

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ЭКОНОМИКИ И УПРАВЛЕНИЯ
КАФЕДРА ЭКОНОМИЧЕСКОЙ ИНФОРМАТИКИ

ИНФОРМАТИКА

Учебно-методический комплекс
по экономическим специальностям

НОВОСИБИРСК
2007

Издается в соответствии с планом учебно-методической работы НГУЭУ

Составители:

доцент Степанова Т.И.
ст. преподаватель Марков Б.И.
ст. преподаватель Михайленко Н.А.
ст. преподаватель Проскурина О.И.
ст. преподаватель Филимонова Н.А.
ассистент Боридько О.Н.

И 74 Информатика: Учебно-методический комплекс. — Новосибирск: НГУЭУ, 2007. — 160 с.

Предложенный учебно-методический комплекс (УМК) представляет из себя пособие для самостоятельного изучения дисциплины «Информатика» в НГУЭУ. УМК включает рабочую программу, курс лекций и методические указания для выполнения контрольной работы.

УМК предназначен для студентов заочного отделения на выездных площадках и филиалах НГУЭУ.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ

Для заочной формы обучения изучение дисциплины осуществляется самостоятельно по предлагаемым методическим материалам и научной литературе. Данный учебно-методический комплекс (УМК) по изучению дисциплины «Информатика» состоит из следующих разделов:

- методические рекомендации по изучению дисциплины;
- рабочая программа;
- курс лекций;
- глоссарий;
- методические указания по выполнению контрольной работы;
- материалы тестовой проверки знаний.

Изучение дисциплины необходимо начать со знакомства с рабочей программой, в частности с ее тематикой и методикой проверки приобретенных студентом знаний. Это необходимо, чтобы иметь представление о структуре и содержании курса.

После этого рекомендуется перейти к освоению теоретического материала, предлагаемого в курсе лекций. Курс можно изучать последовательно или по темам. В конце каждой темы приведены контрольные вопросы для проверки степени усвоения учебного материала. Для более углубленного изучения материала по каждой теме предлагается список рекомендуемой литературы.

Для проверки усвоения дисциплины учебным планом предусмотрены контрольная работа и экзамен. Знание студентом теоретического материала и навыки работы на ПК проверяются как при защите контрольной работы, так и на экзамене.

Контрольная работа включает тестовые вопросы по информационным технологиям, техническим, программным средствам ПК и решение практической задачи средствами табличного процессора Excel. Содержание, правила оформления контрольной работы, пример решения задачи приведены в методических указаниях по выполнению контрольной работы.

Выполнение контрольной работы целесообразно начать после изучения всего теоретического материала курса лекций. Контрольная работа выполняется в течение семестра и сдается на проверку преподавателю.

Экзаменационная оценка является итоговой по дисциплине и проставляется в приложение к диплому (выписке из зачетной книжки).

**ТЕКСТЫ ЛЕКЦИЙ
УЧЕБНОЙ ДИСЦИПЛИНЫ
«ИНФОРМАТИКА»**

ПРЕДИСЛОВИЕ

Роль информации в современном обществе существенно меняется. Информация приобретает преобразующий, определяющий характер. Создание индустрии информатики и превращение информационного продукта в товар приводит к тому, что само общество трансформируется из индустриального в информационное.

Информация охватывает все стороны жизни общества — от материального производства до социальной сферы. Стремительное развитие товарных и финансовых рынков явилось мощным толчком к интенсивному нарастанию процессов информатизации во всех сферах жизни общества. Изменились подходы к оценке роли информации и информационному обслуживанию производственно-хозяйственной, управленческой деятельности.

Качественно новое обслуживание информационных процессов человеческой деятельности связано с использованием современной вычислительной техники, систем телекоммуникаций, созданием сетей ЭВМ.

Рыночные отношения предъявляют повышенные требования к своевременности, достоверности, полноте информации, без которой немыслима эффективная маркетинговая, финансово-кредитная, инвестиционная деятельность.

Вычислительные сети обеспечивают доступ к удаленным информационным ресурсам всего мира. Сегодня руководитель и исполнитель на своем рабочем месте могут практически мгновенно получить исчерпывающую информацию для анализа конкретной производственной или рыночной ситуации.

Сегодня актуальными становятся прежде всего быстрое принятие решений, степень адекватности аналитических данных реальным процессам, возможность использования экономико-математических методов и моделей для анализа конкретных финансово-производственных ситуаций. Такая постановка вопросов требует новых научно обоснованных решений, подхода и квалифицированных кадров.

Формирование управленческих кадров новой формации, овладение ими экономико-математическими методами и необходимыми знаниями и умениями для реализации своих решений в соответствующей информационно-технологической среде является важной проблемой.

Выпускнику экономического вуза приходится:

- работать как конечному пользователю на персональном компьютере в условиях «электронного офиса», интегрированной информационной системы, в локальных и глобальных компьютерных сетях;
- совершенствовать технологические и управленческие процессы на своем рабочем месте с использованием новейших технических и программных средств.

Поэтому при обучении в вузе студенты, получая знания по выбранной ими специальности, должны научиться использовать компьютерную технику в режиме пользователя для решения экономических задач, самостоятельно ставить и решать задачи компьютеризации отдельных участков своей профессиональной деятельности. Это способствует формированию у специалистов современного информационно-технологического мышления.

РАЗДЕЛ 1

ОСНОВЫ ИНФОРМАТИКИ

ТЕМА 1.1. ПРЕДМЕТ «ИНФОРМАТИКА». ИНФОРМАЦИЯ, ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

1.1.1. ИНФОРМАТИКА — ПРЕДМЕТ И ЗАДАЧИ

Информатика — это наука и сфера практической деятельности, связанная с различными аспектами получения, хранения, обработки, передачи и использования информации.

Само слово «Информатика» возникло в начале 60-х годов для обозначения области автоматизированной переработки информации в обществе. Потребность общества в совершенствовании технологии обработки информации обусловила интенсивное развитие вычислительной техники, а также способствовала формированию информатики.

Использование ЭВМ в различных сферах человеческой деятельности приводит к коренной перестройке технологии производства, повышению производительности труда. Это актуально и для сферы экономики.

Существует много определений информатики.

В нашей стране в 1983 г. на сессии годовичного собрания Академии наук СССР было организовано новое отделение информатики, вычислительной техники и автоматики. Там же было дано такое определение информатики:

***Информатика** — это комплексная научная и инженерная дисциплина, изучающая все аспекты разработки, проектирования, создания, оценки, функционирования основанных на ЭВМ систем переработки информации, их применения и воздействия на различные области социальной практики.*

Информатика нацелена на разработку общих методологических принципов построения информационных моделей.

При этом, если под моделью понимают упрощенное представление о реальном объекте, то информационная модель — совокупность информации об объекте или процессе. Для того чтобы создать информационную модель объекта, необходимо изучить его свойства и состояния. Поэтому методы информатики применяют всюду, где существует возможность описания объекта, явления, процесса с помощью информационных моделей.

Развитие прикладной информатики — технологии использования компьютерной техники для реализации конкретных приложений — прошло три этапа:

I — решение задач простого счета;

II — создание информационной поддержки принятия решений, предусматривающее использование традиционных экономико-математических и моделей для решения экономических и других видов задач;

III — применение новых информационных технологий, создание баз данных, экспертных систем, баз знаний.

1.1.2. СТРУКТУРА ИНФОРМАТИКИ

С современной точки зрения информатику можно рассматривать и как науку, и как прикладную дисциплину, и как отрасль народного хозяйства.

Информатика как *отрасль народного хозяйства* состоит из однородной совокупности предприятий разных форм хозяйствования, где занимаются производством компьютерной техники, программных продуктов и разработкой современной технологии переработки информации. Специфика и значение информатики как отрасли производства состоят в том, что от нее во многом зависит рост производительности труда в других отраслях народного хозяйства.

Как наука информатика переживает сейчас период своего бурного развития. Информатика как *фундаментальная наука* занимается разработкой методологии создания информационного

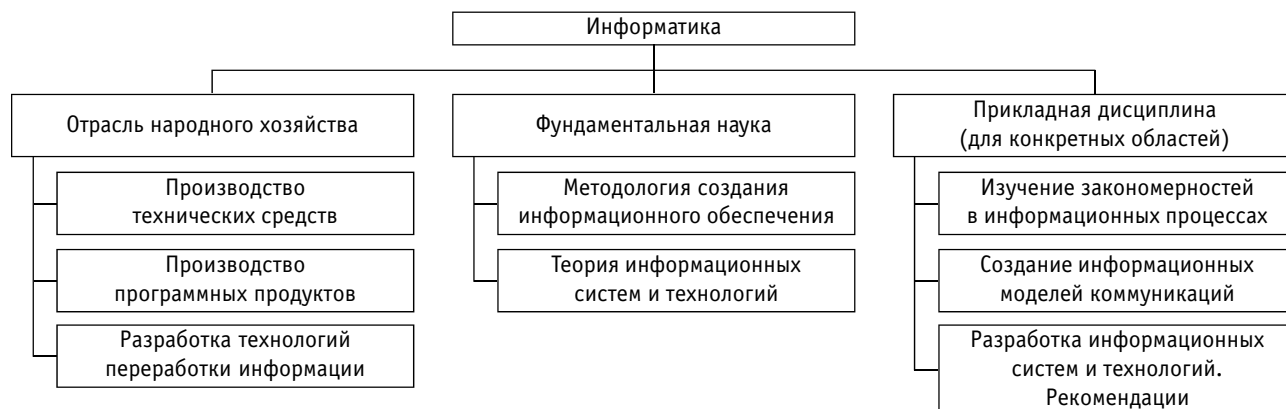


Рис.1. Структура информатики

обеспечения процессов управления любыми объектами (не только в технических системах, но также в природе и обществе) на базе компьютерных информационных систем.

Информатика может быть отнесена к классу естественных наук, поскольку ее основные понятия носят общенаучный характер, используются во многих других науках и видах деятельности: законы обработки информации в общественных, биологических и инженерных системах едины.

Экономической информатикой называется наука, изучающая методы автоматизированной обработки экономической информации с помощью средств вычислительной техники.

Информатика как *прикладная дисциплина* занимается:

- изучением закономерностей в информационных процессах (накопление, переработка, распространение информации);
- созданием информационных моделей коммуникаций в различных областях человеческой деятельности;
- разработкой информационных систем и технологий в конкретных областях и выработкой рекомендаций относительно их жизненного цикла: для этапов проектирования и разработки систем, их производства, функционирования и т.д.

То есть информатика существует не сама по себе, а является комплексной научно-технической дисциплиной, призванной создавать новые информационные технологии и технологии для решения проблем в других областях.

1.1.3. ПОНЯТИЕ ИНФОРМАЦИИ

Понятие «информация» является одним из фундаментальных для информатики. Давайте уточним, что такое информация? Обычно информацию определяют как сообщение, сведения, данные. Однако важны не сведения или данные вообще. Различные данные оказываются неравноценными при равном количестве показателей. Короткий документ зачастую бывает содержательнее пухлых докладов и справок. А это значит, что информация отлична от данных вообще; интуитивно мы, скорее, понимаем ее как то *полезное содержание, которое из данных можно извлечь*.

Большой вклад в изучение природы информации внесла теория информации. Эта теория трактует информацию как уменьшение неопределенности в отношении ожидаемых событий.

В чем отличие «данных» от «информации»?

Данные — это признаки или результаты наблюдений над объектами или явлениями, которые по каким-то причинам не используются, а только хранятся.

Как только данные начинают использовать в каких-либо практических целях, они превращаются в информацию.

Информацией являются новые сведения, воспринятые, понятые и оцененные как полезные для решения тех или иных задач. Исходя из этого можно определить информацию как «используемые данные».

Информатика рассматривает информацию как совокупность концептуально связанных между собой сведений, понятий, данных, изменяющих представление о явлении или объекте окружающего мира, т.е. уменьшающих меру неопределенности знаний об окружающем мире, о чем-либо.

Информация несет человеку новые знания об объектах, процессах, явлениях. На протяжении всей своей жизни человек постоянно участвует во всевозможных информационных процессах.

Информационный процесс — процесс, в результате которого осуществляется прием, передача (обмен), преобразование и использование информации.

При работе с информацией всегда имеется ее источник и потребитель (получатель). Пути и процессы, обеспечивающие передачу сообщений от источника информации к ее потребителю, называются *информационными коммуникациями*.

1.1.3.1. Качество информации

Информация — это важнейший стратегический ресурс. Поэтому к информации предъявляются особые требования. Возможность и эффективность использования информации обуславливаются основными показателями качества.

Адекватность — уровень соответствия создаваемого с помощью полученной информации образа реальному объекту, процессу, явлению.

Репрезентативность (обоснованность отбора существенных признаков и связей отображаемого явления) — правильность отбора и формирования информации в целях адекватного отражения свойств объектов. Нарушение репрезентативности информации приводит к существенным ее погрешностям.

