Wend

Оператор *While...Wend* выполняет последовательность конструкций *statements*, пока заданное условие *condition* имеет значение *True*. Синтаксис:

```
While condition
    [statements]
Wend
```

Оператор повтора *While...Wend* в отличие от *For...Next* работает не заданное число раз, а пока выполняется условие *condition*.

Оператор Do ... Loop

Оператор *Do...Loop* повторяет выполнение набора инструкций *statements* пока условие *condition* имеет значение *True* (случай *While*) или пока оно не примет значение *True* (случай *Until*). Синтаксис:

```
Do
    [statements]
Loop [While | Until condition]
```

Вопросы для самопроверки

79. Определите понятие «переменная».
80. Какие существуют правила имен переменных?
81. Синтаксис объявления переменных.
82. Какие существуют типы данных в VBA? Приведите их описание.

83. Можно ли не описывать переменные в VBA?
84. Тип переменных Variant. Чем он примечателен?
85. «Плюсы» и «минусы» использования типа Variant.
86. Какие функции служат для преобразования типов? Приведите пример использования этих функций.
87. Определите понятие «массив».
88. Синтаксис объявления массива.
89. Перечислите типы констант. Примеры.
90. Перечислите группы операторов в VBA.
91. Оператор выбора. Синтаксис, пример использования.
92. Оператор For...Next. Синтаксис, пример использования.
93. Оператор For Each...Next. Синтаксис, пример использования.
94. Оператор While...Wend. Синтаксис, пример использования.
95. Оператор Do Loop. Синтаксис, пример использования.

9. Практика использования VBA — объекты MS Excel

VBA — пример визуального объектно-ориентированного языка программирования высокого уровня, в котором все визуальные составляющие, такие как формы, элементы управления, меню и панели инструментов являются объектами со своими свойствами и методами.

Объект — это объединение данных с кодом, предназначенным для обработки. Каждый **объект** является экземпляром какого-либо **класса**. Класс задает свойства и поведение объектов класса. Например, каждый рабочий лист MS Excel является объектом класса Worksheet. Таким образом, класс — это логическая конструкция, а объект — это его физическое воплощение. Класс содержит данные и функции, которые обрабатывают данные.

Объект сам по себе не представляет большого значения. Намного существеннее то, какие действия можно совершать над объектом, и какими свойствами он обладает. **Метод** как раз и представляет собой действие, выполняемое над объектом. Синтаксис применения метода:

```
Object.Method
```

Например, метод *Delete*, примененный к рабочему листу, удаляет его из книги.

```
Worksheets(1).Delete
```

Свойство представляет собой атрибут объекта, определяющий его характеристики, такие как размер, цвет, положение на экране или состояние объекта, например, доступность для пользователя или видимость. Чтобы изменить характеристики объекта, надо просто изменить значения его свойств. Синтаксис установки значения свойства:

```
Object. Property = PropertyValue
```

В следующем фрагменте кода в ячейку A1 вводится текст *Пример* посредством установки значения свойства *Value* объекта *Range("A1")*:

```
Range("A1").Value = "Пример"
```

Значения свойств можно не только устанавливать, но и считать. В следующем примере выводится сообщение о том, введено ли в ячейку A1 слово *Пример*.

```
If Range("A1").Value = "Пример" Then  
    MsgBox "В ячейку A1 введено слово Пример"  
Else  
    MsgBox "В ячейку A1 введено какое-то другое слово"  
End If
```

Особое место среди свойств занимают те, которые сами возвращают объект. Например, свойство *Rows* объекта *Range* (диапазон ячеек) возвращает объект *Rows*, через свойства которого можно получить доступ к ячейкам указанного диапазона. Так, конструкция

```
Rcount = Range("A1:A5").Rows.Count
```

присвоит переменной *Rcount* значение 5 — количество ячеек в диапазоне *A1:A5*. Используем эту конструкцию для расчета суммы чисел из ячеек диапазона с именем "Данные". Напомним, — чтобы присвоить имя диапазону ячеек, необходимо, выделив ячейки, в меню выбрать *Вставка — Имя — Присвоить*, и в открывшемся окне занести имя диапазона (в данном примере — "Данные"). Результат расчета будет помещен в ячейку с именем "Результат"

Текст макроса будет выглядеть следующим образом:

```
Sub Test()  
    Dim S As Single  
    Dim i As Byte
```

```

For i = 1 To Range («Данные»).Rows.Count
    S = S + Range («Данные»)(i).Value
Next i
Range («Результат»).Value = S
End Sub

```

В приведенном примере после объявления переменных следует оператор цикла *For...Next*, который выполнит операцию $S=S+Range(«Данные»)(i).Value$ столько раз, сколько ячеек содержит диапазон “Данные”. Оператор $Range(«Результат»).Value = S$ запишет результат в ячейку рабочего листа.

Вопросы для самопроверки

96. Определите следующие понятия: объект, класс. Приведите пример.

97. Что такое метод? Синтаксис метода. Пример метода.

98. Свойство. Определение, синтаксис установки, пример использования.

99. Назначение свойства *Rows* объекта *Range*. Пример использования.

10. Программное обеспечение для решения финансово-экономических задач

В данной главе рассмотрены некоторые наиболее распространенные на российском рынке программные продукты бухгалтерского учета.

В начале 90-х гг. зародился отечественный рынок экономических программ. Эти программы, выйдя из стен вычислительных центров и отделов АСУ, попали непосредственно на рабочий стол бухгалтера и стали рассматриваться как продукт, товар. И многие пользователи уже понимают разницу между надежной, мощной, устойчивой в финансовом отношении фирмой-разработчиком и разработчиком-одиночкой, между фирменным отработанным «коробочным» продуктом с хорошей документацией, гарантированной поддержкой в самых дальних регионах и постоянно дорабатываемой кустарной узкопрофильной разработкой местного значения. Не претендуя на полноту, рассмотрим, тем не менее, некоторые предлагаемые на российском рынке промышленные программные продукты бухгалтерского профиля.

«1С:Бухгалтерия»

Производитель: Фирма «1С».

Адрес в Интернете: www.1c.ru.

Функциональные модули: бухгалтерия, зарплата и кадры, производство и услуги, торговый и складской учет.

Системные требования: Windows 9x/NT/2000/ME; 16МБ оперативной памяти; 20+ МБ на диске.

Стоимость однопользовательской версии: \$70 (базовая версия), \$240 (проф).

Другие версии, дополнительные модули: сетевая версия (SQL Server), масса типовых конфигураций и утилит от «1С» и сторонних разработчиков.

1С-Бухгалтерия уже давно перестала быть просто бухгалтерской программой. Правильней будет назвать ее бухгалтерским конструктором, который позволяет создать любую систему учета с минимальными ограничениями, которые накладывает природа обрабатываемых данных. Впрочем, поставляемые фирмой типовые конфигурации действительно делают из этого конструктора систему бухгалтерского учета в РФ и на Украине.

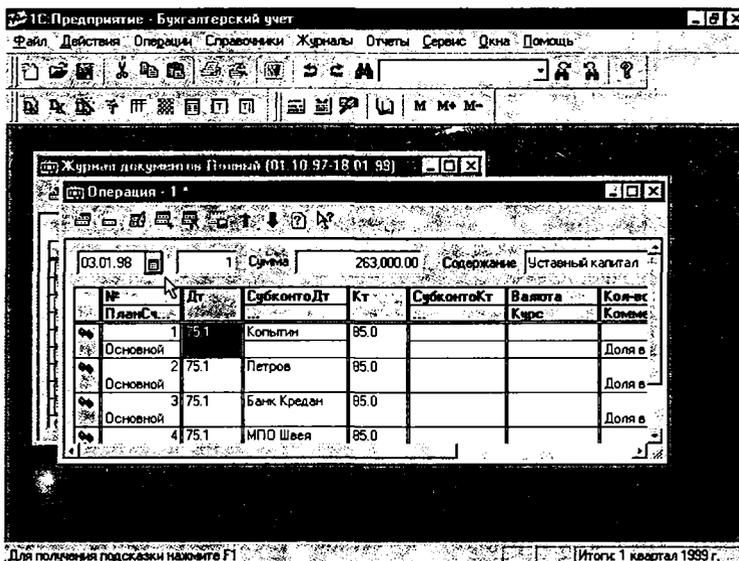


Рис. 13. Внешний вид программы.

При запуске можно выбрать одну из установленных информационных баз (отдельных предприятий, для которых можно вести отдельный учет) и режим запуска — рабочий или конфигурационный. В рабочем режиме существенно ограничены возможности настройки программы, которые вынесены как раз в Конфигуратор. В базовой версии программы типовая конфигурация защищена от изменения, поэтому и стоимость системы низка. Такой подход представляется разумным — среди возможностей конфигурации, которые ориентированы на профессионального программиста, бухгалтер просто запутается. А в рабочем режиме ему доступно только пополнение справочников и незначительное редактирование сформированных отчетов — просто и удобно.

Основой работы в 1С является операция. Операция состоит из ряда проводок и может быть введена вручную или порождена документом. Для упрощения работы предназначены типовые операции и стандартные документы. Большинство повседневных учетных операций охватываются этими двумя средствами быстрого ввода.

— Возможности системы во многом зависят от установленной конфигурации, поэтому нужно определить, какие виды учета будут вестись на предприятии, и выбрать одну из конфигураций:

- «Бухгалтерский учет»;
- «Торговля+Склад»;
- «Зарплата+Кадры»;
- «Производство+Услуги+Бухгалтерия»;
- «Бухгалтерия+Торговля+Склад+Зарплата+Кадры».

Стандартная поставка ориентирована в большей степени на торговлю и несложное производство, но существуют и варианты для других родов деятельности и форм собственности.

Аналитика («Субконто») имеет неограниченную вложенность, а в конфигураторе можно изменить и состав реквизитов по каждому виду субконто. Конечно, при необходимости обеспечивается мультивалютный учет.

Список встроенных отчетов включает оборотно-сальдовые ведомости, «шахматку», ведомости по счету, средства анализа счета и другие. Дополнительно поставляемые отчеты являются в основном внешними — баланс, отчеты во внебюджетные фонды и т. п.

База данных хранится в стандартных и широко распространенных DBF-файлах, и потому нет проблем с хранением, архивацией и

аварийным восстановлением данных. Единственной трудностью будет высокая сложность таблиц, поскольку все они являются перенастраиваемыми и хранят в основном метаданные. В программу встроены неплохие средства аварийного восстановления, которые справляются с типовыми сбоями в структуре данных.

Все серьезные настройки проводятся с помощью модуля «Конфигуратор», в котором в древовидной форме представлены все объекты программы, а пользователю предоставлена возможность изменить характеристики любого объекта, добавить или удалить объект. С помощью этого инструмента можно переработать существующие или добавить новые отчеты, определить новые типы справочников, стандартных операций и документов, изменить сценарии расчета и т. п. Настраиваемость великодушная, но для выполнения этих работ рекомендуется привлекать сертифицированного фирмой «ИС» специалиста или хорошего программиста, способного детально разобраться в языке сценариев.

Пользовательский интерфейс нельзя назвать простым, но после короткого привыкания его признаешь и удобным, и эргономичным. В списки операций и проводок с помощью стандартных кнопок «новая операция», «новая проводка» и «новый документ» можно быстро ввести любую бухгалтерскую информацию, а единственной тонкостью на пути к получению основных отчетов можно признать необходимость расчета итогов. Немного неудобно клавиатурное управление при вводе проводок, но если к нему привыкнуть, заводить новые операции можно очень быстро.

Достоинства: четкое разделение настройки и использования, что упрощает и то и другое. Самые широкие на рынке возможности по настройке, обилие типовых конфигураций.

Недостатки: недостаточно высокая скорость обработки (особенно больших объемов данных), большая перегруженность телефонной службы технической поддержки.

Рекомендуется: для малых и средних предприятий любого профиля.

«Инотек-Бухгалтер»

Производитель: АО «Инотек НТ».

Адрес в Интернете: www.inotec.ru.

Функциональные модули: бухгалтерия, касса, учет договоров, учет ОС и НА, учет ТМЦ, зарплата.

ты — интерфейс подсистемы «Инотек: документы» выполнен на современном уровне и позволяет не только вводить документы, но и редактировать их бланки. Но внутри этой подсистемы происходит лишь работа с первичными документами, а встроенные отчеты редактированию практически не подлежат — можно менять текст, но хотя бы перенести объект на сантиметр ниже уже нельзя. Модуль построения произвольных отчетов есть, и с его помощью созданы финансовые отчеты, но он подразумевает использование старых форм, совместимых с файлами MS-DOS (со шрифтами фиксированной ширины, отсутствием оформительских возможностей и рисованием псевдографикой).

В этот интерфейс упакована система, обладающая следующими возможностями:

- ведение журнала хозяйственных операций в рублях и валюте оценки;

- ведение аналитического учета на основе настраиваемого плана счетов, имеющего 7-уровневую иерархическую структуру субсчетов, в разрезе до 3 аналитических характеристик (объектов учета) по каждому счету (субсчету) и до 8 аналитических отметок — по каждой проводке;

- ведение мультивалютного учета с автоматизированным расчетом курсовых разниц и возможностью замены базовой валюты (рубли) на любую иную валюту;

- разnosку проводок непосредственно из тела заполняемого первичного документа и построение документов на основе выполненных проводок;

- учет взаиморасчетов с дочерними предприятиями, сторонними организациями и подотчетными лицами;

- ведение кассовой книги;

- учет договоров и контрактов;

- учет основных средств и нематериальных активов с возможностью выбора различных моделей расчета амортизации;

- ведение учета товарно-материальных ценностей и списание их по средним ценам либо на основе методов ЛИФО и ФИФО;

- расчет заработной платы в нескольких валютах.

Для выполнения сложных проводок и построения произвольных отчетов может использоваться специальный макроязык, хотя он менее удобный и мощный, чем в 1С.

Данные хранятся в файлах СУБД RAIMA, которая мало распространена, поэтому восстановление данных после сбоя сторонними организациями будет проблематичным. В связи с этим рекомендуется при покупке программы сразу же приобретать дополнительный модуль архивации данных. В принципе, система допускает ведение учета на нескольких предприятиях, но для управления этой функцией придется использовать специальную утилиту, которая не входит в стандартный комплект поставки.

Достоинства: компактность, высокая скорость работы.

Недостатки: многие элементы программы явно живут со времен DOS — неудобный интерфейс, бедное оформление отчетов, нестандартные базы данных.

Рекомендуется: для пользователей предыдущих версий и маломощных компьютеров.

«Турбо-Бухгалтер»

Производитель: фирма «ДИЦ».

Адрес в Интернете: www.dic.ru.

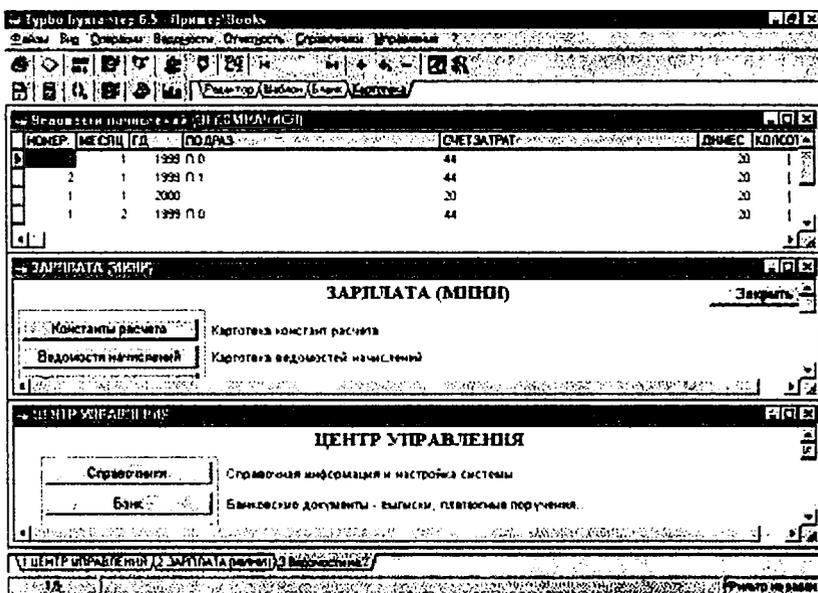


Рис. 15. Внешний вид программы

Функциональные модули: бухгалтерия, мини-зарплата, мини-учет ТМЦ, учет ОС и НА, командировка, налогоплательщик.

Системные требования: Windows 9x/NT/2000, 16 Мб оперативной памяти, 30+ Мб на диске.

Стоимость однопользовательской версии: \$155 (базовая). \$359 (версия «Проф»).

Другие версии, дополнительные модули: версия «Проф» (многовалютный учет, неограниченное число проводок, отладчик бланков), «Лайт» (только бухгалтерия и зарплата), DOS-версия 3.2, клиент-серверная версия, дополнительные модули Зарплата и Налогоплательщик.

Идеология разработки «Турбо-Бухгалтер» основывается на простом положении — «Бухгалтер работает от первичного документа». Уже на первом экране программы пользователя встречает простой и понятный рубрикатор «Банк», «Касса», «Учет НДС», «Учет ТМЦ» и т. д.

Основными «китами», на которых построена система, являются бланки, картотеки, средства ручного ввода операций и журналы операций. Бланком может являться и форма ввода, и отчет, и элемент пользовательского интерфейса для управления программой. Картотека хранит однотипные документы или справочники с дополнительной информацией. Назначение журнала операций понятно из его названия. Все бланки и картотеки являются полностью настраиваемыми, поэтому обладатели версии «Проф» могут переделать программу по своему вкусу. В базовой версии эти возможности доступны не полностью, но зато ее предустановленные настройки и интерфейс очень удобны для начинающего пользователя. Посмотрим, какие возможности предоставляет стандартная поставка программы.

«Турбо-бухгалтер» позволяет автоматизировать банковские и кассовые операции, расчет зарплаты, учет ОС и нематериальных активов, учет ТМЦ и МБП, расчеты с контрагентами, делопроизводство, построение внутренних отчетов — оборотных ведомостей, шахматных ведомостей, журналов-ордеров, главной книги и т. п., отчетов для налоговых органов. Бухгалтерию можно разделять на участки учета, допускается ведение бухгалтерии нескольких предприятий на одном компьютере. Программа позволяет анализировать различные аспекты финансовой и хозяйственной деятельности

предприятия: расчеты с покупателями и поставщиками, выполнение договоров, наличие и движение товаров и основных средств, учет затрат и прибыли различных подразделений. Возможен аналитический учет неограниченной глубины, благодаря этому возможно получать подробную информацию, например, о поставщике по конкретному договору (или группе договоров) на поставку определенного товара (или группы товаров) за заданный период времени. Поддерживается несколько баз данных, т. е. возможность работы с несколькими организациями. Профессиональная версия обеспечивает мультивалютный учет. Базовая версия имеет ограничение на число проводок — не более 32000 на одну организацию в год.

Расчетное ядро программы основывается на мощном языке программирования, похожем на Паскаль, и расширенными функциями работы с картотеками, бланками и бухгалтерскими терминами. На его основе можно даже создавать собственные приложения, но для этого потребуется версия «Проф». В базовой версии будет затруднительным даже добавление одного отчета к существующим, хотя перенастроить уже созданный отчет можно без проблем.

Формы, бланки, сценарии и часть картотек хранятся в текстовой форме, остальные картотеки выполнены в виде баз данных Paradox. Это решение обеспечивает легкое редактирование данных, но замедляет работу системы. Хотя средства аварийного восстановления баз данных Paradox существуют, в программу они не встроены, она просто аварийно завершается при повреждении баз данных, не пытаясь их восстановить. А вот поврежденные или удаленные индексы (вспомогательные данные, необходимые для поиска и сортировки информации) восстанавливаются полностью автоматически.

Весь интерфейс программы состоит из перенастраиваемых бланков, похожих на веб-страницы, работа с которыми интуитивно понятна и осваивается за считанные минуты. Ввести первичный документ, создать и откорректировать отчет, заполнить константы расчета — все эти операции станут привычными и простыми уже через пару дней использования (мы говорим о стандартном интерфейсе базовой версии). За работой с этим интерфейсом проходит большая часть времени, и здесь не возникает особых проблем, поскольку кроме бланков пользователь общается только с картотеками документов и древовидными списками аналитических призна-

ков. Их интерфейс достаточно лаконичен, поэтому на освоение навигации по картотеке уходит две-три минуты.

Хуже обстоит дело с вводом типовых операций или проводок. При вводе проводки необходимо указать дебетуемый и кредитуемый счета, сумму и содержание операции, а также указать аналитические признаки. При этом выбрать можно любой журнал и любой аналитический признак, привязав, например, товар со склада к выдаче заработной платы. Это решение кажется очень сомнительным — найти такую ошибку непросто. Кроме того, вместо имени аналитического объекта в проводке выводится его код. Вам нравится видеть «Ф.001» вместо «Иванов И.И.»?

Достоинства: простой интерфейс для ввода первичных документов и отчетов, простота освоения, мощный язык описания формул.

Недостатки: неудобный и неэргономичный ввод проводок и типовых операций, низкая скорость запуска и открытия документов, слабая система обеспечения целостности и резервного копирования данных.

Рекомендуется: для малых предприятий торговой сферы с большим количеством операций и простым учетом заработной платы.

«Финансы без проблем»

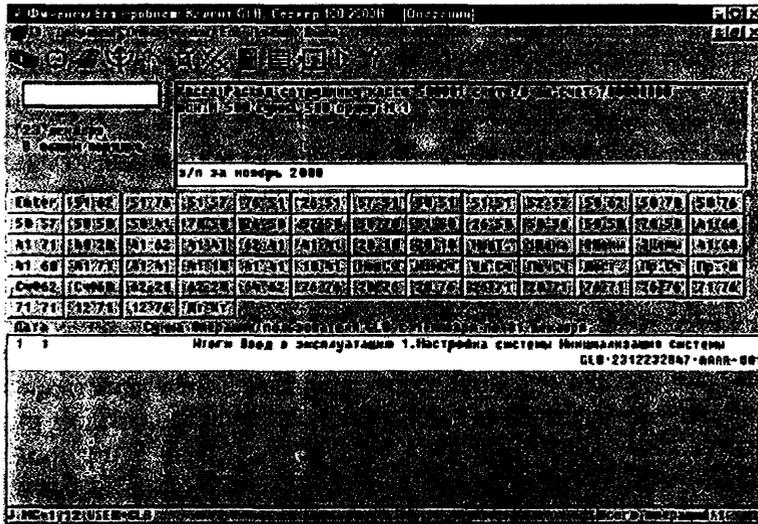


Рис. 16. Внешний вид программы

Производитель: «Хакерс Дизайн», «Кардинал-Софт».