Содержательность — с увеличением содержательности информации для получения одних и тех же сведений требуется преобразовать меньший объем данных.

Достаточность или *полнота* информации означает, что она содержит минимальный, но достаточный для принятия правильного решения состав (набор показателей). Неполная (недостаточная) информация, как и избыточная, снижает эффективность принимаемых пользователем решений.

Доступность восприятию пользователя. Например, в информационной системе информация преобразовывается к доступной и удобной для восприятия пользователем форме.

Актуальность информации определяется степенью сохранения ценности информации для управления в момент ее использования. (А это зависит и от динамики изменения ее характеристик, и от интервала времени, прошедшего с момента возникновения данной информации.)

Своевременность — означает ее поступление в установленные сроки, не позже заранее назначенного момента времени, согласованного с временем решения поставленной задачи.

Точность — определяется степенью близости получаемой информации к реальному состоянию объекта, процесса, явления.

Достоверность — определяется ее свойством отражать реально существующие объекты с необходимой точностью, т.е. достоверность определяет допустимый уровень искажения информации, при котором сохраняется эффективность функционирования системы.

Устойчивость — отражает ее способность реагировать на изменение исходных данных без нарушения необходимой точности.

1.1.3.2. Экономическая информация, ее особенности и классификация

Классифицировать информацию можно по разным признакам.

По виду обслуживаемой ею человеческой деятельности информация подразделяется на:

- научную;
- производственную;
- управленческую;
- медицинскую;
- правовую и т.д.

Каждый из видов информации имеет свои особенности технологии обработки, формы представления, требования к точности и достоверности.

Под *управленческой* понимается информация, которая обслуживает процессы производства, распределения, обмена и потребления материальных благ и обеспечивает решение задач организационно-экономического управления народным хозяйством и его звеньями. Она представляет собой разнообразные сведения экономического, технологического, социального, юридического, демографического и другого содержания.

Важнейшей составляющей управленческой информации является информация *экономическая*, которая отражает социально-экономические процессы в производственной и непроизводственной сферах, во всех отраслях народного хозяйства, во всех органах и на всех уровнях управления.

Особенности экономической информации:

1) экономическая информация — это система показателей, представляющая собой *количественные величины, цифровые значения*, что предопределило возможность широкого использования вычислительной техники;

2) для экономической информации характерна *цикличность*, т.е. для большинства производственных и хозяйственных процессов характерна повторяемость составляющих их стадий и информации, отражающей эти процессы, т.е. цикличность позволяет многократно использовать созданные программы обработки экономической информации;

3) важное значение для обработки информации имеет *форма представления* информации. Экономическая информация непременно отражается в материальных носителях: в первичных и сводных документах, на магнитных носителях;

4) отличительной чертой экономической информации является ее *объемность*. При этом совершенствование управления и возрастание объема производства сопровождаются и увеличением сопутствующих этому информационных потоков;

5) экономическая информация имеет *дискретный характер*, т.е. может быть структурирована и представлена как совокупность отдельных структурных единиц информации. Важнейшими видами структурных единиц информации являются: реквизит, показатель, документ;

6) совокупность сведений, описывающих или отражающих какой-либо объект, процесс, явление, называют *информационной совокупностью*. Информационная совокупность, неделимая далее на более мелкие единицы, называется *реквизитом* (синонимы: слово, элемент данных, атрибут). В любом документе каждый реквизит имеет значение и наименование. Различают два вида реквизитов:

- реквизиты-признаки, характеризующие качественные свойства отражаемых объектов;
- реквизиты-основания, представляющие собой количественные величины, характеризующие данный объект;

7) каждый объект, явление, процесс описываются *экономическими показателями*. Сочетание одного реквизита-основания с одним или несколькими соответствующими ему реквизитами-признаками образуют показатель. Например, информационная совокупность «5000 т угля» состоит из одного реквизита-основания (5000) и двух реквизитов-признаков («т» и «угля») и вполне отражает экономический смысл сообщения и потому является показателем.

1.1.3.3. Классификация экономической информации

1. По месту возникновения:

- *входная (входящая)* — информация, поступающая в ИС (в фирму, в структурное подразделение);
- *результатная* — как результат обработки входящих данных;
- *выходная (исходящая)* — информация, передаваемая за пределы данной ИС, т.е. поступающая из фирмы в другую фирму или организацию.

Кроме того, по месту возникновения информацию можно разделить на внешнюю и внутреннюю:

- *внешняя* информация возникает за пределами объекта;
- *внутренняя* информация — внутри объекта.

2. По отношению к процессам обработки и хранения:

- *первичная (исходная) информация* — это информация, которая возникает непосредственно в процессе деятельности объекта и регистрируется на начальной стадии;
- *промежуточная информация* используется в качестве исходных данных для последующих расчетов;
- *результатная информация* получается в процессе обработки первичной и промежуточной информации и используется для выработки управленческих решений.

3. По отношению к функциям управления.

- *плановая (директивная)* — включает директивные значения планируемых и контролируемых показателей бизнес-планирования на некоторый период в будущем — год, месяц, сутки. Например, планируемый спрос на продукцию или прибыль от ее реализации;
- *нормативно-справочная* — самый объемный и разнообразный вид информации. В общем объеме циркулирующей на фирме информации составляет 50–60%. Например, технологические нормативы, справочники по поставщикам и т.п. Обновление такой информации происходит достаточно редко;
- *учетная* — отражает фактические значения запланированных показателей за определенный период времени. На основании этой информации может быть проведен анализ деятельности организации, скорректирована плановая информация. Например, количество изготовленных изделий рабочим за одну смену и т.п.;
- *оперативная (текущая)* — информация, характеризующая производственные процессы в текущий период времени. Успех работы предприятия во многом зависит от того, насколько быстро и качественно производится обработка этой информации.

4. По степени стабильности.

- *постоянная (условно-постоянная)* — остается без изменений или подвергается незначительным корректировкам в течение более или менее длительного периода времени, это нормативно-справочная информация;
- *переменная*, которая характеризует производственные процессы в текущий (данный) момент времени, отражает результаты выполнения производственно-хозяйственных операций и, как правило, участвует в одном технологическом цикле машинной обработки, это оперативная информация.

5. Вся информационная база состоит из двух частей:

- *внемашинная* — используемая в виде, воспринимаемом человеком (например, документы, ведомости, счета, накладные, акты и т.п.);
- *внутримашинная* — информация, хранящаяся на машинных носителях в виде файлов.

1.1.3.4. Единицы информации

ЭВМ может обрабатывать информацию, представленную только в числовой форме. Любая другая информация (текстовая, графическая) преобразуется в числовую. Так, например, при вводе текста, каждый символ кодируется определенным числом (существуют специальные таблицы кодировки, наиболее известные и распространенные коды ASCII), а при выводе, наоборот, каждому числу соответствует изображение определенного символа.

Поскольку ЭВМ работают в двоичной системе счисления, то все числа представляются с помощью двух цифр — 0 и 1. Поэтому, несмотря на особенности каждого вида информации, общим для них является использование при кодировании двоичной системы счисления. Недостаток двоичного кодирования — длинные коды. Но в технике легче иметь дело с большим числом простых однотипных элементов, чем с небольшим числом сложных.

Такой двоичный разряд, принимающий значение 0 или 1, называется *битом*. Бит — это наименьшая единица информации в ЭВМ.

Восемь двоичных разрядов позволяют закодировать $2^8 = 256$ символов, этого достаточно, чтобы закодировать любую букву, цифру или служебный символ. Нажатие клавиши на клавиатуре приводит к тому, что сигнал посылается в компьютер в виде двоичного числа, которое хранится в кодовой таблице. Кодовая таблица символов — это внутреннее представление символов в компьютере. Во всем мире в качестве стандарта принята таблица ASCII (American Standard Code for Information Interchange) — Американский стандартный код для обмена информацией.

Первые 128 символов (от 0 до 127) — это цифры, прописные и строчные буквы латинского алфавита, управляющие символы. Вторая половина кодовой таблицы (от 128 до 255) предназначена для национальных символов (в том числе кириллицы), математических символов и так называемых псевдографических символов, которые используются для рисования рамок.

Например, для символа 0 двоичный код — 00110000;

А (лат.) — 01000001;

А (рус.) — 10000000.

В разных странах, на разных моделях компьютеров могут использоваться и разные варианты второй половины кодовой таблицы.

Нужно иметь в виду три особенности алфавита в кодовой таблице и их следствия:

- 1) прописные и строчные буквы представлены разными кодами, т.е. «А» и «а» — разные объекты;
- 2) при упорядочивании слов по алфавиту сравниваются между собой десятичные коды букв. Например, код латинских букв «меньше» чем русских;
- 3) латинские и русские буквы имеют разные коды, хотя некоторые визуально неразличимы.

Итак, компьютер способен распознавать только значения бита. Однако он редко работает с конкретными битами в отдельности, а совокупность из 8 битов, воспринимаемая компьютером как единое целое, называется *байтом*.

Вся работа компьютера — это управление потоками байтов, которые вводятся в компьютер с клавиатуры, считываются с дисков или передаются по линии связи, преобразовываются по командам программ, записываются на постоянное хранение на магнитный диск или выводятся на экран дисплея или бумагу в виде символов: букв, цифр, значков.

Для хранения двоичного кода одного символа выделен 1 байт памяти, равный восьми битам. Следующими более крупными единицами информации являются:

1 Кб (килобайт) = 1024 б (байт);

1 Мб (мегабайт) = 1024 Кб;

1 Гб (гигабайт) = 1024 Мб;

1 Тб (терабайт) = 1024 Гб.

Одна средняя страница текста занимает около 2 Кб памяти.

1.1.4. ПРОЦЕССЫ СБОРА, ПЕРЕДАЧИ, ОБРАБОТКИ И НАКОПЛЕНИЯ ИНФОРМАЦИИ

Понятие «информация» предполагает наличие двух объектов — источника информации и потребителя информации. Для целенаправленного использования информации ее необходимо собирать, преобразовывать, передавать, накапливать. Под информационным процессом понимается процесс восприятия, сбора, передачи, обработки и накопления информации. Информационный процесс может состояться только при наличии информационной системы. Информационные системы и обеспечивают все эти процессы при решении задач из любой области.

Восприятие информации необходимо для любой информационной системы, так как благодаря восприятию информации обеспечивается связь системы с внешней средой.

Сбор информации — это процесс получения информации из внешнего мира и приведение ее к стандарту для данной информационной системы. Сбор информации, как правило, сопровождается ее регистрацией, т.е. фиксацией информации на материальном носителе (документе или машинном носителе).

Передача информации может осуществляться различными способами: с помощью курьера, по почте, с помощью транспортных средств или по каналам связи. Специальные технические средства позволяют сокращать время передачи данных.

Обработка информации — исполнение взаимосвязанных операций в определенной последовательности, в результате которых исходная информация преобразуется в результатную. Операции преобразования информации могут быть разнообразны как по назначению, так и по сложности, технике реализации.

Накопление и хранение информации связано с необходимостью многократного ее применения, использования постоянной информации. Хранение информации осуществляется в виде информационных массивов на материальном носителе.

1.1.5. ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

Одним из основных направлений, по которому осуществляется информатизация общества, является повсеместное использование информационных систем и технологий.

В работе информационной системы на равных участвуют как технические и программные средства, так и человек.

Информационная система — это взаимосвязанная совокупность средств, методов и персонала, участвующих в обработке данных.

Информационная система содержит следующие элементы:

- **информационные ресурсы** — вся та информация, которая циркулирует в системе (массивы накопленной информации, всевозможные архивы на разных носителях и т.д., а также методики, инструкции и программы, регламентирующие процессы прохождения информации в системе, ее обработки, хранения, представления);
- **материальные ресурсы**, в том числе носители информации, технические средства сбора, передачи, обработки информации;
- **каналы циркулирования информации**;
- **определенный контингент работников**.

Информационные системы существовали с момента появления общества, поскольку на любой стадии развития общество требует для своего управления систематизированную, предварительно подготовленную информацию. Особенно это касается производственных процессов.

Информационная система должна обеспечивать прием поступающей из источника информации, ее преобразование (обработку), хранение и передачу результатов преобразования потенциальному потребителю (потребитель понимается здесь в обобщенном смысле). Им может быть любой объект живой и неживой природы: человек, устройство, другая информационная система. После приема информации потребитель должен производить какую-то операцию, т.е. реагировать. То есть в информационной системе происходят следующие процессы:

- ввод информации из внешних или внутренних источников;
- преобразование (обработка) входной информации и представление ее в удобном виде;
- хранение как входной информации, так и результатов обработки;
- вывод информации для отправки потребителю или в другую систему;
- ввод информации от потребителя через обратную связь.