Адрес в Интернете: www.cardinal-soft.com, www.hd.ru.

Функциональные модули: бухгалтерия, основные фонды, зарплата.

Системные требования: различаются для различных платформ, для Windows-версии — 20+ Мб на диске.

Стоимость однопользовательской версии: \$50.

Другие версии, дополнительные модули: многопользовательская версия, различные конфигурации.

Помните старую рекламу по телевидению — элегантного молодого человека, который делал баланс за пять минут? Это была реклама программы «Финансы без проблем» (ФБП) и едва ли не первый на российском телевидении ролик программного продукта. Система существует до сих пор, хотя ее вид и назначение заметно изменились. Точнее говоря, назначение осталось тем же — обеспечивать ведение бухгалтерского учета, а вот выполнена эта функция очень необычным образом.

Программа всегда поставляется в клиент-серверной версии. Только клиент и сервер не имеют громкого имени Microsoft, Oracle и т. п. и потому стоят существенно дешевле. А выполняют при этом гораздо больше функций. Обычный SQL-сервер обрабатывает запросы к базе данных и этим ограничивается. Клиентская часть должна выполнять достаточно большой объем работы по проведению специфических бухгалтерских вычислений и созданию отчетов. Поэтому и клиент, и сервер оказываются громоздкими приложениями, требовательными к ресурсам.

В системе ФБП применен совершенно иной подход. Специализированный бухгалтерский сервер понимает запросы вида «Выполнить типовую операцию», «Сделать проводку», «Построить баланс». Благодаря чему можно перенести всю тяжесть вычислительных операций на производительный сервер, в то время как клиент останется «сверхтонким» приложением, которое работает практически на любой платформе с минимальными требованиями к ресурсам. Серверы существуют для операционных систем Windows, Linux, Novell Netware, OS/2. Клиентская часть работает через DOS, Windows и даже через Интернет с помощью веб-браузера!

Из-за такой реализации клиент обладает весьма необычным многооконным интерфейсом, в котором поля ввода появляются не

все сразу, а по одному. Система довольно непривычна, но с ней можно освоиться, и работа будет достаточно быстрой.

На этом ядре реализованы самые разные модели учета, но наиболее доступными, пожалуй, являются реализации «ODA» и «L2» фирмы «Кардинал-Софт». Эти разработки включают около 70 типовых операций по банку, кассе, учету ТМЦ и основных средств, а также заработной плате. Помимо стандартных форм отчетности в органы финансового контроля, разработана масса полезных аналитических отчетов, общее число разных вариантов результатных документов превышает полторы сотни! Все отчеты могут быть отредактированы с помощью обычного текстового редактора. К сожалению, родной платформой для генератора отчетов являлась MS-DOS, поэтому средства оформления здесь ограничены.

Естественно, эта платформа достаточно универсальна для обеспечения учета в нескольких валютах и на многих предприятиях. Уже стандартная поставка подразумевает учет на трех предприятиях одновременно.

Достоинства: мультиплатформенность, высокая скорость работы, настраиваемость.

Недостатки: сложность начальной настройки и запуска в однопользовательском окружении, необычный интерфейс требует привыкания.

Рекомендуется: для средних и крупных предприятий, специфика которых требует использования многих рабочих мест в сети, выполненных на различных платформах.

«Янус»

Производитель: фирма «Порт».

Адрес в Интернете: www.port.obninsk.ru.

Функциональные модули: бухгалтерия, основные фонды, зарплата.

Системные требования: DOS, 8 Мб оперативной памяти, 9+ Мб на диске.

Стоимость однопользовательской версии: \$95 (одновалютная), \$115 (многовалютная).

Другие версии, дополнительные модули: сетевая версия, «склад», «зарплата и кадры».

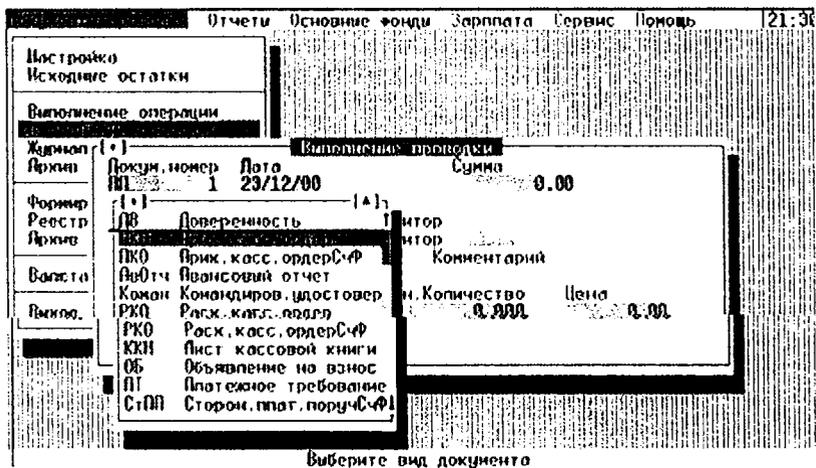


Рис. 17. Внешний вид программы

Редчайший на современном рынке программного обеспечения случай — система, которая разработана для MS DOS и которая по-прежнему поддерживается авторами. Более того, регулярно выходят новые версии. Потребителями этой продукции являются инертные пользователи, которые не хотят переходить на современные интерфейсы, а также владельцы старых компьютеров, на которых монстры типа 1С-бухгалтерии для Windows попросту не запустятся или же будут работать крайне медленно. К сожалению, в России достаточно много и тех и других, поэтому «Янус» и вошел в этот обзор.

Впрочем, среди DOS-программ с текстовым интерфейсом этот экземпляр смотрится выигрышно. Он укомплектован средствами редактирования типовых операций и справочников, довольно мощным редактором отчетных форм. Конечно, есть проблемы с расширяемостью системы, но связь с другими приложениями возможна уже потому, что все отчеты выводятся в текстовые файлы, где они могут быть отредактированы и переданы в другое приложение, например, в MS Excel.

Данные вводятся одним из трех стандартных способов — с помощью диалога заполнения проводок, мастера типовых операций или создания первичного документа. При этом достигается высокая скорость ввода, поскольку переходы между полями и появление выпадающих списков выбора происходят автоматически, и не надо

тянуться к мышке. В этом смысле многие DOS-программы по-прежнему остаются непревзойденными, поскольку стандартный Windows-интерфейс обычно не оптимизируется для быстрого заполнения форм.

Через отдельные меню «Основные фонды» и «Зарплата» вводятся первичные данные для соответствующих разделов учета. Ведомости по этим данным формируются там же — меню «Отчеты» предназначено только для внутренней, финансовой и редактируемой пользователем отчетности.

Система имеет многовалютную версию и поддерживает настраиваемые группы аналитики с вложенностью до третьего уровня. Этого вполне достаточно для бухгалтерского учета на малых и средних предприятиях. Работа программы в любом каталоге позволит легко организовать учет на нескольких предприятиях простым копированием всей папки с программой и данными.

Вычислительное ядро «Янус» имеет одну особенность, связанную с тем, что программа написана на языке Clipper. Пользователю предоставляется доступ к полному набору процедур и функций языка и программы, то есть возможности пользователя и разработчика равны. Платой за это является полностью английский интерфейс языка программирования.

База данных выполнена в стандартных файлах формата DBF, а готовые отчеты хранятся в текстовом виде в отдельном каталоге. Благодаря такому подходу аварийное восстановление данных будет довольно простой задачей. При каждом выходе из программы она предлагает выполнить архивацию данных. Эта настойчивость похвальна — сами пользователи обычно забывают о резервном копировании.

Разработка фирмы «Порт» имеет еще несколько «изюминок». В частности, системой поддерживается построение отчетов в графической форме, а функция «Секретарь» напомнит о том, что подходит срок сдачи того или иного отчета и перечисления денег в бюджет.

Достоинства: очень компактная, быстрая и надежная. Высокая эргономичность.

Недостатки: морально устаревший DOS-интерфейс, недостаточный для качественного оформления отчетов, слабые возможности по связи с другими приложениями, проблематична работа многопользовательской версии на большом количестве рабочих мест.

Рекомендуется: для малых и средних предприятий, на которых используется морально устаревшая компьютерная техника или очень высоки требования бухгалтера к эргономичности ввода.

Вопросы для самопроверки

100. Какие из приведенных программных продуктов не требуют наличия операционной системы Windows?

101. Опишите достоинства и недостатки приведенных выше программных продуктов.

ЗАКЛЮЧЕНИЕ

Представленный в конспекте лекций материал отражает в основном все вопросы программы дисциплины. Однако область эта является, во-первых, очень широкой и, во-вторых, достаточно динамичной. На рынке компьютерных продуктов постоянно появляются новые разработки экономического назначения или модернизируются существующие. Они отличаются как своими возможностями, так и сложностью, и исполнением. В настоящее время разработчики экономических программных продуктов все больше обращают внимание на расширение функциональных возможностей своих систем в области интеллектуальной обработки данных, поддержки принятия управленческих решений, мультимедиа, визуализации данных, методов общения человека с компьютером и других наукоемких технологий. Возрастают сложность и темпы программных проектов. В связи с этим все больше внимания уделяется технологиям индустриальной разработки, основанным на CASE-системах, технологиях RAD, использовании готовых компонентов.

Очень быстро растут и возможности аппаратного обеспечения. В торговых салонах конфигурация компьютера в среднем за полгода перемещается с верхней строчки прайс-листа на нижнюю. В общем, компьютерная индустрия переживает стремительные времена. Успеть за всем этим вряд ли получится, особенно, если профессионально занимаешься другими задачами. Однако знать об общих тенденциях, видах компьютерных продуктов и их возможностях, современных подходах в инженерии программирования будет полезно для специалистов любой области, где используются информационные технологии.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ

Тема: «Использование табличного редактора MS Excel и VBA в экономических расчетах»

Ситуация: В данной работе рассматривается следующая упрощенная модель: государство стремится увеличить сумму налоговых поступлений в бюджет; величина налоговых отчислений зависит от величины налоговой ставки ($TXRT$ — tax rate) и от величины прибыли предприятия (PRF — profit); величина прибыли, в свою очередь, зависит от рентабельности предприятия (Rn). Возникает вопрос, как при столь сложных связях установить величину налоговой ставки, чтобы получить максимум отчислений в бюджет.

Задание: Исследовать зависимость поступлений в бюджет от величины налоговой ставки и от рентабельности предприятия. Для исследований взять период в 10 лет. Найти и обосновать оптимальную величину налога. Поиск вести не программным способом, а визуально — по графикам зависимости.

Порядок выполнения работы

1. Получить исходные данные.

Как видно из сказанного выше, на величину поступления в бюджет оказывают влияние следующие параметры:

- ставка налога па прибыль ($TXRT$ — tax rate), %;
- начальный капитал предприятия (CP_0), руб;
- рентабельность предприятия (RN), %.