Внедрение ИС способствует:

- получению более рациональных вариантов решения управленческих задач за счет внедрения математических методов, интеллектуальных систем и т.д.;
- освобождению работников от рутинной работы за счет ее автоматизации;
- обеспечению достоверности информации;
- замене бумажных носителей данных на магнитные (или оптические) диски или магнитные ленты, что приводит к более рациональной организации переработки информации на компьютере и снижению объемов документов на бумаге;
- совершенствованию структуры потоков информации и системы документооборота в фирме;
- уменьшению затрат на производство продуктов и услуг;
- отысканию новых рыночных ниш.

Информационная система экономического объекта — это совокупность средств и методов, обеспечивающих реализацию всего комплекса операций по обеспечению процесса управления необходимой информацией.

Информационная система имеется в любом экономическом объекте, являясь для него естественной составной частью. Взаимосвязь информационных потоков, средств обработки, передачи и хранения данных, а также сотрудников управленческого аппарата, выполняющих операции по переработке данных, и составляет ИС экономического объекта.

Основными компонентами ИС являются:

1. **Функциональные компоненты.** Под функциональными компонентами понимается система функций управления — полный набор взаимоувязанных во времени и пространстве работ по управлению, необходимых для достижения поставленных перед предприятием целей. Декомпозиция ИС по функциональному признаку включает в себя выделение ее отдельных частей, называемых функциональными подсистемами. Функциональный признак определяет назначение подсистемы, т.е. то, для какой области деятельности она предназначена и какие основные цели, задачи и функции она выполняет. Функциональные подсистемы в значительной степени зависят от предметной области ИС. Ряд функциональных подсистем имеют одно и то же наименование (например, бухгалтерский учет, отчетность), но внутреннее содержание для различных объектов значительно отличается друг от друга.

2. Компоненты системы обработки данных. Основная функция систем обработки данных — реализация типовых операций обработки данных:

- сбор, регистрация и перенос информации на машинные носители;
- передача информации в места ее хранения и обработки;
- обработка информации и т.д.

Принято выделять:

информационное обеспечение — совокупность методов и средств по размещению и организации информации: системы классификации и кодирования информации; унифицированные системы документации, главная цель которых — обеспечение сопоставимости показателей различных сфер общественного производства (чтобы уменьшить дублирующие показатели, чтобы не было неиспользуемых показателей и т.д.); схемы информационных потоков; методологии построения баз данных;

техническое обеспечение (аппаратное обеспечение) — комплекс технических средств, предназначенных для работы ИС (компьютеры, устройства сбора, обработки, передачи и вывода информации, устройства передачи данных и т.д.);

математическое обеспечение — средства моделирования процессов управления, типовые задачи управления, методы математического программирования, математической статистики, теории массового обслуживания и др., т.е. совокупность математических методов и моделей для реализации целей и задач ИС;

программное обеспечение — комплексы программ для решения типовых задач обработки информации, а также программы, разработанные для конкретной ИС. К программному инструментарию относятся программные продукты, использование которых позволяет достичь поставленную пользователем цель. Это, например, программные продукты общего назначения: текстовые редакторы, электронные таблицы, системы управления базами данных, электронные записные книжки и т.п.;

организационное обеспечение — анализ существующей системы управления организацией, где будет использоваться ИС, и выявление задач, подлежащих автоматизации; подготовка задач к решению на компьютере, включая техническое задание на проектирование ИС и технико-экономическое обоснование ее эффективности;

правовое обеспечение — совокупность правовых норм, определяющих создание и функционирование ИС: статус ИС; права, обязанности и ответственность персонала; порядок создания и использования информации.

3. Организационные компоненты — совокупность методов и средств, позволяющих усовершенствовать организационную структуру объектов и управленческие функции, выполняемые структурными подразделениями; определить штатное расписание и численный состав каждого структурного подразделения; разработать должностные инструкции персоналу управления в условиях функционирования системы обработки данных.

Одним из базовых элементов компьютерной информационной системы является информационная технология. Информационные технологии определяют способы, методы и средства сбора, регистрации, передачи, хранения, обработки и выдачи информации в информационной системе.

Информационная технология — процесс, использующий совокупность средств и методов обработки и передачи первичной информации для получения информации нового качества о состоянии объекта, процесса или явления.

Цель информационной технологии — производство информации для ее последующего анализа и принятия на его основе решения по выполнению какого-либо действия.

Внедрение ПК в информационную сферу и использование телекоммуникаций определило новый этап развития информационной технологии, которая с этого момента получает наименование «новая», «компьютерная».

Выделяют три основных принципа компьютерной информационной технологии:

- интерактивный (диалоговый) режим работы с компьютером;
- интеграция с другими программными продуктами;
- гибкое изменение данных и поставленных задач.

В роли технических средств производства информации выступают аппаратное, программное и математическое обеспечение этого процесса. С их участием перерабатывается первичная информация в информацию нового качества.

К программному инструментарию относятся программные продукты, использование которых позволяет достичь поставленную пользователем цель. Это, например, текстовые редакторы, электронные таблицы, системы управления базами данных, 1С:Бухгалтерия и т.д.

Каково соотношение между информационной системой и информационной технологией?

Информационная технология представляет собой процесс, состоящий из четко регламентированных правил выполнения различных операций с данными, хранящимися в компьютере. Основная цель — получить необходимую для пользователя информацию в результате целенаправленных действий по переработке первичной информации.

Информационная система — это среда, равноправными элементами которой являются: работники персонала, компьютеры, компьютерные сети, программные продукты, базы данных и т.д. Основная цель — организация хранения и передачи информации.

Реализация функций информационной системы невозможна без знания ориентированной на нее информационной технологии. Информационная технология может существовать и вне сферы ИС.

Контрольные вопросы

1. Как и для чего появилась информатика? Три этапа развития информатики.
2. Что такое информатика?
3. Охарактеризуйте информатику как отрасль, как науку, как прикладную дисциплину.
4. В чем отличие «данных» от «информации»?
5. Назовите характеристики для оценки качества информации.
6. Экономическая информация, ее особенности.
7. Классификация экономической информации.
8. Каковы особенности экономической информации?
9. Чем вызвано использование двоичной системы счисления?
10. Что такое бит, байт? Какие единицы измерения информации вы знаете?
11. Назначение кодов ASCII.
12. Что такое система, какие вы знаете системы?
13. Что понимается под информационным процессом?
14. Что такое информационная система?
15. Назовите элементы информационной системы.
16. Как вы понимаете — информационная система экономического объекта?
17. Перечислите и охарактеризуйте основные компоненты ИС.
18. Что такое информационная технология?
19. Как следует понимать «новая информационная технология»?
20. Как соотносятся информационные системы и информационные технологии?

Литература

1. Автоматизированные информационные технологии в экономике: Учебник / М.И.Семенов, И.Т.Трубилин, В.И.Лойко, Т.П.Барановская; Под общ. ред. И.Т.Трубилина. — М.: Финансы и статистика, 2001. — С.13–45, 60–64.
2. Автоматизированные информационные технологии в экономике: Учебник / Под ред. проф. Г.А.Титоренко. — М.: Компьютер; ЮНИТИ, 1998. — С.6–40.
3. *Евдокимов В.В. и др.* Экономическая информатика: Учебник для вузов. — СПб.: Питер, 1997. — С.13–31.
4. Информатика. Базовый курс / С.В.Симонович и др. — СПб.: Питер, 2001. — С.11–36.
5. Информатика: Учебник / Под ред. проф. Н.В.Макаровой. — М.: Финансы и статистика, 1998. — С.13–59.
6. Информационные системы в экономике: Учебник / Под ред. проф. В.В.Дика. — М.: Финансы и статистика, 1996. — С.8–42.
7. Информационные технологии управления: Учебное пособие / Под ред. Ю.М.Черкасова. — М.: ИНФРА-М, 2001. — С.5–33.
8. Экономическая информатика / Под ред. П.В.Конюховского и Д.Н.Колесова. — СПб.: Питер, 2000. — С.14–30.

ТЕМА 1.2. ТЕХНИЧЕСКИЕ И ПРОГРАММНЫЕ СРЕДСТВА РЕАЛИЗАЦИИ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ

1.2.1. НАЗНАЧЕНИЕ ТЕХНИЧЕСКИХ СРЕДСТВ ИНФОРМАЦИОННЫХ СИСТЕМ

Обязательным условием функционирования информационной системы является техническое обеспечение. *Техническое обеспечение* — комплекс технических средств, предназначенных для работы информационной системы, и соответствующая документация на эти средства.

В работе информационной системы можно выделить несколько этапов:

- формирование первичных сообщений;
- размещение и систематизация данных для обеспечения быстрого поиска информации;
- обработка данных;
- представление данных в виде, удобном для восприятия пользователем.

Для выполнения каждого из этих этапов необходимы соответствующие технические средства.

Комплекс технических средств информационной системы составляют:

- устройства регистрации информации;
- устройства сбора, накопления, обработки, передачи и вывода информации;
- электронно-вычислительные машины (ЭВМ);
- средства телекоммуникации и связи;
- оргтехника и др.

Для ввода, обработки, хранения и вывода информации в современных информационных системах предназначены вычислительные машины (ЭВМ) или компьютеры.

1.2.2. ТЕНДЕНЦИИ РАЗВИТИЯ ЭВМ

В истории ЭВМ были моменты, когда новые технические возможности позволяли не только разрабатывать новые совершенные программы, но и менять организацию работы на ЭВМ.

Так, в 60-е годы пользователи пришли к необходимости изменения организации использования ЭВМ: до этого ресурсы ЭВМ предоставлялись в распоряжение одного пользователя, а это не позволяло рационально использовать потенциал машины. Тогда возникла так называемая *пакетная обработка* заданий. Пользователь подготавливал свое задание и передавал его оператору, при этом он был отделен от машины. Из заданий пользователей формировалась *очередь заданий*. Таким образом, машина не простаивала в ожидании следующего задания или реакции пользователя на свои сообщения.

Следующей идеей была организация *многозадачного использования процессора*. Суть заключалась в том, что когда в какой-то программе очередь доходила до обмена с внешним устройством, эта операция перепоручалась недорогому специализированному устройству, а центральный процессор продолжал выполнять другую программу, т.е. процессор как бы одновременно выполнял несколько программ.

Одним из направлений этой идеи явились так называемые *многопользовательские (многопультовые) системы*, работающие в режиме *разделения времени*. Эти системы представляли собой центральную ЭВМ и группу видеотерминалов. Пользователь почти не замечал, что центральная ЭВМ одновременно работает с несколькими программами (уделяя, например, каждому терминалу по несколько миллисекунд в течение секунды).

Следующей идеей является появление *многопроцессорных ЭВМ*, в которых несколько процессоров работают одновременно, и производительность машины равна сумме производительностей процессоров. Мультипроцессорный принцип обработки информации — расчленение решаемой задачи на несколько параллельных подзадач или частей. Каждая часть решается на своем процессоре. За счет такого разделения существенно увеличивается производительность.

При увеличении объемов информации и появлении баз данных возникла необходимость доступа к информационным ресурсам многих пользователей, работающих на своих ЭВМ. Так возникла идея создания сначала *локальных*, а затем и *глобальных вычислительных сетей*.

Наблюдаемые ныне тенденции выражаются в следующем:

- продолжается рост вычислительной мощности микропроцессоров (увеличивается тактовая частота);
- в одном элементе совмещается больше устройств, т.е. на одной печатной плате реализуется больше функций и, следовательно, сокращается число отдельных устройств;
- расширяется набор функций, реализуемых одним ПК («мультимедийный» компьютер помимо обработки алфавитно-цифровой информации способен работать со звуком, воспроизводить видеосигнал).

1.2.2.1. Классификация ЭВМ

Электронная вычислительная машина (ЭВМ) — комплекс технических средств, предназначенных для автоматической обработки информации в процессе решения вычислительных и информационных задач.

ЭВМ различают, например:

- по этапам создания и используемой элементной базе (ЭВМ условно делятся на поколения);
- по назначению;
- по мощности;
- по размерам и т.д.

Рассмотрим классы ЭВМ по функциональным возможностям и по габаритным характеристикам. С развитием всех этих классов часто границы между ними размываются.

Например, современные микроЭВМ не уступают по некоторым своим характеристикам миниЭВМ выпуска прошлых лет, а стоимость портативного персонального компьютера значительно превышает стоимость настольного компьютера, имеющего такие же основные параметры.

Функциональные возможности ЭВМ обуславливают важнейшие технико-эксплуатационные характеристики:

- быстродействие, измеряемое усредненным количеством операций, выполняемых машиной за единицу времени;
- разрядность и формы представления чисел, с которыми оперирует ЭВМ;
- номенклатура, емкость и быстродействие всех запоминающих устройств;
- номенклатура и технико-экономические характеристики внешних устройств хранения, обмена и ввода-вывода информации;
- система и структура машинных команд и т.д.