В данной работе для всех вариантов заданий расчет провести для значений $TXRT$ от 10% до 80% с шагом в 10%. Значения прочих исходных данных приведены в таблице 3. Значения рентабельности выбирать из указанного диапазона с шагом в 5% (для всех вариантов).

Таблица 3.

Исходные данные для расчета

№ варианта	Последняя цифра номера зачетной книжки	СР ₀	Диапазон значений RN.
1	1, 2	100	10%...30%
2	3, 4	200	30%...50%
3	5, 6	300	50%...70%
4	7, 8	400	70%...90%
5	9, 0	500	90%...110%

2. Создать пользовательскую функцию для расчета поступлений в бюджет.

Алгоритм вычислений, проводимых этой функцией:

Для каждого года исследуемого периода (то есть 10 раз) выполняются следующие шаги.

Шаг 1. Вычисляется величина доналоговой прибыли по формуле:

$$PRF = CP \cdot RN$$

где CP — величина капитала предприятия в предыдущем году (в первом году она равна CP_0).

Шаг 2. Вычисляется величина налогового отчисления:

$$BD = PRF \cdot TXRT$$

Шаг 3. Вычислить величину капитала предприятия в текущем году:

$$CP = CP + PRF - BD$$

Шаг 4. Увеличить общий объем отчислений в бюджет BD_t на величину отчислений за этот год (см. рис.18).

Для реализации этого алгоритма необходимо:

- создать рабочую книгу MS Excel с названием «Test.xls»;
 - запустить редактор VBA и создать новый модуль, как показано в восьмой главе конспекта лекций;
 - в окне модуля напечатать текст программы, согласно рис. 18.
- Этим текстом определяется пользовательская функция Excel. При помощи этой функции будет рассчитана и заполнена таблица в Excel.

```

Книга1 - Module1 (Code)
(General) BDT
Function BDT(Срo, Rn, TXRT, t)
Dim PRF As Single ' текущая прибыль
Dim CP As Single ' текущий капитал
Dim BD As Single ' текущее поступление в бюджет
Dim i As Integer ' текущий год
' расчет
CP = Срo
BDt = 0
For i = 1 To t
PRF = CP * Rn
BD = PRF * TXRT
CP = CP + PRF
BDt = BDt + BD
Next i
End Function

```

Рис. 18. Пользовательская функция для расчета.

Расчет строится на основе приведенных выше формул, а повторение вычислений обеспечивается оператором цикла *For Next*

3. Закрывать редактор VBA и вернуться в рабочий лист книги «Test.xls». На рабочем листе создать таблицу так, как показано на рис. 19.

	A	B	C	D	E	F	G
1							
2		CРo = 100					
3		TXRT \ Rn	10%	15%	20%	25%	30%
4		10%					
5		20%					
6		30%					
7		40%					
8		50%					
9		60%					
10		70%					
11		80%					

Рис. 19. Образец заполнения рабочего листа.

4. Выполнить необходимые расчеты отчислений в бюджет, заполнив таблицу. Для этого нужно использовать созданную в п.2 пользовательскую функцию BDt (см. также конспект лекций).

5. По полученным данным построить диаграмму (для построения диаграммы использовать мастер диаграмм Excel). Пример результата для диапазона рентабельности 60% — 120% представлен на рис. 20.

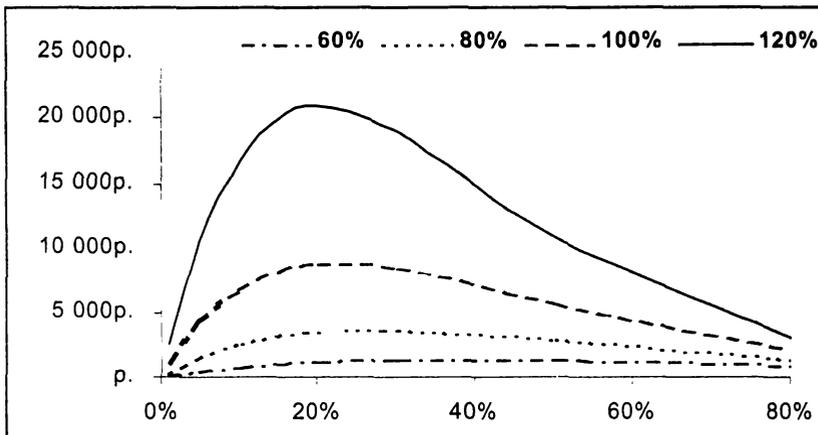


Рис. 20. Зависимость поступлений в бюджет от величины налога и рентабельности

6. Провести анализ результатов по полученным графикам. Ответить на следующие вопросы: как влияет на отчисления в бюджет увеличение рентабельности? Какие фирмы (с высокой или низкой рентабельностью) целесообразнее облагать высоким налогом?

6. Используя поиск решения, найти для различных значений рентабельности оптимальное значение налоговой ставки.

На новом листе рабочей книги «test.xls» MS Excel построить таблицу зависимости поступления в бюджет при оптимальной ставке налога, что подразумевает максимальную величину BDt от рентабельности (см табл. 4). Начальный капитал предприятия считать равным 100 руб. Поиск оптимальной ставки налога вести с помощью программы Excel Поиск решений (см Приложение). В качестве целевых ячеек выбирать ячейки из столбца $BDt(Rn)$; изменяемые ячейки — из столбца $TXRT$. При условии правильного выполнения поиска в ячейках столбца $TXRT$ будут получены значения оптимальной ставки налога.

Таблица 4.

**Зависимость поступлений в бюджет от рентабельности
(при оптимальной ставке налога)**

Сро	Rn	TXRT,%	BDt(Rn)
100	30%		
100	60%		
100	90%		
100	120%		

Оформление отчета

Отчет представляется на бумажном носителе и должен содержать название работы, цель, распечатку исходных данных, таблиц, диаграмм и текстов созданных на VBA функций.

Критерии оценки

— законченность работы: выполнены все поставленные задачи, построены все необходимые диаграммы и таблицы;

— аккуратность исполнения и оформления результатов работы (читабельность и убедительность).

ВОПРОСЫ ТЕСТА ДЛЯ ИТОГОВОГО КОНТРОЛЯ

1. Аббревиатура ПК расшифровывается как:
 - a) переносной компьютер;
 - b) персональный компьютер;
 - c) пользовательский компьютер.
2. Какой компьютер американцы называют первым ПК?
 - a) Альтаир-8800;
 - b) Apple-2;
 - c) IBM PC-286.
3. В каком году была представлена конструкция IBM PC XT?
 - a) 1981;
 - b) 1968;
 - c) 1973.
4. Какой процессор применялся в первом ПК конструкции IBM PC XT?
 - a) Intel i8088;
 - b) Intel Pentium;
 - c) ZX-Spectrum 486.
5. «Открытая архитектура компьютера IBM PC» — это:
 - a) особая конфигурация компьютера, при которой все внутренние устройства машины видны пользователю;
 - b) возможность организовывать работу нескольких персональных компьютеров в локальной сети;
 - c) возможность расширять архитектуру (конструкцию) компьютера путем подключения дополнительных блоков через специальные устройства подключения.
6. Выбрать правильный ответ. BIOS — это:
 - a) обозначение протоколов передачи данных в сети Интернет;
 - b) обозначение оперативной памяти компьютера;
 - c) обозначение базовой системы ввода-вывода — программы, с которой начинается запуск компьютера.

7. Что понимается под компьютерными продуктами?
- a) аппаратные и программные средства компьютера как объекты производства и коммерческой деятельности;
 - b) совокупность программных средств, их документации, гарантии качества, рекламных материалов, мер по обучению пользователей, распространению и сопровождению готового программного обеспечения;
 - c) совокупность характеристик, методов и средств разработки приложений, которые обеспечивают наиболее быстрое создание прототипа приложения.
8. Что понимается под программным продуктом?
- a) аппаратные и программные средства компьютера как объекты производства и коммерческой деятельности;
 - b) совокупность программных средств, их документации, гарантии качества, рекламных материалов, мер по обучению пользователей, распространению и сопровождению готового программного обеспечения;
 - c) совокупность характеристик, методов и средств разработки приложений, которые обеспечивают наиболее быстрое создание прототипа приложения.
9. Выбрать правильный ответ. Аббревиатура ПО означает:
- a) программа отладки;
 - b) программное обеспечение;
 - c) программное оборудование.

Т.1. Классификация персональных компьютеров

10. Кто является разработчиком спецификации РСхх персональных компьютеров?
- a) Intel и Microsoft;
 - b) IBM и Intel;
 - c) Apple и Intel.
11. Выбрать ответ, в котором наиболее полно и правильно перечислены классы ПК (в соответствии со спецификацией РС99):
- a) пользовательский, офисный, рабочая станция, игровой или развлекательный, мобильный компьютер;
 - b) рабочая станция, офисный и игровые ПК;
 - c) персональный, многопользовательский, универсальный, специализированный;
 - d) Pentium, AMD, Cuyrix.

- 12.** Выбрать правильный ответ. В соответствии со спецификацией PC99 Entertainment PC — это:
- a) рабочая станция;
 - b) развлекательный компьютер;
 - c) мультимедийный компьютер.
- 13.** Какая фирма выдает сертификат соответствия аппаратуры требованиям спецификации PC99, который разрешает использование на ПК логотипа Windows?
- a) Microsoft;
 - b) Intel;
 - c) AMD;
 - d) IBM.

T.2. Блоки обработки информации

- 14.** Материнская плата ПК — это:
- a) плата-разъем для подсоединения всех периферийных устройств ПК, а также системной платы компьютера;
 - b) центральное процессорное устройство;
 - c) системная плата, на которой размещается процессор.
- 15.** Выбрать тот вариант ответа, где приведены элементы, которые НЕ располагаются на системной плате ПК.
- a) микросхема базовой системы ввода-вывода BIOS и аккумулятор для ее питания;
 - b) блок питания компьютера;
 - c) слоты расширения, разъемы для подключения интерфейсных кабелей.
- 16.** В персональном компьютере объединяющим элементом, который обеспечивает соединение разных частей и устройств в единую систему, является:
- a) базовая система ввода-вывода BIOS;
 - b) материнская (системная) плата;
 - c) слоты расширения, разъемы для подключения интерфейсных кабелей.
- 17.** Кэш-память это:
- a) любое внешнее запоминающее устройство, подключаемое к ПК с помощью специального разъема;
 - b) запоминающее устройство на жестком диске;
 - c) сверхоперативное запоминающее устройство.