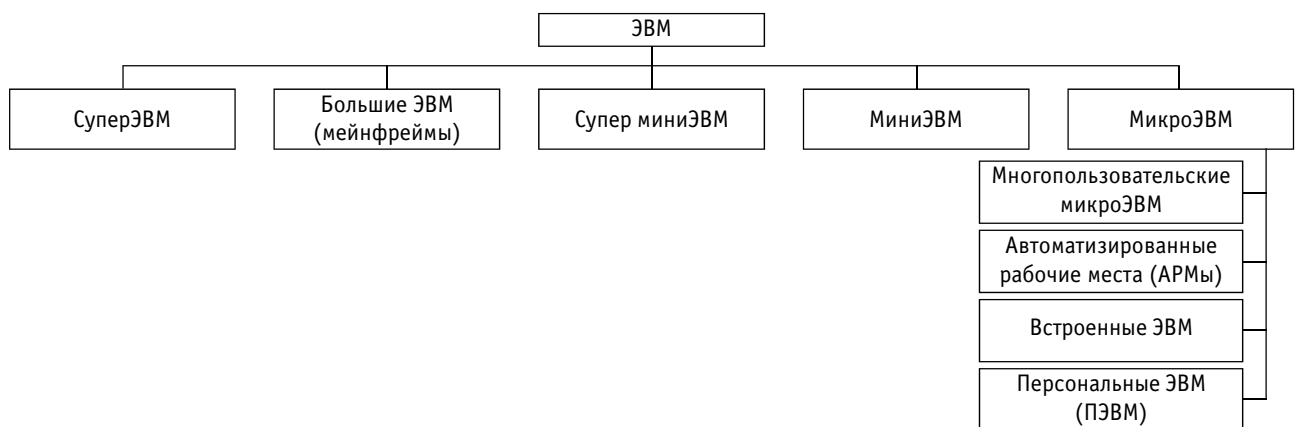


Рис.2. Классификация ЭВМ по размерам и вычислительной мощности

1. *СуперЭВМ* — вычислительная система, относящаяся к классу самых мощных систем. Такие ЭВМ требуют специальных помещений, так как имеют большие габариты, сложны в обслуживании. Число параллельно работающих процессоров — более 100.

Назначение — сложные научные расчеты, решение исследовательских и инженерных задач в областях «высоких технологий», метеорологическое прогнозирование, управление крупными банками.

2. *Большие ЭВМ (мейнфреймы)* — универсальные компьютеры общего назначения. Исторически эти ЭВМ появились первыми. Большие ЭВМ используют, как правило, в режиме разделения времени, обслуживают одновременно многих пользователей (до 1000 рабочих мест). На компьютерах этого класса сейчас находится около 70% «компьютерной» информации.

Назначение — поддерживают работу по управлению крупными фирмами, предприятиями, средними и малыми банками. Используются для обработки больших массивов информации, больших баз данных, а также в качестве серверов вычислительных сетей.

3. *Супер миниЭВМ* — вычислительные машины, относящиеся по архитектуре, размерам и стоимости к классу миниЭВМ, а по производительности сопоставимы с большой ЭВМ.

Назначение — системы управления предприятиями, многопользовательские вычислительные системы.

4. *МиниЭВМ* — используются, когда есть избыточность ресурсов больших ЭВМ. Эти компьютеры не требуют специальных помещений, работают в режиме разделения времени.

Назначение — используются в системах управления предприятиями среднего уровня, многопользовательских вычислительных системах.

5. *МикроЭВМ* — появление этих ЭВМ обусловлено появлением микропроцессоров.

Назначение — индивидуальное обслуживание пользователей, работа в локальных автоматизированных системах управления.

а) *многопользовательские микроЭВМ* — микроЭВМ, оборудованные несколькими видео-терминалами и работающие в режиме разделения времени;

б) *АРМ или рабочая станция* — ЭВМ со специальным программным обеспечением, оборудованная всеми средствами, необходимыми для выполнения работ определенного типа. Например, технические или инженерные АРМ, АРМ для автоматизированного проектирования, АРМ для издательской деятельности, так называемые настольные издательские системы, и др.;

в) *встроенные ЭВМ* представляют собой вычислители (используемые, например, станком или боевым средством), бортовой компьютер для обработки измерений. Конструктивно они выполняются в виде одной или нескольких плат и не обеспечивают реализацию широкого спектра вычислительных функций;

д) *персональные ЭВМ* — универсальные однопользовательские микроЭВМ. Последние 15–20 лет характеризуются широким распространением персональных ЭВМ во всех сферах человеческой деятельности. Мощные ПЭВМ способны обеспечить работу нескольких пользователей одновременно. Технические характеристики ПЭВМ приближаются к техническим характеристикам АРМ, поэтому на базе ПЭВМ можно построить АРМ, снабдив ее специальным оборудованием и соответствующим программным обеспечением.

Персональные компьютеры делят на настольные, портативные и карманные.

Настольные ПК являются стационарными компьютерами на рабочем месте. Одним из их достоинств является простота изменения конфигурации.

Портативные ПК очень удобны при отсутствии постоянного места работы или в поездках. Обычно они оснащены удобными средствами связи и по производительности не уступают настольным ПК.

Карманные ПК имеют размер записных книжек и используются для хранения и обработки оперативных данных.

1.2.3. АРХИТЕКТУРА ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА

Персональные ЭВМ или персональные компьютеры являются основной технической базой информационных технологий. Возможности персональных компьютеров (ПК) определяются характеристиками его функциональных блоков.

Основные функциональные элементы ПК размещены в устройствах различных компьютеров по-разному, но обязательно входят в состав любого из них.

Характерной чертой всех ПЭВМ является *модульность* структуры.

Все электронное оборудование расчленено на *модули* (так называемые *электронные платы*), связанные между собой *системной шиной*.

Каждый модуль соединен с системной шиной так, что любой модуль может передавать информацию в системную шину и любой модуль может ее принимать.

Модуль, в котором помещен микропроцессор (МП), называется *системной платой (модулем)* или *материнской платой*.

Схемы, управляющие внешними устройствами компьютера (контроллеры и адаптеры), находятся на отдельных платах, вставляемых в материнскую плату.

На системной плате расположено несколько гнездовых разъемов (так называемые *слоты*) для подключения дополнительных плат. Добавляя платы подходящих типов, можно получать требуемую конфигурацию компьютера. Так, например, с помощью дополнительных плат можно увеличить емкость оперативной памяти.

С 1980 г. на рынке появились ПК IBM PC, самой важной особенностью которых стала так называемая *открытая архитектура*. Эта архитектура, во-первых, использует принцип взаимозаменяемости, т.е. использования для сборки ПК узлов от разных производителей (но соответствующих определенным соглашениям), а во-вторых, предоставляет возможность доукомплектования ПК, наращивания его мощности уже в ходе эксплуатации ПК.

1.2.3.1. Системный блок

Центральная часть ПК, содержащая в себе практически все основные устройства, — системный блок.

В системном блоке размещены основные узлы компьютера: микропроцессор, оперативное запоминающее устройство (ОЗУ), постоянное запоминающее устройство (ПЗУ), дисководы для жестких магнитных дисков («винчестеров»), для гибких магнитных дисков и для компакт-дисков, системная шина, блок питания и др.

1.2.3.2. Материнская плата

Материнская (системная) плата — самая важная плата в компьютере. Именно к ней подключаются все другие устройства, входящие в состав системного блока ПК.

Функции материнской платы — связь и координация действий всех устройств компьютера, передача сигнала от одного устройства к другому. Видимая часть материнской платы — набор разъемов, предназначенных для установки тех или иных комплектующих.

На материнской плате, как правило, размещаются:

- микропроцессор;
- генератор тактовых импульсов;
- блоки (микросхемы) оперативной памяти (ОЗУ) и постоянной памяти (ПЗУ);
- адаптеры клавиатуры, жестких дисков (НЖМД) и гибких дисков (НГМД);
- таймер и др.

Существует несколько типов материнских плат, предназначенных для установки разных классов процессоров.

На материнской плате находятся слоты (разъемы) разных типов для подключения звуковой карты, встроенного модема, видеокарты, оперативной памяти. А также разъемы («порты») на задней стенке компьютера, предназначенные для подключения таких внешних устройств, как принтер, дисковод ZIP, «мышь» и т.д.

1.2.3.3. Микропроцессор

Микропроцессор — это основа ПК, его центральный блок, предназначенный для управления работой всех блоков компьютера и для выполнения арифметических и логических операций над информацией.

Первый микропроцессор Intel 4004 был создан в 1971 г. командой во главе с доктором Тедом Хоффом по заказу японской фирмы и предназначался для микрокалькуляторов. Но фирма обанкротилась, эта разработка перешла в собственность фирмы Intel, которая нашла другое применение этому МП, и началась эпоха персональных компьютеров.

Микропроцессор — программно-управляемое (т.е. функционирует путем выполнения некоторой программы) электронное цифровое устройство, предназначенное для обработки

информации, представленной в цифровом виде и построенное на одной или нескольких БИС (большая интегральная схема), в которых сосредоточена сложнейшая логическая схема.

МП выполняет следующие функции:

- управление и координация работы всех других компонентов микрокомпьютера;
- выборка команд и обрабатываемых данных из основной памяти;
- выполнение с помощью АЛУ арифметических, логических и других операций, закодированных в командах; передача данных между МП и основной памятью, между МП и устройствами ввода-вывода;
- обработка сигналов от устройств ввода-вывода, в том числе обработка сигналов прерывания с этих устройств.

Микропроцессор представляет собой небольшую кремниевую пластинку с несколькими десятками выводов, в ней сосредоточена сложнейшая логическая схема. В состав МП входят:

АЛУ — *арифметико-логическое устройство*, в котором выполняются арифметические и логические операции над данными, хранящимися в регистрах арифметического устройства. Основу АЛУ составляет операционный блок, который может настраиваться на различные операции и непосредственно осуществлять их. Настройка операционного блока на конкретную операцию и последовательность шагов ее выполнения обеспечиваются с помощью управляющих сигналов от УУ;

УУ — *устройство управления*, которое подает во все блоки в нужный момент времени определенные сигналы управления, обусловленные спецификой выполняемых операций, определяет последовательность операций над данными (определяет, какую операцию выполнить над какими данными, куда поместить результат, что делать на следующем шаге).

МПП — *микропроцессорная память*, которая служит для хранения информации, используемой в вычислениях в ближайшие такты работы компьютера. МПП строится на регистрах. Регистры — это электронное цифровое устройство для временного запоминания информации в форме двоичного числа или кода. Операции над числами в регистре реализуются с помощью управляющих сигналов от УУ. Многие регистры специализированы по своей функции, например, регистр-аккумулятор предназначен для хранения одного из операндов (данного) или результата операции, регистр команд используется для хранения кода текущей выполняемой команды и т.д.

Математический сопроцессор. Арифметика с плавающей запятой, как правило, более медленная. В некоторых моделях ПК для ускорения выполнения этих операций к АЛУ подключается дополнительный математический сопроцессор — специальный блок для операций с плавающей запятой — в результате чего замедление работы программы становится не очень заметным. Применяется для особо точных и сложных расчетов, а также для работы с рядом графических программ.

МП обменивается информацией с другими устройствами через порты ввода-вывода. Многие стандартные устройства НГМД, НГМД, принтеры, клавиатура и т.д. имеют постоянно закрепленные за ними порты ввода-вывода.

1.2.3.4. Основные характеристики микропроцессоров

Тактовая частота — общепринятый показатель скорости процессора, измеряется в мегагерцах (1 МГц = 1 млн тактов в секунду) и показывает, сколько операций способен выполнить МП в течение секунды. Чем выше тактовая частота МП (при прочих равных условиях), тем выше его быстродействие. Создание новых поколений процессоров — средство реализовать более высокую тактовую частоту. Сменилось несколько поколений процессоров Intel: 8088, 80286, 80386, 80486, Pentium, Pentium II, Pentium III, Pentium Pro, Pentium MMX и т.д.

В пределах одного поколения чем выше тактовая частота, тем выше производительность и цена МП.

Реально при решении различных задач используются различные наборы операций. Но каждая операция требует для своего выполнения вполне определенное количество тактов. Когда говорят о производительности ПК, нужно иметь в виду, что оценка производительности компьютера всегда приближительна, так как при этом ориентируются на некоторые усредненные или, наоборот, на конкретные виды операций. Поэтому для характеристики ПК обычно указывают тактовую частоту.

Например, МП с частотой 100 МГц обеспечивает выполнение приблизительно 20 млн коротких операций (сложение и вычитание с фиксированной запятой, пересылка информации и др.) в секунду.

Разрядность — максимальное количество разрядов двоичного кода, которые могут обрабатываться или передаваться одновременно, т.е. обрабатываться за один такт (16-, 32- или 64-разрядные МП и т.д.). Чем больше разрядность, тем, при прочих равных условиях, будет больше и производительность ПК.