- 18.** Mainboard — это:
- а) системная плата;
 - б) название конструкции ПК, не получившей дальнейшего развития;
 - с) один из вариантов названия системного блока.
- 19.** Кэш-память применяется для:
- а) ускорения работы ПК;
 - б) выполнения математических расчетов;
 - с) управления дисковыми.
- 20.** Выбрать наиболее полный и правильный ответ. SCSI-адаптер (разъем) применяется:
- а) для ускорения работы с графикой на ПК;
 - б) для подключения внешнего факс-модема, а также других внешних устройств и аппаратуры;
 - с) только для подключения внешнего факс-модема.
- 21.** Слот (slot) и сокет (Socket) в ПК — это:
- а) два вида процессорных разъемов на материнской плате;
 - б) сверхоперативное запоминающее устройство;
 - с) два вида модулей оперативных запоминающих устройств;
- 22.** Какая из приведенных ниже аббревиатур обозначает процессор компьютера?
- а) USB;
 - б) CPU;
 - с) LPT.
- 23.** Какая из приведенных ниже аббревиатур является обозначением порта ПК, к которому подключается принтер?
- а) USB;
 - б) CPU;
 - с) LPT.
- 24.** Какая из приведенных ниже аббревиатур является обозначением порта ПК, к которому подключаются внешние устройства (накопитель, сканер и т. п.)?
- а) USB;
 - б) CPU;
 - с) LPT.
- 25.** Основной характеристикой процессора, влияющего на производительность ПК, является:
- а) скорость вращения диска;

- b) степень интеграции чипа и тактовая частота;
 - c) объем памяти оперативного запоминающего устройства.
- 26.** Основной характеристикой накопителя на жестких дисках, влияющего на производительность ПК, является:
- a) скорость вращения и объем памяти диска;
 - b) способ адресации памяти и число дисков;
 - c) степень интеграции чипа и тактовая частота;
 - d) объем памяти оперативного запоминающего устройства.
- 27.** Основной характеристикой оперативно-запоминающего устройства, влияющего на производительность ПК, является:
- a) частота обращения к сегментам памяти;
 - b) способ адресации памяти;
 - c) степень интеграции чипа и тактовая частота;
 - d) объем памяти.
- 28.** Как назывались процессоры первых ПК IBM PC XT?
- a) K7-Athlon;
 - b) Intel 8088;
 - c) Z-80.
- 29.** Какая фирма выпускает процессоры Celeron?
- a) фирма AMD;
 - b) фирма Intel;
 - c) фирма Apple.
- 30.** Какая из перечисленных ниже фирм является соперником Intel в производстве процессоров для IBM-совместимых компьютеров?
- a) Apple;
 - b) AMD;
 - c) Samsung.
- 31.** Назовите английский аналог русской аббревиатуры ОЗУ.
- a) CD ROM;
 - b) RAM;
 - c) SDRAM.
- 32.** Выбрать вариант, в котором назван термин, НЕ имеющий отношения к оперативной памяти ПК:
- a) DIMM;
 - b) SIMM;
 - c) SCSI.

- 33.** Аббревиатуры SIMM, DIMM, DDR RAM обозначают:
- а) различные модули оперативной памяти, отличающиеся числом и расположением контактов для подключения микросхем памяти;
 - б) различные виды процессоров, отличающихся производительностью;
 - с) различные виды накопителей на гибких дисках, отличающихся скоростью вращения дисков.

Т.3. Блоки хранения информации

- 34.** Термин «Винчестер» обозначает:
- а) накопитель на жестких дисках для записи и хранения информации;
 - б) устройство для воспроизведения и записи компакт-дисков;
 - с) накопитель на гибких магнитных дисках.
- 35.** Аббревиатура CD ROM обозначает:
- а) накопитель на жестких дисках для записи и хранения информации;
 - б) устройство (дисковод) для воспроизведения компакт-дисков;
 - с) накопитель на гибких магнитных дисках.
- 36.** Аббревиатура CD-RW обозначает:
- а) устройство (дисковод) для воспроизведения и записи компакт-дисков;
 - б) устройство (дисковод) только для воспроизведения компакт-дисков;
 - с) магнитооптический накопитель.
- 37.** Выбрать правильный вариант ответа. В стандартной комплектации ПК символами «А:» или «В:» обозначаются следующие дисководы:
- а) дисководы флоппи дисков (FDD);
 - б) дисководы жестких дисков;
 - с) дисководы компакт-дисков.
- 38.** Выбрать вариант, где указано устройство, не относящееся к съемным накопителям информации в ПК.
- а) ленточный накопитель — стример;
 - б) флэш-память;
 - с) оперативное запоминающее устройство.

Т.4. Подключение внешних устройств

- 39.** Термин Plug&Play применяется для обозначения:
- а) технологии подключения внешних устройств ПК, при которой все необходимые настройки оборудования делаются пользователем вручную;
 - б) технологии подключения внешних устройств ПК, при которой операционная система самостоятельно выполняет необходимые настройки для обеспечения работы подключаемых устройств;
 - с) технологии инсталляции игрового программного обеспечения.
- 40.** Необходимым условием для автоматической настройки параметров системы при подключении внешних устройств является:
- а) наличие в ПК других, ранее подключенных устройств;
 - б) поддержка подключаемым устройством технологии Plug&Play;
 - с) подключение устройства к шине расширения ISA.
- 41.** Термином COM обозначается:
- а) параллельный порт для подключения принтера;
 - б) любое подключаемое устройство, которое поддерживает технологию Plug&Play;
 - с) коммуникационный (последовательный) порт ПК для подключения некоторых внешних устройств и связи компьютеров между собой.
- 42.** Термином LPT обозначается:
- а) параллельный порт для подключения принтера;
 - б) любое подключаемое устройство, которое поддерживает технологию Plug&Play;
 - с) коммуникационный (последовательный) порт ПК для подключения некоторых внешних устройств и связи компьютеров между собой.
- 43.** Термином USB обозначается:
- а) любое подключаемое устройство, которое поддерживает технологию Plug&Play;
 - б) быстрый интерфейс для «горячего» подключения внешних устройств (лазерный принтер, сканер, цифровая камера, переносной винчестер и т. д.);
 - с) интерфейс для подключения внешних устройств через последовательный порт ПК.

44. Среди таких шин (интерфейсов) для подключения внешних устройств, как ISA, PCI, USB, наиболее устаревшей является:
- а) PCI;
 - б) USB;
 - в) ISA.

Т.5. Определение и классификация программных продуктов

45. Выберите правильный вариант ответа. По сфере применения программное обеспечение ПК делится на два класса:
- а) вертикальное и горизонтальное;
 - б) локальное и глобальное;
 - в) системное и прикладное.
46. Вертикальное программное приложение — это:
- а) «заказное» приложение, разработка которого финансируется заказчиком и которое предназначено для нужд данного заказчика;
 - б) «коробочное» приложение, разработка которого финансируется производителем и которое предназначено для последующей продажи различным пользователем;
 - в) программное приложение, предназначенное для автоматизации деятельности многоуровневых систем управления.
47. Горизонтальное программное приложение — это:
- а) «заказное» приложение, разработка которого финансируется заказчиком, предназначенное для нужд данного заказчика;
 - б) «коробочное» приложение, разработка которого финансируется производителем, предназначено для последующей продажи различным пользователем;
 - в) программное приложение, предназначенное для автоматизации деятельности многоуровневых систем управления.
48. Программный продукт — это:
- а) совокупность программных и аппаратных средств, их документации, гарантии качества, рекламных материалов, мер по обучению пользователей, распространению и сопровождению готового обеспечения;
 - б) совокупность отдельных программных средств, их документации, гарантии качества, рекламных материалов, мер

- по обучению пользователей, распространению и сопровождению готового программного обеспечения;
- с) совокупность аппаратных средств, их документации, гарантии качества, рекламных материалов, мер по обучению пользователей, распространению и сопровождению готового обеспечения.
- 49.** Программный продукт, который предназначен для решения задач обработки экономической информации на конкретном предприятии относится к классу:
- а) системного программного обеспечения;
 - б) прикладного программного обеспечения;
 - с) инструментальных средств программирования.
- 50.** Выберите наиболее полный и правильный вариант. В зависимости от масштаба выделяют следующие градации программных приложений:
- а) прикладные и системные;
 - б) заказные и коробочные;
 - с) настольные, групповые, масштаба предприятия;
 - д) настольные, групповые, масштаба отдела, масштаба предприятия.
- 51.** Настольное программное приложение — это:
- а) приложение, которое имеет, как правило, одного пользователя, может выполняться на одном компьютере и работать автономно, без сети;
 - б) приложение для нескольких пользователей, которые могут работать с ним порознь или одновременно, но все принадлежат к одной группе (или категории) пользователей;
 - с) любое программное приложение, ярлык которого вынесен на рабочий стол компьютер.
- 52.** Групповое программное приложение — это:
- а) приложение для нескольких пользователей, которые могут работать с ним порознь или одновременно, но все принадлежат к одной группе (или категории) пользователей;
 - б) приложение, которое имеет значительное количество пользователей, принадлежащих к различным группам, выполняет несколько различных связанных между собой функций и работает в локальной или глобальной сети;
 - с) любое программное приложение, если оно скопировано в различные группы папок на компьютере.

53. Для обеспечения совместной работы многих пользователей разных категорий при выполнении ими различных взаимосвязанных функций используется:

- а) программное приложение масштаба предприятия;
- б) настольное программное приложение, работающее в локальной или глобальной сети;
- в) групповое программное приложение.

Т.6. Стратегия разработки программного обеспечения

54. Программная инженерия — это:

- а) совокупность инструментальных средств для управления программным проектом;
- б) совокупность инженерных методов и средств создания программного обеспечения;
- в) процесс разработки алгоритмов и отладки программного обеспечения.

55. Жизненный цикл программного обеспечения:

- а) период, который начинается с момента покупки ПО и заканчивается в момент его полного изъятия из эксплуатации;
- б) период, который начинается с момента принятия решения о необходимости создания ПО и заканчивается в момент его полного изъятия из эксплуатации;
- в) период использования ПО по назначению.

56. В дисциплине «Компьютерные продукты» аббревиатура ЖЦ ПО обозначает:

- а) жизненный цикл предметной области;
- б) жизненный цикл программного обеспечения;
- в) жизненная цель программы-отладчика.

57. К какой группе процессов жизненного цикла ПО относится процесс его разработки?

- а) основному;
- б) вспомогательному;
- в) организационному.

58. К какой группе процессов жизненного цикла ПО относятся процессы его приобретения и поставки?

- а) основному;
- б) вспомогательному;
- в) организационному.

- 59.** Выбрать вариант, в котором наиболее полно и правильно перечислены группы процессов в жизненном цикле ПО (в соответствии с классификацией по стандарту ISO):
- а) основные, вспомогательные, организационные процессы;
 - б) процессы разработки, покупки; эксплуатации;
 - с) приобретение, сопровождение, разработка, эксплуатация, утилизация.
- 60.** Выбрать вариант, в котором указаны процессы ЖЦ ПО, не входящие в группу основных:
- а) приобретение;
 - б) сопровождение;
 - с) верификация.
- 61.** Выбрать вариант, в котором указаны процессы ЖЦ ПО, не входящие в группу вспомогательных:
- а) документирование;
 - б) создание инфраструктуры проекта;
 - с) аудит.
- 62.** Выбрать вариант, в котором указаны процессы ЖЦ ПО, входящие в группу организационных:
- а) документирование, обеспечение качества, верификация, аттестация, оценка, аудит;
 - б) управление проектом, создание инфраструктуры проекта, обучение пользователей;
 - с) приобретение, поставка, разработка, эксплуатация, сопровождение.
- 63.** Выбрать вариант, в котором указаны процессы ЖЦ ПО, входящие в группу основных процессов:
- а) документирование, обеспечение качества, верификация, аттестация, оценка, аудит;
 - б) управление проектом, создание инфраструктуры проекта, обучение пользователей;
 - с) приобретение, поставка, разработка, эксплуатация, сопровождение.
- 64.** Выбрать вариант, в котором указаны процессы ЖЦ ПО, входящие в группу вспомогательных процессов:
- а) документирование, обеспечение качества, верификация, аттестация, оценка, аудит;

- b) управление проектом, создание инфраструктуры проекта, обучение пользователей;
 - c) приобретение, поставка, разработка, эксплуатация, сопровождение.
- 65.** Такое определение процесса как «внесение изменений в ПО в целях исправления ошибок, повышения производительности или адаптации к изменившимся условиям работы или требованиям» относится к следующему процессу ЖЦ ПО:
- a) разработки;
 - b) верификации;
 - c) сопровождения;
 - d) управления конфигурацией.
- 66.** Определение соответствия требованиям, планам и условиям договора на разработку (поставку) ПО выполняется в следующем процессе ЖЦ ПО:
- a) разработки;
 - b) аудита;
 - c) сопровождения;
 - d) управления конфигурацией.
- 67.** Структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач, выполняемых на протяжении ЖЦ, называется:
- a) моделью ЖЦ;
 - b) инфраструктурой ЖЦ;
 - c) ЖЦ программного обеспечения.
- 68.** Назовите две основные модели ЖЦ:
- a) каскадная модель, спиральная модель;
 - b) внутренняя модель, внешняя модель;
 - c) базовая модель, вспомогательная модель.
- 69.** Каскадная модель жизненного цикла ПО предполагает:
- a) разбиение всей разработки ПО на этапы, причем переход с одного этапа на следующий происходит только после того, как будет полностью завершена работа на текущем;
 - b) разбиение всей разработки на этапы и параллельное (одновременное) их выполнение коллективом разработчиков;
 - c) циклический процесс, связанный с анализом, проектированием, последовательным уточнением требований и разработкой прототипов будущего ПО.

- 70.** Спиральная модель жизненного цикла ПО предполагает:
- a) разбиение всей разработки ПО на этапы, причем переход с одного этапа на следующий происходит только после того, как будет полностью завершена работа на текущем;
 - b) разбиение всей разработки на этапы и параллельное (одновременное) их выполнение коллективом разработчиков;
 - c) циклический процесс, связанный с анализом, проектированием, последовательным уточнением требований и разработкой прототипов будущего ПО.
- 71.** Что такое CASE-средства?
- a) программные средства, поддерживающие процессы жизненного цикла ПО;
 - b) программные средства для экономического анализа;
 - c) программные комплексы для тестирования компьютерных систем.
- 72.** Применение CASE-средств позволяет:
- a) устанавливать новое программное обеспечение на компьютер с автоматической настройкой параметров системы;
 - b) подключать внешние устройства ПК к шине USB по технологии Plug&Play;
 - c) автоматизировать выполнение процессов ЖЦ ПО (документирование требований, анализ области применения, проектирование, разработка и т. д.).

Т.7. Технология быстрой разработки приложений — RAD

- 73.** Термин RAD применяется для обозначения:
- a) технологии подключения внешних устройств, при которой настройка параметров системы производится автоматически;
 - b) технологии управления программным проектом;
 - c) технологии быстрой разработки программных приложений.
- 74.** Выбрать ответ, в котором наиболее полно отражается особенность технологии RAD. В технологии RAD для быстрой разработки приложений:
- a) применяются развитые средства визуального программирования, основанные на использовании готовых фрагментов программ;
 - b) привлекается большой коллектив профессиональных разработчиков (аналитиков, разработчиков, менеджеров);
 - c) применяются методы объектно-ориентированного программирования.

- 75.** Подход RAD предусматривает группу разработчиков из:
- a) 1-2 человек;
 - b) 30-50 человек;
 - c) 3-7 человек.
- 76.** Выбрать вариант, в котором приведены средства программирования технологии RAD:
- a) Visual C++, Visual Basic, Delphi;
 - b) Pascal, C, Prolog;
 - c) Simulinc, Matlab, Maple.
- 77.** Что из перечисленного не относится к инструментальным средам разработки ПО:
- a) Visual C++;
 - b) Visio;
 - c) Visual Basic.
- 78.** Для создания офисных приложений в пакет MS Office интегрирован следующий язык визуального программирования:
- a) Visual C++;
 - b) Visual Basic;
 - c) Visual Basic for Applications (VBA).
- 79.** Какая компания является создателем и активно продвигает VBA?
- a) Apple;
 - b) Microsoft;
 - c) Borland;
 - d) Basic Inc.
- 80.** Можно ли запустить VBA непосредственно из MS Excel?
- a) нет, для запуска требуется отдельное самостоятельное приложение VBA;
 - b) да, через меню «Сервис» или с помощью комбинации клавиш;
 - c) да, только через меню «Сервис».
- 81.** Для запуска редактора VBA из MS Excel используется комбинация клавиш:
- a) Alt+I;
 - b) Alt+F11;
 - c) Ctrl+ Alt + Del.
- 82.** Со служебного слова Sub в VBA начинается:
- a) описание переменных;
 - b) описание процедуры;
 - c) описание подстроки.

- 83.** Функция отличается от процедуры:
- a) количеством переменных;
 - b) тем, что функцией возвращается некоторое значение;
 - c) способом описания переменных;
 - d) ничем не отличается.

Т. 8. Элементы языка VBA

- 84.** Термин «Переменная» в языке программирования — это:
- a) имя, определяющее область памяти для хранения некоторой величины, изменяющейся при работе программы;
 - b) подпрограмма, которая после выполнения некоторых установленных действий возвращает определенное значение;
 - c) имя, определяющее область памяти для хранения некоторой константы.
- 85.** Выбрать вариант, где правильно указаны символы, которые нельзя использовать в названиях переменных в VBA:
- a) буквы, цифры и знак подчеркивания;
 - b) 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 и пробел;
 - c) «.», «!», «@», «&», «\$», «#» и пробел.
- 86.** Выбрать вариант, где правильно объявлена переменная в процедуре языка VBA:
- a) Dim Count As Integer
 - b) Var Count: Integer;
 - c) Sub Count As Integer.
- 87.** Выбрать вариант, где на языке VBA дано объявление двумерного массива целых чисел:
- a) Dim A(2, 2, 2) As Integer
 - b) Dim A(3,3) As String
 - c) Dim A(3,3) As Integer.
- 88.** Объявление переменных в VBA:
- a) обязательно;
 - b) не обязательно;
 - c) не используется.
- 89.** Выбрать вариант, где на языке VBA дано правильное объявление константы:
- a) Const NameConst As Integer;
 - b) Dim NameConst As Integer = 10;
 - c) Const NameConst As Integer = 10.

- 90.** Массив — это:
- a) множество возможных значений некоторой переменной;
 - b) упорядоченная группа однотипных переменных, которые объединены одним общим именем;
 - c) совокупность всех описанных в процедуре переменных.
- 91.** Выбрать вариант, в котором правильно указаны отличия массива от переменной:
- a) в массиве, в отличие от переменной, можно хранить одновременно разные значения;
 - b) массив применяется для хранения постоянного (неизменного) значения, а переменная — для хранения изменяемого значения;
 - c) массив используется только в функциях и не может быть использован в процедурах.
- 92.** Выбрать вариант, в котором приведен оператор VBA — «возведение в степень»:
- a) «^»;
 - b) «*»;
 - c) «\».
- 93.** Ниже используются две переменные целого типа — X и Y. Выбрать вариант процедуры VBA, где правильно записано условие «Если X меньше или равно Y, то нужно X умножить на 2, иначе — X разделить на 2»:
- a) If X <=Y Then X = X*2 X = X/2;
 - b) If X <= Y Then X *2 Else X/2;
 - c) If X <=Y Then X = X*2 Else X = X/2.
- 94.** Какой из приведенных ниже операторов является оператором цикла, позволяющим выполнять циклические вычисления заданное число раз?
- a) While...Wend;
 - b) For...Next;
 - c) Do...Loop.
- 95.** Какой из приведенных ниже операторов является оператором цикла, позволяющим выполнять циклические вычисления, пока истинно некоторое заданное условие?
- a) While...Wend;
 - b) For...Next;
 - c) If Then Else.

- 96.** Оператор For...Next выполняется:
- a) заданное число раз;
 - b) пока истинно заданное пользователем условие;
 - c) пока не выполнится заданное пользователем условие.
- 97.** Оператор While...Wend выполняется:
- a) заданное число раз;
 - b) пока истинно заданное пользователем условие;
 - c) пока не выполнится заданное пользователем условие.
- 98.** Оператор For Each...Next:
- a) выполняется заданное число раз;
 - b) выполняется, пока истинно заданное условие;
 - c) повторяет выполнение группы операторов (инструкций) для каждого элемента заданного массива.

Т.9. Практика использования VBA — объекты MS Excel

- 99.** Объект в VBA — это:
- a) объединение данных с методами (кодом) для их обработки;
 - b) стандартный набор операций над данными;
 - c) набор организованных некоторым образом переменных.
- 100.** Метод объекта — это:
- a) атрибут объекта, определяющий его характеристики, такие как размер, цвет, положение на экране, или состояние объекта;
 - b) действие, выполняемое над объектом;
 - c) любая подпрограмма VBA, в которой происходит обращение к данному объекту.
- 101.** Свойство объекта — это:
- a) атрибут объекта, определяющий его характеристики, такие как размер, цвет, положение на экране, или состояние объекта;
 - b) действие, выполняемое над объектом;
 - c) любая подпрограмма VBA, в которой происходит обращение к данному объекту.

Т.10. Программное обеспечение для решения финансово-экономических задач

102. Какой из приведенных ниже программных продуктов не требует для работы наличия операционной системы Windows?

- a) «Турбо-Бухгалтер»;
- b) «1С:Бухгалтерия»;
- c) «Янус».

103. Имеются ли возможности поставки системы «1С:Бухгалтерия» заказчику в той или иной конфигурации?

- a) да;
- b) нет;
- c) не знаю.

ГЛОССАРИЙ

Алгоритм — совокупность действий со строго определенными правилами выполнения.

Алгоритмизация — составление алгоритмов для решения поставленных задач.

Алфавит языка программирования — совокупность допустимых символов языка.

Архитектура клиент-сервер — способ организации взаимодействия программ или компонент многокомпонентной программы. А именно, имеется программа или компонента программы, называемая *сервером*, и одна или несколько других программ или компонент, называемых *клиентами*. Клиент имеет возможность асинхронно для сервера инициировать выполнение процедур сервера и получать результаты их выполнения.

База данных (БД) — поименованная, целостная, единая система данных, организованная по определенным правилам, которые предусматривают общие принципы описания, хранения и обработки данных.

Базовое приложение (для офисного приложения) — то офисное средство разработки, которое использовано для разработки данного приложения.

Быстрая разработка приложений — обобщающее название для совокупности характеристик методов и средства разработки приложений, которые обеспечивают наиболее быстрое создание прототипа приложения.

Вертикальное приложение — приложение, которое сделано по конкретному индивидуальному заказу.

Века — одномоментное идентифицируемое событие в процессе разработки, сопровождающееся появлением и фиксацией некоторого отчуждаемого материала (документа, программы, и др.).

Визуальный язык программирования — язык программирования, при использовании которого взаимодействие пользователя с системой программирования при составлении и отладке программ реализуется диалоговыми средствами графического интерфейса пользователя. См. *графический интерфейс пользователя*.

Влияющие ячейки — ячейки рабочего листа, значения которых используются формулой в данной ячейке.

Вложенный цикл — оператор цикла, который находится внутри тела другого оператора цикла.