Архитектура МП — под архитектурой МП понимают принцип действия МП, состав регистров, систему команд, конфигурацию и взаимное соединение основных его узлов.

1.2.3.5. Контроллеры и адаптеры

Все блоки ПК соединяются между собой шиной. *Шина* — это электрическое соединение или группа параллельных соединений, которые обеспечивают обмен информацией между компонентами компьютера.

В состав системного блока входят функциональные блоки, предназначенные для управления работой системных периферийных устройств. Эти модули называются *контроллерами*. Так, в состав системного блока входят контроллеры ВЗУ на МД — контроллеры дисков, контроллер клавиатуры, контроллеры манипуляторов и т.д.

Управление дополнительными периферийными устройствами осуществляют другие функциональные модули, которые называются *адаптерами*, так как большинство этих периферийных устройств служат для преобразования, т.е. адаптации сигналов внешнего интерфейса к системной шине.

Например, адаптеры используются для связи ПЭВМ между собой — *сетевые адаптеры* (т.е. для сопряжения ПК с физическим каналом передачи данных).

Видеоадаптер — устройство, преобразующее набор данных, подлежащих изображению на экране, в видеосигнал, посылаемый монитору по кабелю. Причем, так как скорость воспроизведения изображения на мониторе меньше скорости работы МП, то адаптер монитора может иметь специальную микросхему — память, называемую *видеопамятью*, в которую МП и записывает изображение, а затем уже видеоадаптер будет выводить содержимое видеопамяти (иногда конструктивно это может быть реализовано в форме выделения участка основного ОЗУ). Требования к объему видеопамяти возрастают с увеличением разрешающей способности и количества воспроизводимых цветов. Существует несколько стандартов на видеотракт ПК. Они различаются наборами показателей разрешающей способности, количеством отображаемых цветов. Например, стандарт EGA: разрешаемая способность — 640×350, число цветов — 16 из 64; стандарт VGA, соответственно — 640×480 и 256 из 4096, а стандарт SVGA, соответственно — 1280×1024 и выше и до 16,7 млн цветов.

1.2.3.6. Оперативная память

Память ЭВМ делится на внутреннюю и внешнюю. Внутренняя память включает в себя ОЗУ, кэш-память, ПЗУ.

ОЗУ (RAM — Random Access Memory — память с произвольным доступом) — *оперативное запоминающее устройство*, которое позволяет с большой скоростью записывать и считывать информацию, подготовленную для МП.

ОЗУ — это совокупность специальных электронных ячеек, каждая из которых может хранить 1 байт информации. Эти ячейки нумеруются. Номер ячейки называется *адресом*. Различают адрес ячейки и *содержимое* ячейки памяти.

В ОЗУ хранятся программы и данные, необходимые для решения задачи в данный момент времени, а также для постоянного хранения встроенного блока операционной системы. При включении компьютера в ОЗУ заносятся (загружаются) цепочки байтов, в которых хранится операционная система. Далее в ОЗУ с диска помещаются прикладные программы и данные, которые обрабатываются этими программами. Содержимое многих ячеек памяти (байтов) постоянно изменяется в процессе работы программ (пересылка байтов, арифметические операции и запись результатов в ОЗУ и др.). После загрузки новой программы, если старую «закрыли», прежнее содержимое ОЗУ замещается новым, а после выключения ПК пропадает вовсе.

Оперативная память выпускается в виде микросхем, собранных в специальные модули. Каждый модуль может вмещать 16, 32, 64, 128 Мб памяти. На большинстве материнских плат установлены разъемы для подключения модулей оперативной памяти, таким образом ОЗУ можно легко расширить.

Характеристики ОЗУ:

ОП является *энергозависимой*, т.е. при выключении компьютера содержимое ОЗУ пропадает. Поэтому следует сохранять на магнитном носителе результаты своей работы.

Быстродействие обмена информацией очень большое — соизмеримо со скоростью работы процессора.

Объем памяти — сравнительно небольшой 64 Мб, 128 Мб, 256 Мб, 512 Мб и т.д. Для расширения ОП используют дополнительные блоки памяти — платы расширения памяти.

1.2.3.7. Кэш-память

Кэш-память (так называемая сверхоперативная память) предназначена для согласования скорости работы медленных устройств с более быстрыми. Для «быстрых» компьютеров необходимо обеспечить быстрый доступ к оперативной памяти, иначе МП будет простаивать, т.е. быстродействие ПК уменьшится. Для этого в качестве буфера между ОП и МП используется кэш-память. То есть МП непосредственно обменивается информацией с кэш-памятью, а она уже осуществляет обмен с ОЗУ. Наличие кэш-памяти может увеличить производительность компьютера на 20%. Аналогично кэш-память используется при обмене данными между оперативной памятью и внешним накопителем.

Одна кэш-память — самая быстрая, встроена непосредственно в процессор. Это *кэш-память первого уровня*. Есть еще одна кэш-память, более медленная, но размер ее больше. Это *кэш-память второго уровня*. Находится она вне процессора на материнской плате.

1.2.3.8. Постоянное запоминающее устройство (ПЗУ)

ПЗУ (ROM — Read-Only Memory — память только для чтения) — *постоянное запоминающее устройство* также строится на основе установленных на материнской плате модулей. ПЗУ используется для хранения и чтения неизменной информации, некоторых часто встречающихся величин, стандартных программ и т.п. Как правило, информация в ПЗУ записывается на заводе (фирме)-изготовителе. Например, на системной плате устанавливается специальная микросхема BIOS (Basic Input-Output System) — базисная система ввода-вывода. В этом ПЗУ записаны программы, реализующие функции ввода-вывода, а также программа тестирования компьютера в момент включения питания и ряд других специальных программ. ПК не может изменять программы ПЗУ или добавлять новые.

Эта память является *энергонезависимой*.

1.2.3.9. Системная шина

Системная шина — это основная интерфейсная система компьютера, обеспечивающая сопряжение и связь всех его устройств между собой.

Системная шина обеспечивает три направления передачи информации:

- между МП и основной памятью;
- между МП и портами ввода-вывода внешних устройств;
- между основной памятью и портами ввода-вывода внешних устройств (в режиме прямого доступа к памяти).

По проводам системной шины и осуществляется передача данных.

От типа системной шины, так же как и от типа микропроцессора, зависит скорость обработки информации персональным компьютером. К основным характеристикам системной шины относятся тактовая частота и разрядность канала связи.

Однако системная шина, как основная информационная магистраль, не может обеспечить достаточную производительность для внешних устройств. Для решения этой проблемы в компьютере стали использовать локальные шины, которые связывают МП с периферийными устройствами.

1.2.3.10. Порты

Связь компьютера с различными внешними устройствами осуществляется через порты — специальные разъемы, расположенные на тыльной стороне системного блока.

Порты бывают последовательные и параллельные.

Параллельные порты используются для подсоединения внешних устройств, которым необходимо передавать большой объем информации на близкое расстояние. Через параллельный порт к системному блоку подключается принтер, сканер. Параллельные порты имеют имена LPT1, LPT2, LPT3 (Line PrinTer — линия принтера).

Последовательные порты используются для подключения к системному блоку манипуляторов, модемов. Последовательный порт посылает последовательный поток данных по 1 биту. Последовательная передача данных используется для передачи информации на большие расстояния, поэтому последовательные порты часто называют «коммуникационными». Последовательным портам присвоены имена COM1, COM2, COM3, COM4 (COMmunication — коммуникационный порт).

1.2.3.11. Внешняя память

Внешняя память предназначена для долговременного хранения программ, данных, для хранения большого объема информации, хотя скорость обращения к этой информации ниже, чем к информации оперативной памяти.

Накопители — это запоминающие устройства, предназначенные для длительного хранения и многократного использования информации, т.е. они являются *энергонезависимыми*. Причем, накопитель можно рассматривать как совокупность носителя информации и соответствующего привода-дисковода. Различают накопители на гибких магнитных дисках НГМД (дискеты, флоппи-диски) и накопители на жестких магнитных дисках НЖМД (винчестеры), магнитооптические, оптические диски.

Накопители на гибких магнитных дисках (НГМД) — гибкая пластмассовая основа диаметром 3,5 дюйма. Считывание и запись информации осуществляется расположенной ввне магнитной головкой через окно, вырезанное в конверте (оболочке дискеты). Диск приводится в движение только во время ввода-вывода информации, а в остальное время покоится. Емкость гибких дисков 1,44 Мб и более, а стоимость низкая. Самое главное достоинство таких дисков — они являются съемными.

Информация на НГМД размещается вдоль концентрических окружностей, называемых *дорожками*. Количество дорожек на магнитном диске и их информационная емкость зависят от типа магнитного диска, качества магнитных головок и магнитного покрытия. Каждая дорожка содержит определенное число секторов — под сектором понимают участок дорожки МД, хранящий минимальную порцию информации, которая может быть считана с диска или записана на него. В одном секторе может быть помещено 128, 256, 512 и 1024 байт, но обычно 512 байт данных. Между секторами имеется межсекторный интервал. Разбиение на секторы осуществляется при подготовке диска к работе — *форматировании* (или инициализации) диска.

Накопители на жестких магнитных дисках (НЖМД, винчестеры) — это устройство, как правило, с несъемным носителем. В этих накопителях информация записывается на нескольких жестких дисках с ферромагнитным слоем, при этом работает соответственно группа магнитных головок, собранных в единый блок. Этот пакет дисков непрерывно вращается с большой скоростью, пока компьютер включен. Вся электромеханическая часть заключена в герметичный корпус, такая конструкция позволяет достичь высокой плотности записи и большой скорости считывания/записи информации. Емкость НЖМД достигает десятков гигабайт (40, 60, 80, 120 Гб).

В последнее время для резервного копирования данных получают распространение накопители на сменных жестких дисках, к которым относится накопитель *Jaz*. Емкость используемого катриджа — 1 или 2 Гб.

Накопители на оптических дисках (НОД). Принцип всех существующих ныне оптических дисководов основан на использовании луча лазера для записи и чтения информации в цифровом виде.

По функциональному признаку НОД делятся на три категории:

1) *НОД только для чтения* (без возможности записи). В связи с ростом объемов и сложности программного обеспечения, широким внедрением мультимедиа-приложений получили широкое распространение устройства для чтения оптических дисков — CD-ROM. Чтение информации с таких дисков осуществляется с помощью луча лазера небольшой мощности. При этом обеспечивается высокая надежность хранения информации при многократном считывании. Объем информации на таком диске — 650 Мб и более.

2) *НОД с однократной записью и многократным чтением* (CD-R). На оптический диск в таких дисководы пользователь может один раз записать информацию, но ни стереть, ни перезаписать не удастся. Такие оптические диски удобны для архивирования и там, где важно хранить единожды записанную информацию в неизменном виде.

3) *Перезаписывающие НОД* (CD-RW) — дисководы с возможностью многократной записи информации. Такие диски могут не читаться на некоторых устаревших приводах CD-ROM.

Накопители DVD. Накопитель DVD (Digital Versatile Disc) — цифровой универсальный диск, предназначенный для хранения компьютерной информации большого объема и видео, аудио информации высокого качества (такие носители выводят на новый уровень воспроизведение видео- и аудиоинформации на ПК). DVD-диски имеют емкость от 4,7 Гб (односторонние диски) до 17 Гб (двусторонние, двухслойные). DVD как и НОД могут быть с однократной записью и многократным чтением (DVD-R) и перезаписывающие DVD (DVD-RW).

Флэш-память. В 1984 г. компания Toshiba впервые представила новый вид памяти — флэш-память. Однако массово производить и широко использовать флэш-память начали только несколько лет назад. Сейчас флэш-память постепенно все активнее применяется для хранения и переноса данных. Флэш-память — это особый вид памяти, которая благодаря своим параметрам идеально подходит для мобильных устройств:

- флэш-память не содержит механически движущихся частей (как обычные жесткие диски или CD), а поэтому потребляет значительно меньше энергии;
- флэш-память допускает перезапись (изменение) хранимых в ней данных. В зависимости от типа флэш-памяти возможна перезапись информации от 10000 до 1000000 раз;
- информация, записанная на флэш-память, может храниться очень длительное время (более 20 лет) и способна выдерживать значительные механические нагрузки (в несколько раз превышающие предельно допустимые для обычных жестких дисков);
- флэш-память очень компактна. Размер флэш-карты составляет от 20 до 40 мм в длину и ширину, а толщина флэш-карты — до 3 мм;
- флэш-память обладает достаточной емкостью — до 2 Гб.

Флэш-карта может быть вмонтирована в мобильное устройство, а может быть переносной и использоваться в нескольких устройствах. Например, информацию, записанную на флэш-карте цифрового фотоаппарата, можно прочитать на компьютере.

Существует несколько типов карт флэш-памяти. На данный момент, по оценкам аналитиков, наиболее распространенными типами флэш-карт являются CompactFlash и SD/MMC. CompactFlash является сейчас наиболее универсальным типом флэш-карт и может использоваться во многих устройствах. CompactFlash, в частности, используется во многих цифровых фотоаппаратах.