Внедрение приложения — процесс перевода офисного приложения из состояния разработки в состояние эксплуатации.

Вычислительная сеть — совокупность ЭВМ, объединенных сетью связи, позволяющей компьютерам проводить обмен информацией.

Глобальная сеть — сеть, в которой объединены компьютеры в различных странах, на различных континентах.

Горизонтальное приложение — приложение, которое рассчитано на массовый спрос и которое можно купить в магазине.

Графический интерфейс пользователя — интерфейс пользователя, основанный на средствах машинной графики.

Групповое приложение — приложение, с которым работает группа пользователей.

Деловая графика — графические образы, часто используемые в офисных приложениях.

Диалоговое окно — один из основных элементов организации интерфейса приложения.

Жизненный цикл программного обеспечения — период, который начинается с момента принятия решения о необходимости создания ПО и заканчивается в момент его полного изъятия из эксплуатации.

Заказчик — субъект (отдельное лицо или организация), заказывающий разработку офисного приложения.

Идентификатор — последовательность символов, первый символ которой обязан быть буквой.

Интеграция (в системе или систем) — восстановление и (или) повышение качественного уровня взаимосвязей между элементами системы, а также процесс создания из нескольких разнородных систем единой системы, с целью исключения (до технически необходимого минимума) функциональной и структурной избыточности и повышения общей эффективности функционирования.

Интерфейс пользователя — способ и средства взаимодействия пользователя с приложением.

Информационные процессы — процессы сбора, обработки, накопления, хранения, поиска и распространения информации.

Класс — множество объектов, связанных общностью структуры и поведения. Любой объект является экземпляром класса.

Компьютерные продукты — аппаратные и программные средства компьютера как объекты производства и коммерческой деятельности.

Константа — элемент данных, не меняющий своего значения.

Контроллер — устройство управления.

Конфигурация ПО — совокупность его функциональных и физических характеристик, установленных в технической документации и реализованных в ПО.

Коэффициент расширения — коэффициент, который определяет, как в среднем относится количество исполнимых машинных команд после перевода программы на машинный язык к количеству операторов в исходной программе. Если этот коэффициент близок к 1, то язык считают языком низкого уровня, а если существенно больше 1, то язык считают языком высокого уровня. Например, VBA — это язык высокого уровня, а ассемблер — низкого.

Кэш — сверхоперативное запоминающее устройство; реализует быстродействующую буферную память между процессором и основной памятью.

Локальная вычислительная сеть — сеть, объединяющая компьютеры в комнате или соседних помещениях.

Магнитные диски — средство хранения информации в ЭВМ.

Массив — совокупность однотипных индексированных переменных.

Масштаб приложения — качественная оценка размера приложения с точки зрения его использования. См. *настольное приложение, групповое приложение, приложение масштаба предприятия*.

Метод объекта — процедура или функция, которая позволяет работать с данными объекта.

Метод проектирования ПО — организованная совокупность процессов создания ряда моделей, которые описывают различные аспекты разрабатываемой системы с использованием четко определенной нотации.

Модель — абстрактное описание свойств, структуры, связей и закономерностей функционирования реальной или вымышленной системы.

Модель (ПО) — полное описание системы ПО с определенной точки зрения.

Модель ЖЦ ПО — структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач на протяжении ЖЦ.

Модуль — часть программы, оформленная в виде, допускающем ее независимую трансляцию.

Модульное программирование — способ разработки программ, при котором программа разбивается на относительно независимые составные части — программные модули.

Мультимедийные средства — интерактивные средства, позволяющие одновременно проводить операции с неподвижными изображениями, видеofilмами, анимированными графическими образами, текстом, речевым и звуковым сопровождением. Мультимедиа являются обобщением гипертекстовых систем.

Настольное приложение — автономное приложение для одного пользователя.

Объект — термин объектно-ориентированного программирования, которым обозначается программный объект, являющийся классом или экземпляром класса.

Объектно-ориентированное программирование (ООП) — наиболее популярная в настоящее время парадигма программирования, являющаяся развитием структурного программирования. Центральной идеей ООП является инкапсуляция, т. е. структурирование программы на модули особого вида, объединяющего данные и процедуры их обработки, причем внутренние данные модуля не могут быть обработаны иначе, как предусмотренными для этого процедурами.

Операнд — элемент данных, участвующий в операции.

Оператор — допустимое в языке программирования высокого уровня предложение, задающее целостное законченное действие или описание.

Операционная система — главная управляющая программа (комплекс программ) на ЭВМ.

Открытая архитектура компьютера IBM PC — возможность расширять конструкцию компьютера путем подключения дополни-

тельных блоков (карт расширения) через специальное устройство подключения, называемое шиной расширения.

Отладка программы — специальный этап в разработке программы, состоящий в выявлении и устранении программных ошибок, факт существования которых уже установлен.

Офисное приложение — прикладное вертикальное приложение масштаба предприятия, предназначенное для ввода, вывода, хранения и обработки документов, и не связанное критически с реальным временем.

Офисное средство разработки — приложение, которое одновременно является как офисным приложениям, так и средством разработки офисных приложений. Например, все приложения пакета Microsoft Office 2000 являются офисными средствами разработки.

Переменная — элемент данных, которому присвоено имя.

Персональные компьютеры — малогабаритные вычислительные машины, предназначенные для персонального использования.

Предметная область — конкретная область человеческой деятельности, для автоматизации которой используется офисное приложение. Например, ведение документов отдела кадров предприятия.

Приведение типов — автоматическое преобразование значения одного типа в эквивалентное значение другого типа.

Прикладная программа — программа, предназначенная для решения определенных задач и получения конкретных результатов. Например, программа расчета заработной платы.

Приложение масштаба предприятия — приложение, с которым работает несколько категорий пользователей локальной или глобальной сети.

Приложение реального времени — приложение, в котором точный учет хода астрономического (реального) времени является критически важным для выполнения основных функций приложения.

Программная инженерия — 1. Совокупность инженерных методов и средств создания ПО. 2. Дисциплина, изучающая применение строгого систематического количественного (т. е. инженерного) подхода к разработке, эксплуатации и сопровождению ПО.

Программное обеспечение (программный продукт) — это совокупность отдельных программных средств, их документации,

гарантии качества, рекламных материалов, мер по обучению пользователей, распространению и сопровождению готового программного обеспечения.

Продолжающаяся разработка — процесс разработки приложения, включающий в себя как разработку в обычном смысле, так и модификацию программы (и другие действия) в процессе эксплуатации.

Проект — набор программных модулей, образующих приложение или его компонент.

Процедура — минимальная семантически законченная программная конструкция, допускающая выполнение.

Процесс разработки приложения — определенная последовательность сменяющих друг друга видов деятельности, нацеленная на создание приложения.

Процесс (ЖЦ ПО) — совокупность взаимосвязанных действий, преобразующих некоторые входные данные в выходные.

Процесс создания ПО — совокупность упорядоченных во времени, взаимосвязанных и объединенных в стадии работ, выполнение которых необходимо и достаточно для создания ПО, соответствующего заданным требованиям.

Разработка ПО — комплекс работ по созданию ПО и его компонентов в соответствии с заданными требованиями, включая оформление проектной и эксплуатационной документации, подготовку материалов, требуемых для проверки работоспособности и соответствующего качества программных продуктов, материалов, необходимых для организации обучения персонала, и т. д.

Разработчик — субъект (отдельное лицо или организация), выполняющий разработку офисного приложения по заказу. См. *офисное приложение*.

Реакция на событие — процедура, которая выполняется при возникновении события.

Репозиторий — база данных, предназначенная для хранения проектных метаданных (версий проекта и его отдельных компонентов), синхронизации поступления информации от различных разработчиков при групповой разработке, контроля метаданных на полноту и непротиворечивость.

Свойство объекта — данные, хранящиеся в объекте. См. *класс, объектно-ориентированное программирование*.

Системная программа — программа, предназначенная для обеспечения работы прикладных программ. Например, операционная система.

Сопровождение ПО — внесение изменений в ПО в целях исправления ошибок, повышения производительности или адаптации к изменившимся условиям работы или требованиям.

Средство разработки — приложение, используемое при разработке других приложений.

Стадия ЖЦ ПО — часть процесса создания ПО, ограниченная определенными временными рамками и заканчивающаяся выпуском конкретного продукта (моделей ПО, программных компонентов, документации), определяемого заданными для данной стадии требованиями.

Структурное программирование — исторически первая оформленная парадигма программирования.

Тело процедуры — последовательность операторов в процедуре.

Тестирование — процесс исполнения программы в целях обнаружения ошибок.

Технология проектирования ПО — совокупность технологических операций проектирования в их последовательности и взаимосвязи, приводящая к разработке проекта ПО.

Уровень языка программирования — качественная характеристика языка программирования.

Фаза — часть процесса, во время которой выполняются определенные функции с целью достижения определенной вехи.

Форма — типовой документ, в котором меняются только некоторые поля, а основная часть остается неизменной.

Функция рабочего листа — функция, которую можно использовать в формулах на рабочем листе.

Шина — совокупность проводов, спроектированных таким образом, чтобы выполнялось соглашение о правилах передачи информации между чипами, подключенными к шине.

Электронная таблица — структура данных, внешне похожая на таблицу, в ячейках которой могут храниться значения и формулы. Используется для организации вычислений и другой обработки данных на персональном компьютере. Например, Excel 2000.

Элемент управления — средство графического интерфейса пользователя, предназначенное для ввода/вывода информации и

управления работой программы. Воздействие пользователя на элемент управления вызывает событие в программе.

Язык программирования VBA (Visual Basic for Applications) — визуальный объектно-ориентированный язык программирования высокого уровня.

Язык программирования высокого уровня — язык программирования, в который введены элементы, допускающие описание программы в наглядном, легко воспринимаемом человеком виде, упрощающие и автоматизирующие процесс программирования.

BIOS (Basic Input Output System) — базовая система ввода-вывода. Программа, с которой начинается запуск ПК.

CASE-средство — программное средство, поддерживающее процессы жизненного цикла ПО (определенные в стандарте ISO/IEC 12207:1995), включая анализ требований к системе, проектирование прикладного ПО и баз данных, генерацию кода, тестирование, документирование, обеспечение качества, конфигурационное управление и управление проектом, а также другие процессы.

ПРИЛОЖЕНИЕ

Программа оптимизации Поиск решения

Для вызова программы оптимизатора выберите команду меню *Сервис>Поиск решения*. Если команда *Поиск решения* отсутствует в меню *Сервис*, то надо установить эту надстройку.

1. Установка программы Поиск решения

В меню *Сервис* выберите команду *Надстройки*.

В диалоговом окне *Надстройки* установите флажок *Поиск решения*. Но если диалоговое окно *Надстройки* не содержит команды *Поиск решения*, нажмите кнопку *Обзор* и укажите диск и папку, в которой содержится файл надстройки *Solver.xla* (как правило, это папка *Library\Solver* folder), или запустите программу *Setup*, если найти файл не удастся.

Надстройка, указанная в диалоговом окне *Надстройки*, остается активной до тех пор, пока она не будет удалена.