Магнитооптические диски. Информация на магнитооптических дисках хранится на магнитном носителе, защищенном прозрачной пленкой, а чтение и запись осуществляются с помощью луча лазера. Внешне такие диски похожи на НГМД, но емкость таких дисков (да и стоимость тоже) значительно больше — от 128 Мб до нескольких Гигабайт.

Устройства ZIP. ZIP — устройство для записи и считывания информации с магнитных дисков повышенной емкости — до 250 Мб. Подключается такое устройство в порт, параллельный принтеру.

Стримеры. Информация, хранящаяся на НЖМД, из-за физической порчи диска, действия компьютерных вирусов, случайного или преднамеренного уничтожения файлов может быть испорчена или даже уничтожена. Поэтому следует иметь архивные копии и систематически обновлять копии рабочих файлов. Для этого используют стримеры — наиболее дешевые устройства для записи информации на кассеты с магнитными лентами. Емкость используемых кассет до нескольких десятков Гбайт. Стримеры, как правило, имеют собственные средства сжатия данных.

1.2.3.12. Устройства ввода информации в компьютер

Клавиатура — устройство для ввода в компьютер текстовой и цифровой, а также некоторой управляющей информации. Клавиатура бывает:

- обычная;
- эргономичная (как бы «разломанная» надвое);
- на инфракрасных лучах, без подключения к системному блоку, управление такой клавиатурой — дистанционное.

Манипуляторы — устройства управления курсором (координатно-указательные устройства):

- *мышь* (кроме обычных выпускаются «оптические мыши» — более сложное и дорогое, требующее специального планшета, но более надежное и долговечное, и «беспроводные мыши», питание которых осуществляется от батареек, а радиус действия — несколько метров);
- *джойстик* — рычаг, установленный на соответствующем корпусе, обеспечивает перемещение курсора на экране;
- *трекбол* — манипулятор в форме шара. Для управления курсором нужно вращать этот шар, т.е. не требуется поверхность для передвижения по ней устройства (как для мыши), поэтому такой манипулятор используется в портативных ПК;
- *трекпойнт* — маленький джойстик, который размещается обычно в центре клавиатуры и управляется нажатием пальца;
- *тачпад* — площадка, чувствительная к нажатию пальца;
- *световое перо* — устройство, представляющее собой ручку с фотоэлементом внутри, для указания точки на экране или формирования изображений.

Сканер — устройство для считывания в компьютер графической и текстовой информации. Сканеры бывают:

- *настольные*, которые, в свою очередь, подразделяются на:
 - *планшетные* — когда неподвижное изображение сканирует подвижная камера, т.е. возможно сканировать переплетенные документы, книги, журналы;
 - *страничные* — отдельные листы бумаги протягиваются через устройство, а сканирующие головки остаются на месте. Такие сканеры занимают мало места и оснащаются автоподачей листов;
- *ручные*, когда пользователь проводит сканирующей головкой по соответствующему изображению. Современные ручные сканеры обеспечивают автоматическую «склепку» вводимого изображения. Такие сканеры портативны, дешевле настольных, но работают менее точно;
- *штрих-сканеры* — разновидность ручных сканеров, предназначены для ввода данных, закодированных в виде штрих-кода.

Графические планшеты (дигитайзеры) — устройства для ввода в компьютер контурных изображений (например, для ввода чертежей), автоматизируя их создание. Представляет собой наклонную рабочую поверхность, панель управления и специальное перо для формирования изображения.

Цифровые фото- и видеокамеры, которые формируют изображение сразу в компьютерном формате.

Микрофон для ввода звуковой информации.

1.2.3.13. Устройства вывода информации

Мониторы (дисплеи) — устройства визуализации (отображения) текстовой и графической информации на экране.

По физическим принципам формирования изображения различают:

- *дисплеи на базе электронно-лучевой трубки* (используются в стационарных ПК);
- *дисплеи с жидкокристаллическими экранами* (используются в портативных и стационарных ПК).

Возможность ПК по отображению информации определяется характеристиками дисплея и его адаптера. Так как скорость воспроизведения картинки на экране меньше скорости работы микропроцессора, то любой адаптер содержит специальную память, называемую *видеопамятью*, в которую микропроцессор и записывает информацию, воспроизводимую в дальнейшем видеоадаптером на экране. Объем видеопамяти позволяет в текстовом режиме организовать в ней несколько страниц, что ускоряет смену текстового изображения на экране.

Принтеры — печатающие устройства, т.е. устройства для вывода информации на бумажный носитель или специальные прозрачные пленки. По технологии печати различают:

- *матричные принтеры*, печатающим элементом которых является головка с иглками (с 9, 18, 24 иглками). При работе принтера удар этих головок через специальную красящую ленту формирует символ на бумаге;
- *струйные принтеры* — принтеры, в которых печатающая головка через сопла (количество их может быть несколько сотен) разбрызгивает специальные чернила на бумагу;
- *лазерные принтеры* — принтеры, изображение в которых формируется лазерным лучом.

Основные характеристики принтеров:

- *разрешение* — насколько мелкие детали изображения может передавать принтер. Разрешение измеряется в dpi — в точках на дюйм. В современных принтерах разрешение от 600 dpi и выше;
- *количество цветов*. Любой цвет обычно получают путем смешивания нескольких основных цветов, к ним добавляется черный цвет, который получить сложно. Различают модели принтеров, в которых количество цветов, используемых для смешивания, равно трем или шести (три основных и три дополнительных для изображения светлых тонов);
- *скорость печати* — обычно количество страниц, печатаемых в минуту (до 12 страниц в минуту);
- *ресурс одной заправки* — сколько страниц можно напечатать, используя одну заправку.

Графопостроители (плоттеры) — устройства для вывода на бумагу чертежей. Плоттеры в зависимости от вида бумажного носителя бывают двух типов:

- *планшетного типа*, которые работают с листами бумаги;
- *барabanного типа*, выводящие чертежи на рулоны бумаги.

Синтезаторы звука — электронные генераторы звука, синтезаторы речи.

Модем — это устройство, которое следует выделить отдельно. Оно предназначено для обмена информацией между удаленными компьютерами через каналы телефонной связи и выполняет функции модуляции и демодуляции сигналов — преобразует аналоговые сигналы телефонных линий в цифровые биты, и наоборот. Модемы исправляют ошибки, возникающие при передаче данных по телефонным каналам. Модемы бывают во внешнем и во внутреннем исполнении.

1.2.4. ПОНЯТИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Программное обеспечение (Software) является необходимой составной частью любой ЭВМ.

Под *системой программного обеспечения* понимается совокупность программ и правил вместе со всей связанной с этими компонентами документацией, позволяющая выполнять определенный круг задач как по организации работы самого компьютера, так и по решению конкретных задач.

Стремительное развитие вычислительной техники и расширение сферы ее применения привели к интенсивному развитию программного обеспечения, в настоящее время речь идет о десятках и сотнях тысяч программных пакетов (рис.3).

В зависимости от функций, выполняемых различными элементами программного обеспечения (ПО), его можно разделить на две группы — базовое (системное) ПО и прикладное ПО.

К *системному ПО* относятся программы, предназначенные для организации и контроля вычислительного процесса, для управления распределением ресурсов во время функционирования вычислительной системы и для автоматизации разработки алгоритмов и программ.

Прикладное ПО — совокупность программ для решения задач из различных сфер человеческой деятельности.

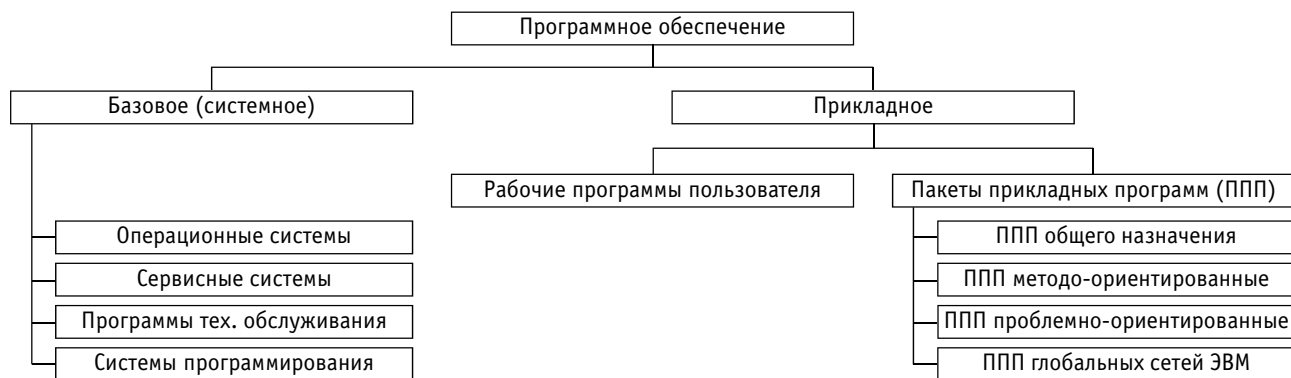


Рис.3. Классификация программного обеспечения

1.2.4.1. Системное программное обеспечение

Операционные системы (MS DOS, UNIX, OS/2, Windows 95, Windows 2000, Windows NT и т.д.) — обязательное дополнение ЭВМ, они предназначены для планирования и управления вычислительными ресурсами ЭВМ, организуют выполнение пользовательских программ и взаимодействие пользователя с компьютером.

Сервисные системы расширяют и дополняют пользовательский и программный интерфейс операционной системы:

Оболочки операционных систем повышают уровень пользовательского интерфейса за счет «меню» и использования функциональных клавиш, а также предоставления дополнительных возможностей. Например, широко известны оболочки ОС — Norton Commander, PC Tools, PC Shell.

Интерфейсные системы, в основном графического типа, позволяют пользователю при работе манипулировать образами, а не символами, причем с использованием «мыши». Они являются многооконными, в отличие от большинства операционных систем. Примером может служить система Windows 3.1.

Утилиты — обслуживающие программы, которые предоставляют пользователю сервисные услуги, их отличие от программ-оболочек: оболочки — универсальны, а утилиты — специализированы. Утилиты — программы вспомогательного назначения, предоставляют дополнительные услуги, реализующие такие функции, как:

- обслуживание магнитных дисков (например, UnErase — для восстановления ошибочно удаленных файлов и каталогов);
- создание и обновление архивов, как со сжатием, так и без сжатия (например, утилиты PKZIP, PKUNZIP, архиватор-разархиватор ARJ);
- тестирование аппаратных программных средств;
- защита от компьютерных вирусов и т.д.

Программы технического обслуживания — предназначены для облегчения тестирования оборудования и поиска неисправностей, они являются инструментом специалистов по эксплуатации аппаратной части компьютеров.

Системы программирования — средства для написания программ:

- **машинно-зависимые программные средства** — ориентированы на конкретную вычислительную машину, это языки машинных кодов, языки типа Ассемблер;
- **машинно-независимые программные средства** — не зависят от вычислительной машины, это алгоритмические языки высокого уровня, например, Фортран, Паскаль, Си. На язык машинных кодов программы, написанные на алгоритмических языках, переводятся специальными программами-трансляторами.

1.2.4.2. Прикладное программное обеспечение

Прикладное программное обеспечение, в свою очередь, можно разделить на рабочие программы пользователя и пакеты прикладных программ.

Рабочие программы пользователя создаются каждым пользователем индивидуально с использованием средств программирования в соответствии с правилами той операционной системы, под управлением которой работает вычислительная система.

Пакеты прикладных программ (ППП) являются самым многочисленным классом программных продуктов. Они выполняют обработку информации различных предметных областей.

Пакет прикладных программ — комплекс взаимосвязанных программ для решения задач определенного класса конкретной предметной области.

По своему назначению ППП делятся на:

- ППП общего назначения;
- методо-ориентированные ППП;
- проблемно-ориентированные ППП;
- ППП глобальных сетей и др.

ППП общего назначения автоматизируют типовые процедуры взаимодействия пользователя с компьютером при обработке информации. Сложилось несколько типов ППП данного класса:

- *текстовые процессоры* (редакторы);
- *графические редакторы*;
- *табличные процессоры* для обработки данных, представленных в виде таблиц;
- *системы управления базами данных (СУБД)* — для накопления данных в виде взаимосвязанных файлов на магнитных носителях и поиска данных по запросам;
- *интегрированные системы*, позволяющие обрабатывать текстовую информацию, таблицы, формировать базы данных и отображать рассчитанные зависимости в виде графиков;
- *сетевые пакеты*, дающие возможность пользователю работать с ресурсами вычислительной сети.

Методо-ориентированные ППП реализуют какие-либо экономико-математические методы решения задач. Например, методы математического программирования, математической статистики, сетевого планирования и управления и др.

Проблемно-ориентированные ППП предназначены для решения какой-либо задачи в конкретной функциональной области. Например, ППП бухгалтерского учета (1С:Бухгалтерия, БЭСТ и др.), банковские ППП (RS-BANK, Диасофт-БАНК и др.) и т.д.

ППП глобальных сетей обеспечивают удобный, надежный доступ пользователей к распределенным общесетевым ресурсам, базам данных т.д. Стандартные ППП используют для организации электронной почты, телеконференций, электронной доски объявлений и др. Например, средства доступа и навигации в сети — Netscape Navigator, Explorer.

Контрольные вопросы

1. Что относится к техническим средствам информационных систем?
2. Что вы можете сказать о тенденции изменения организации решения задач на ЭВМ?
3. Охарактеризуйте каждый класс ЭВМ: суперЭВМ, мейнфреймы, малые ЭВМ и микроЭВМ.
4. Назовите важнейшие технико-эксплуатационные характеристики ЭВМ.
5. Какие ЭВМ можно отнести к классу персональных ЭВМ?
6. Назначение основных функциональных блоков ПК.
7. Что такое микропроцессор, его состав и какие функции он выполняет?
8. Характеристики микропроцессора.
9. Какие типы компьютерной памяти вы знаете? Их отличия.
10. Что такое ОЗУ, кэш-память?
11. Назначение ПЗУ.
12. В чем смысл форматирования диска?
13. Проанализируйте преимущества и недостатки НГМД, НЖМД и CD.
14. Как вы понимаете модульность структуры ПЭВМ?
15. Что такое контроллер, адаптер?
16. Перечислите и охарактеризуйте устройства ввода-вывода информации.
17. Что такое программное обеспечение?
18. Что входит в системное и прикладное программное обеспечение?
19. Что такое пакет прикладных программ?
20. Чем отличаются проблемно-ориентированные ППП от методо-ориентированных ППП?

Литература

1. Евдокимов В.В. и др. Экономическая информатика: Учебник для вузов. — СПб.: Питер, 1997. — С.35–66, 93–109.
2. Информатика. Базовый курс / С.В.Симонович и др. — СПб.: Питер, 2001. — С.39–91.
3. Информатика: Учебник / Под ред. проф. Н.В.Макаровой. — М.: Финансы и статистика, 1998. — С.119–200, 297–334.
4. Фигурнов В.Э. IBM PC для пользователя. — М.: ИНФРА-М, 1995. — С.11–81.
5. Экономическая информатика / Под ред. П.В.Конюховского и Д.Н.Колесова. — СПб.: Питер, 2000. — С.32–86.

ТЕМА 1.3. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПК. ПРОГРАММЫ ПАКЕТА MICROSOFT OFFICE

1.3.1. ОПЕРАЦИОННАЯ СИСТЕМА WINDOWS

Компьютер выполняет определенные действия в соответствии с какой-либо программой. И при работе пользователя на ПК часто возникают одни и те же операции:

- просмотреть содержимое магнитного диска (МД);
- скопировать программы с одного МД на другой;
- удалить информацию с МД;
- записать программу на МД;
- запустить какую-либо программу на выполнение (т.е. выполнить программу) и т.д.

Эти операции используются очень часто при работе с различными программами. Поэтому такие типовые операции были выделены и реализуются с помощью специализированных программ, которые организуют работу устройств и не связаны со спецификой решаемой задачи. Такие программы называли операционными системами.

Операционная система (ОС) — это комплекс специальных программных средств, предназначенных для управления загрузкой, запуском и выполнением других (пользовательских) программ, а также для планирования и управления вычислительными ресурсами ЭВМ.

В 80–90-е годы одной из самых популярных операционных систем была операционная система, разработанная фирмой Microsoft, — MS DOS. Несмотря на то, что в среде MS DOS работали тысячи программ, разработка таких программ была связана со значительными трудностями. Программистам приходилось либо самим разрабатывать средства для создания диалогового интерфейса (меню, запросов, окон и т.д.), либо использовать библиотеки программ. И крупным фирмам приходилось содержать множество сотрудников, занятых разработкой программ графического интерфейса, поддержки десятков типов мониторов и сотен типов принтеров. Это увеличивало сроки и финансовые расходы на создание и сопровождение программ, замедляло развитие всей отрасли разработки программного обеспечения.

В отличие от компьютеров фирмы IBM, для ПК типа Macintosh фирмы Apple была разработана операционная система, которая представляла пользователям удобный графический интерфейс, средства взаимодействия с внешними устройствами.

В 1985 г. и фирма Microsoft выпустила собственную операционную среду Windows. В 1987–1989 гг. стали появляться и удобные программы, работающие в среде Windows — Microsoft Word для Windows, Excel.

Windows предлагает пользователю *оконный интерфейс*, когда каждой выполняемой программе отводится *экранное окно*, которое может занимать часть экрана или весь экран. То есть появилась возможность одновременного использования нескольких программ. Причем информацию из окна одной программы в окно другой программы можно было передавать (через копирование) при помощи *буфера обмена*.

Элементы окон, кнопок, значков для Windows-приложений стандартизированы.

Для подготовки документов, содержащих текст, Windows позволяет использовать масштабируемые шрифты, применяемые как для экранного вывода, так и для распечатки на принтере. Благодаря этому в процессе подготовки документа можно видеть на экране практически то же, что будет получено на бумаге — так называемый принцип WYSIWYG (What You See Is What You Get).

Выпуск графической операционной оболочки Microsoft Windows 3.0 стал главным событием 1990 г. на программном рынке. В 1992 г. появилась версия Windows 3.1. Интерфейс был улучшен, в частности были усилены возможности управления экранными объектами мышью (drag-and-drop — метод перетаскивания).

Что же пользователь получает при использовании Windows и Windows-приложений?

1. *Единый пользовательский интерфейс*, т.е. программисты не изобретают собственные средства для создания пользовательского интерфейса.

2. *Многозадачность* — Windows обеспечивает возможность одновременного выполнения нескольких задач и простого переключения с одной задачи на другую.

3. *Совместимость с DOS-приложениями*, т.е. для выполнения DOS-программ, как правило, нет необходимости выходить из Windows.

4. Средства обмена данными:

- *буфер обмена данными*;
- *динамический обмен данными* между приложениями (DDE — Dynamic Data Exchange) — одна программа может использовать данные, созданные другой программой;
- *механизм связи и внедрения объектов* (OLE) является усовершенствованием средств DDE. Здесь приложение, использующее данные, может запустить программу, с помощью которой были созданы «внедренные» данные, для их изменения.

5. *Поддержка масштабируемых шрифтов*.

6. *Удобство поддержки устройств*. Для подключения к ПК любого нового устройства достаточно установить драйвер этого устройства, предназначенный для Windows, после чего все Windows-приложения смогут работать с этим устройством.

7. *Поддержка мультимедиа*. При подключении соответствующих устройств Windows может воспринимать звуковую информацию от микрофона, компакт-диска, выводить звуки и движущиеся изображения.

Принято считать версии Windows 3.0 и Windows 3.1, предшествующие Windows 95, *операционными оболочками*: т.е. программами, облегчающими работу с MS DOS. Загрузка этих версий Windows производится после загрузки MS DOS соответствующей командой.

Новая версия Windows 95 была принципиально новой операционной системой фирмы Microsoft с колоссальным числом особенностей и возможностей. Windows 95 заменила MS DOS, так как она включала в себя все, что ей нужно от MS DOS, и могла запускать программы MS DOS так, как будто это программы Windows.

В переводе с английского Windows — это «окна». Программа называется так не случайно — при работе с этой программой используется так называемый *многооконный графический интерфейс*. Пользователь общается с компьютером посредством *графических символов*, т.е. маленьких картинок (которые называются «*пиктограммами*» или «*иконками*»). Причем в Windows можно выполнять несколько программ одновременно, т.е. в любой момент можно перейти к работе с программой одного из открытых окон.

Windows — это высокопроизводительная, многозадачная и многопоточная операционная система с расширенными сетевыми возможностями.

Работа над несколькими задачами возможна благодаря новым программам, которые обеспечивают:

- *вытеснение*, т.е. операционная система в любой момент может прервать их выполнение и переключиться на другую задачу;
- *отдельное адресное пространство*. Приложения выполняются в своей защищенной области памяти, что делает невозможным нарушение их целостности со стороны других программ;
- *поддержку потоков*. Обеспечивается многозадачность в пределах одного приложения (т.е. приложения могут одновременно запускать несколько потоков);
- *технология OLE*.

Все эти средства операционной среды Windows сохраняются и в последующих версиях ОС Windows. Кроме того, в новые версии входят программы, применение которых повышает производительность компьютера, надежность его работы, в том числе в вычислительных сетях.

1.3.2. ОРГАНИЗАЦИЯ ДАННЫХ НА МАГНИТНЫХ НОСИТЕЛЯХ

Файловая система — часть операционной системы, управляющая размещением и доступом к файлам и каталогам на магнитных дисках.

При этом учетной единицей информации в операционной системе служит файл.

Файл — логически связанная совокупность данных, для размещения которой во внешней памяти выделяется именованная область.

Любые действия с информацией осуществляются над файлами: запись на МД, вывод на экран, печать и т.д.

На диске файл не требует для своего размещения непрерывного пространства, а занимает обычно свободные кластеры в разных частях магнитного диска.

Кластер является минимальной единицей пространства на магнитном диске, которое может быть отведено файлу.

Самый маленький файл занимает 1 кластер, большие файлы — несколько десятков кластеров.

Кластер — группа смежных секторов. Кластер для гибкого МД — 1 сектор (512 байт) или 2 сектора (1 Кбайт), кластер для жесткого МД — 4, 8, 16 секторов. Кластеры пронумерованы.

Желательно, чтобы кластеры, выделенные для хранения файла, шли подряд, так как это позволяет сократить время его поиска. Однако эти кластеры могут находиться и в разных местах магнитного диска — в виде отдельных фрагментов в свободных на момент записи на диск кластерах. В этом случае говорят, что файл *фрагментирован*.

Для организации доступа к файлу операционная система должна иметь сведения о номерах кластеров, где размещается каждый файл, для этого предназначена FAT-таблица.

При обращении к файлу сначала производится обращение к ячейке FAT-таблицы, адрес которой определяется первым номером, хранящимся в записи о файле. В этой ячейке указан номер второго кластера этого файла и т.д. — в конце цепочки находится код конца (FFF или FFFF).

Если файл фрагментирован, то его можно с помощью специальной программы *дефрагментировать*.

К файлу обращаются с помощью имен и спецификаций.

Каждый файл должен иметь *имя*. В зависимости от типа данных различают разные *типы файлов*: текстовые файлы, программные файлы, графические файлы и т.д. Для обозначения типа файла его имя может дополняться так называемым *расширением*. Имя файла и расширение, разделенные точкой, образуют *полное имя файла*. Например, полное имя файла **lec.doc**. Здесь имя файла **lec**, а расширение **doc**. Если к этому имени файла добавить обозначение дисководов, например, **A:\lec.doc**, то тем самым мы уточняем, что этот файл находится на дисковом дисковом A.

Если нужно выполнить какие-то команды с группой файлов (например, выделить или удалить), то можно использовать так называемые шаблоны имен файлов.

Шаблон имени файла — специальная форма, в которой в полях имени и типа файла используются символы ? и *. Символ ? позволяет «замаскировать» любой один символ в имени или расширении файла (поэтому шаблон иногда называют «маской»), символ * обозначает любое число любых символов в имени или расширении файла. Например, шаблон ***.doc** обозначает все файлы с любым именем и с расширением **doc**.

На одном диске могут храниться десятки и даже сотни файлов. Чтобы работать с конкретным файлом, нужно знать, где он находится на диске, его адрес, размер в байтах. Эта информация обо всех файлах регистрируется при их создании в специальном *файле-папке*.

Папка, как любой файл, имеет свое имя, но без расширения.

В одной папке не может быть зарегистрировано два файла с одинаковыми полными именами.

В Windows принята так называемая *иерархическая* или *древовидная* структура папок. На любом магнитном диске обязательно имеется главная, так называемая *корневая папка*. Она создается при форматировании диска и не может быть удалена средствами ОС. В корневой папке (она обозначается символом «\») регистрируются файлы и могут быть зарегистрированы папки, называемые *папками 1-го уровня*. В любой папке 1-го уровня могут быть зарегистрированы также файлы и *папки 2-го уровня* и т.д.

В Windows принят следующий принцип организации доступа к файлу: чтобы перейти из главной папки в папку *n*-го уровня, нужно обязательно пройти через все предыдущие папки внешнего уровня. Это делается с помощью указания *маршрута* или *пути* к файлу.

Путь к файлу — цепочка соподчиненных папок, которую необходимо пройти по иерархической структуре папок, разделенных символом «\».