2. Диалоговое окно Поиск решения

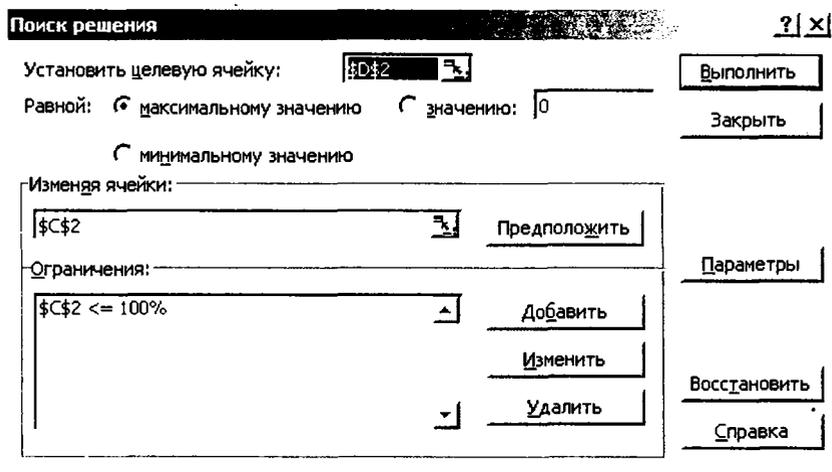


Рис. 21. Внешний вид окна Поиск решения

Окно *Поиск решения* (рис. 21) вызывается командой меню *Сервис* \square *Поиск решения*. Поле *Установить целевую ячейку* служит для указания целевой ячейки, значение которой необходимо максимизировать, минимизировать или установить равным заданному числу. Эта ячейка должна содержать формулу,

Кнопка *Равной* служит для выбора варианта с заданным значением целевой ячейки. Чтобы установить заданное число, введите его в поле. В наших примерах мы нажимаем кнопку *максимальному значению*.

Поле *Изменяя ячейки* служит для указания ячеек, значения которых изменяются в процессе поиска решения до тех пор, пока не будут выполнены наложенные ограничения и условие оптимизации значения ячейки, указанной в поле *Установить целевую* ячейку. В поле *Изменяя ячейки* вводятся имена или адреса изменяемых ячеек, разделяя их запятыми. Допускается установка до 200 изменяемых ячеек.

Поле *Предположить* используется для автоматического поиска ячеек, влияющих на формулу, ссылка на которую дана в поле *Установить целевую* ячейку. Результат поиска отображается в поле *Изменяя ячейки*.

Поля *Ограничения* служат для отображения списка граничных условий поставленной задачи. В нашем примере это величина налоговой ставки — она не должна превышать 100%.

Команда *Добавить* служит для отображения диалогового окна *Добавить ограничение*.

Команда *Изменить* служит для отображения диалоговое окна *Изменение ограничения*.

Команда *Удалить* служит для снятия указанного курсором ограничения.

Команда *Выполнить* служит для запуска поиска решения поставленной задачи.

Команда *Закрыть* служит для выхода из окна диалога без запуска поиска решения поставленной задачи. При этом сохраняются установки, сделанные в окнах диалога, появлявшихся после нажатий на кнопки *Параметры*, *Добавить*, *Изменить* или *Удалить*.

Кнопка *Параметры* служит для отображения диалогового окна *Параметры поиска решения*, в котором можно загрузить или сохранить оптимизируемую модель и указать предусмотренные варианты поиска решения.

Кнопка *Восстановить* служит для очистки полей окна диалога и восстановления значений параметров поиска решения, используемых по умолчанию.

3. Ввод и редактирование ограничений

Диалоговые окна изменения и добавления ограничений одинаковы (рис. 22). В поле *Ссылка на ячейку* вводится адрес или имя ячейки или диапазона, на значения которых накладываются ограничения.

Выберите из раскрывающегося списка условный оператор, который необходимо разместить между ссылкой и ее ограничением. Это знаки операторов: не более, не менее, равно и т. д.

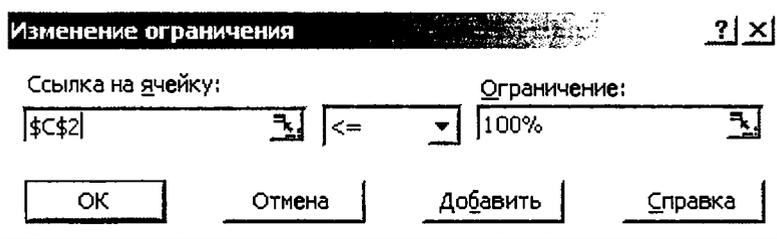


Рис. 22. Диалоговое окно Изменение ограничения

В поле *Ограничения* введите число, формулу или имя ячейки или диапазона, содержащих или вычисляющих ограничивающие значения.

Чтобы приступить к набору нового условия, нажмите кнопку *Добавить*.

3. Вычисления и результаты решения задачи

Для запуска оптимизатора нажмите кнопку *Выполнить* в окне *Поиск решения*.

Программа начинает работать, в строке сообщений (слева внизу листа) появляется сообщение *Постановка задачи...* Ваша таблица с моделью и параметрами алгоритма автоматически приводится к стандартам постановки задач математического программирования. Это преимущество Excel. В других пакетах Вам пришлось бы оторваться от экономической сути задачи и заниматься формальной математической постановкой задачи. После этапа постановки решается задача.

Чтобы прервать поиск решения, нажмите клавишу ESC. Microsoft Excel пересчитает лист с учетом найденных значений влияющих ячеек.

По окончании счета появляется диалоговое окно *Результаты поиска решения* (рис. 23).

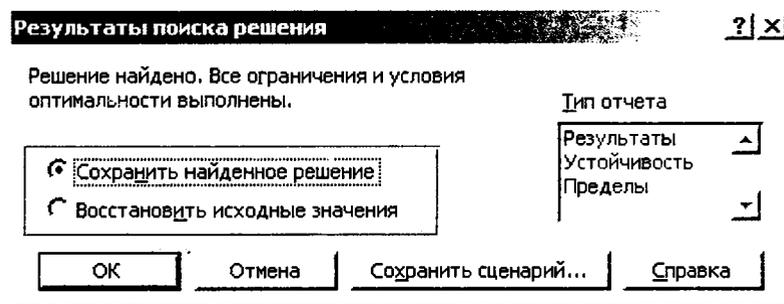


Рис. 23. Диалоговое окно Результаты поиска решения

Нажав соответствующую кнопку, можно сохранить найденное решение во влияющих ячейках модели или восстановить исходные значения.

Поле *Тип отчета* служит для указания типа отчета, размещаемого на отдельном листе книги.

Отчет *Результаты* используется для создания отчета, состоящего из целевой ячейки и списка влияющих ячеек модели, их исходных и конечных значений, а также формул ограничений и дополнительных сведений о наложенных ограничениях.

Отчет *Устойчивость* используется для создания отчета, содержащего сведения о чувствительности решения к малым изменениям в формуле модели или в формулах ограничений. Такой отчет не создается для моделей, значения в которых ограничены множеством целых чисел. В случае нелинейных моделей отчет содержит данные для градиентов и множителей Лагранжа. В отчет по нелинейным моделям включаются ограниченные затраты, фиктивные цены, а также диапазоны ограничений.

Отчет *Пределы* используется для создания отчета, состоящего из целевой ячейки и списка влияющих ячеек модели, их значений, а также нижних и верхних границ. Такой отчет не создается для моделей, значения в которых ограничены множеством целых чисел. Нижним пределом является наименьшее значение, которое

может содержать влияющая ячейка, в то время как значения остальных влияющих ячеек фиксированы и удовлетворяют наложенным ограничениям. Соответственно верхним пределом называется наибольшее значение.

К сожалению, эти отчеты очень неудобны. Они перегружены плохо читаемыми абсолютными ссылками со знаками доллара. Желает лучшего и перевод с английского на русский.

Кнопка *Сохранить сценарий* служит для отображения диалогового окна *Сохранение сценария*, в котором можно сохранить сценарий решения задачи, чтобы использовать его в дальнейшем с помощью диспетчера сценариев Microsoft Excel. В поле *Название сценария* введите имя сценария. Чтобы создать сценарий, не сохраняя найденное решение и не отображая результатов на листе, сохраните сценарий в диалоговом окне *Результаты поиска решения*, а затем выберите *Восстановить исходные значения*.

ЛИТЕРАТУРА

1. Козырев А. А. Информационные технологии в экономике и управлении: Учебник. 2-е изд. СПб.: Изд-во Михайлова В. А., 2001.
2. Брукшир Дж., Гленн. Введение в компьютерные науки. Общий обзор: Пер с англ. 6-е изд. М.: Издательский дом «Вильямс», 2001.
3. Ильина О. П. Информационные технологии бухгалтерского учета. СПб.: Питер, 2001.
4. Новиков Ю., Черепанов А. Персональные компьютеры: аппаратура, системы, Интернет: Учебный курс. СПб.: Питер, 2001.
5. Ветров С. И. Компьютерное «железо»: полное руководство пользователя. М.: СОЛОН-Р, 2002.
6. Васильев А., Андреев А. VBA в Office 2000: Учебный курс. СПб.: Питер, 2001.
7. Гарнаев А. Ю. Excel, VBA, Internet в экономике и финансах. СПб.: БХВ-Петербург, 2002.
8. Гарнаев А. Ю. и др. Microsoft Office 2000: разработка приложений / Под общ. ред. Ф. А. Новикова. СПб.: БХВ-Петербург, 2000
9. www.ixbt.ru
10. www.ferra.ru
11. www.potrebitel.ru

СОДЕРЖАНИЕ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ		
1. Учебные цели	3	
2. Связь с другими дисциплинами	3	
3. Программа дисциплины	4	
4. График изучения дисциплины и форма контроля	5	
5. Рекомендации по изучению дисциплины	5	
КОНСПЕКТ ЛЕКЦИЙ		
Введение	7	
Часть I. Аппаратное обеспечение ПК	10	
1. Классификация персональных компьютеров	10	
2. Блоки обработки информации.	13	
3. Блоки хранения информации	20	
4. Подключение внешних устройств	26	
Часть II. Программное обеспечение ПК	30	
5. Определение и классификация программных продуктов	30	
6. Стратегия разработки ПО	32	
7. Технология быстрой разработки приложений — RAD	41	
8. Элементы языка VBA	46	
9. Практика использования VBA — объекты MS Excel	53	
10. Программное обеспечение для решения финансово-экономических задач	55	
Заключение	69	
МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ		70
ВОПРОСЫ ТЕСТА ДЛЯ ИТОГОВОГО КОНТРОЛЯ		75
ГЛОССАРИЙ		93
ПРИЛОЖЕНИЕ		101
ЛИТЕРАТУРА		106

*Игорь Николаевич ГЛУХИХ
Дмитрий Витальевич СЕМИХИН*

КОМПЬЮТЕРНЫЕ ПРОДУКТЫ

Учебное пособие

Редактор *Н. П. Дементьева*
Корректор *Т. В. Камбарова*
Технический редактор *Н. Г. Яковенко*
Компьютерная верстка *Е. С. Боровинская*
Трафаретная печать *А. В. Ольшанский*
Офсетная печать *В. В. Торопов, С. Г. Наумов*

13.01.04

10

ЛР 020405 от 14.05.97
Подписано в печать 13.01.04. Тираж 3000 экз.
Объем 6,75 п. л. Формат 60х84/16. Заказ 10.

Издательство Тюменского государственного университета
625000, г. Тюмень, ул. Семакова, 10.