Полная спецификация файла — это:

<Имя дисковода>:\<путь>\<полное имя файла>

Какие возможности предоставляет такая структура папок пользователю?

- возможность выделять в отдельные папки файлы определенного вида (например, все текстовые файлы или все бухгалтерские данные);
- возможность для каждого пользователя иметь свою папку с индивидуальной структурой папок;
- возможность при создании файлов не заботиться о том, есть ли в других папках файлы с такими же именами, так как идентификация файлов происходит по полным спецификациям файлов, включающим маршрут к файлу.

Например (рис.4), полная спецификация файла **f1.doc** из корневой папки — **f1.doc**, из папки GR01 — **\GR01\f1.doc**, а из папки GR02 — **\GR02\f1.doc**.

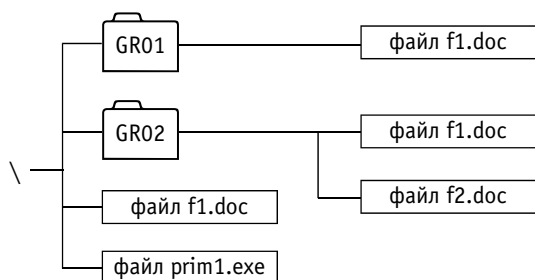


Рис.4. Пример файловой структуры

1.3.3. ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС WINDOWS

Современный пользовательский интерфейс Windows удобен и нагляден. Плоскость экрана применительно к Windows называется «*рабочим столом*». Это фоновая область экрана, на которой размещаются все окна.

На рабочем столе находятся *значки* (*пиктограммы* или *иконки*), каждый значок соответствует прикладной программе, документу (файлу данных) или *папке*. Наиболее близким к понятию «папка», хотя и не исчерпывающим его содержание, является термин «каталог». Внутри одной папки может находиться другая, в ней еще одна папка и т.д.

Windows изменяет способ работы на компьютере: например, чтобы что-то выполнить в Windows, надо выбрать соответствующий значок и щелкнуть левой кнопкой мыши два раза подряд так быстро, чтобы получился «*двойной щелчок*».

Следует заметить, что, несмотря на то, что большинство команд в Windows может выполняться с помощью клавиатуры, в первую очередь эта операционная система ориентирована на работу с манипулятором «мышь». Windows рассчитана на работу с двумя кнопками. Как правило, по умолчанию активной считается левая кнопка (хотя при желании активной можно сделать и правую кнопку).

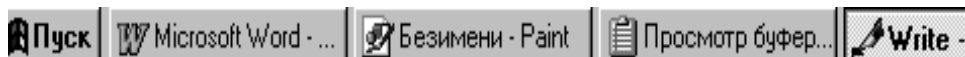
На рабочем столе можно увидеть так называемые *ярлыки*. Значки ярлыков отличаются от значков оригинальных объектов элементом со стрелочкой в левом нижнем углу значка. Ярлыки обеспечивают быстрый доступ к наиболее часто используемым файлам, папкам, устройствам. Для одного и того же документа или программы можно создать сколько угодно ярлыков и разместить их на рабочем столе, в одной или в нескольких часто используемых папках.



На рабочем столе всегда находятся, по крайней мере, две папки: *Мой компьютер*, которая «содержит» объекты, соответствующие всем устройствам компьютера, и *Корзина*, в которую попадают удаленные файлы.

Папка *Корзина* выполняет все функции настоящей корзины для бумаг для временного хранения удаленных файлов и папок. Причем по мере накопления в *Корзине* удаляемых объектов пиктограмма *Корзины* изменяется («наполняется бумагой»). В любой момент можно просмотреть содержимое *Корзины* (как любой папки), а при необходимости удаленные в *Корзину* объекты можно «вернуть на старое место» или же освободить *Корзину* частично или полностью.

Как правило, вдоль нижней границы экрана располагается *панель задач*. Слева на ней находится кнопка *Пуск* (Start). Нажатие кнопки Пуск открывает *главное меню*, содержащее команды доступа к прикладным и служебным программам, системе помощи Windows и находящимся в работе документам.



Панель задач является важнейшим элементом интерфейса Windows. Как только открывается окно с любой программой, это приводит к появлению на панели задач *кнопки*, соответствующей этой программе. По завершении работы программы ее кнопка исчезнет. Использование кнопок панели задач — наиболее удобный способ переключения между программами: чтобы переключиться с окна на окно, достаточно щелкнуть на соответствующей кнопке с именем окна в панели задач.

Справа на панели задач находятся *индикатор*, отображающий текущий язык интерфейса (русский или английский), и *часы*.



1.3.3.1. Работа с окнами

Как уже было сказано, Windows предлагает пользователю многооконный интерфейс, т.е. для каждой выполняемой программы отводится свое экранное окно на рабочем столе. Окна порождаются и загруженными программами, и самой Windows (открытые папки, окна диалога и т.д.).

В Windows имеется два типа окон: окна прикладных программ и окна документов.





Окна прикладных программ представляют собой окна, в которых содержатся текущие, т.е. запущенные на выполнение, программы. Поэтому их иногда называют также *программными окнами*. Окна прикладных программ могут перемещаться в любое место на экране.

Окна документов используются при работе прикладных программ, они всегда находятся внутри программного окна и могут перемещаться только в пределах «своего» окна для прикладных программ.

Практически всякое окно содержит элементы для изменения его размеров и перемещения по экрану с помощью мыши или клавиатуры.

Окно можно развернуть на полный экран или свернуть до минимальных размеров. Свернутое окно не будет видно на рабочем поле, но кнопка для этого окна на панели задач останется, т.е. в любой момент можно перейти в это окно и оно вновь появится на рабочем столе. Размер развернутого или свернутого окна нельзя динамически изменить, можно лишь восстановить промежуточный размер окна.

Если окно закрывается, то оно исчезает с рабочего поля и убирается соответствующая кнопка на панели задач, т.е. заканчивается работа с данным окном (и соответствующей программой).

В правой части строки заголовка окна имеются кнопки для указанных операций:  — *кнопка свертывания* окна до минимальных размеров,  — *кнопка развертывания* окна на весь экран,  — *кнопка восстановления* первоначального размера окна и  — *кнопка закрытия* окна.

1.3.3.2. Главное меню

При нажатии кнопки ПУСК появляется *главное меню* Windows, содержащее несколько пунктов.

- Пункт **Программы** открывает иерархическое меню доступных прикладных и служебных программ. В меню *Программы* особое внимание нужно обратить на подменю *Стандартные* и *Проводник*.

- Пункт **Документы** вызывает меню, в котором накапливаются имена последних документов, открывавшихся пользователем. Если Вы загрузили документ из окна папки, то его имя автоматически появится в списке этого пункта. После этого его можно будет открывать, просто щелкнув мышью на этом пункте в меню *Документ*.

- Пункт **Настройка** содержит команды: *Панель управления* (для настройки устройств и режимов работы), *Принтеры* (для установки драйверов принтера и управления печатью) и *Панель задач* (для настройки меню *Пуск* и *Документы* и параметров панели задач).
- Пункт **Поиск** вызывает программу для поиска данных, например, папок или файлов.
- Пункт **Справка** загружает справочную систему о возможностях Windows и о работе с ней.
- Пункт **Завершение работы** вызывает диалоговое окно «Завершение работы с Windows» для корректного завершения работы. Нельзя завершать работу с Windows обычным выключением или перезагрузкой, так как это может привести к потере данных.

1.3.3.3. Работа с объектами

Ориентироваться в сложной системе папок и устройств компьютера помогают программы *Мой компьютер* и *Проводник*. Обе программы выполняют в основном одни и те же функции, но несколько по-другому. Некоторые пользователи предпочитают программу *Проводник*, основанную на тексте, другие — программу *Мой компьютер*, базирующуюся на изображении.

В папке *Мой компьютер* находятся значки дисководов компьютера и папки для работы с устройствами компьютера. *Проводник* более удобен, например, для операций копирования и переноса файлов и папок перетаскиванием мышью, потому что он одновременно отображает содержимое текущей папки и структуру всего дерева папок.

При помощи команды *Вид/Упорядочить значки* можно упорядочить значки в окне различным образом: по имени, по типу и т.д.

С помощью соответствующих команд меню *Вид* можно легко изменить формат показа папок и файлов (вариант с крупными значками или для экономии места малыми значками, вывод списка файлов с детализированной информацией или без нее).

Стандартные сведения, касающиеся имени, размера, даты и атрибутов объекта, предоставляются в окне *Свойств объекта*, полученном командой меню *Файл/Свойства*.

Чтобы работать с объектами — папка, файл, их необходимо выделить.

Чтобы выделить несколько объектов (в том числе и несмежных), надо нажать клавишу **Ctrl** и, удерживая ее, выделить все нужные объекты. Группу смежных объектов можно выделить с помощью мыши или, удерживая нажатой клавишу **Shift**, щелкнуть мышью по первому и последнему объекту.

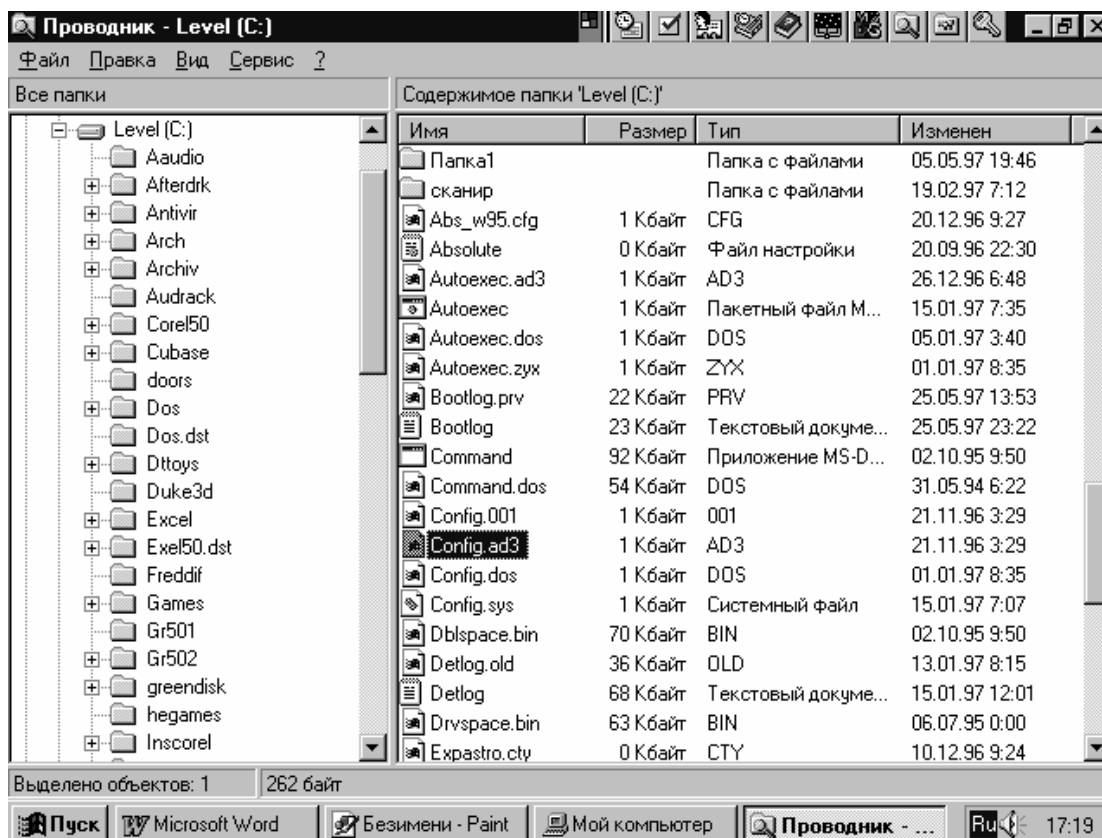
Любая операционная система имеет команды для работы с дисками, папками, файлами. Многие команды можно выполнять, выбирая их из основного меню окон *Мой компьютер* и *Проводник*, или из контекстного (нажимая правую кнопку мыши), или используя определенные клавиши.

Для форматирования дисков используют команду *Форматировать*.

Создать новую папку можно при помощи команды *Файл/Создать/Папка*.

При удалении папок и файлов (командой *Файл/Удалить*) они переносятся в папку *Корзина* (Recycled), в которой хранится журнал удалений. При таком удалении объекты можно вернуть из *Корзины*, для этого достаточно, находясь в папке *Корзина*, выделить необходимые объекты и выполнить команду меню *Файл/Восстановить*. Чтобы удалить файлы безвозвратно, следует дать команду меню окна папки *Корзина* *Файл/Удалить* или *Файл/Очистить корзину* для удаления всех объектов из *Корзины*.

Выделив объекты (папки и файлы), их можно копировать (командой *Правка/Копировать*), переименовывать (командой *Файл/Переименовать*).



1.3.4. ИНСТРУМЕНТАРИИ РЕШЕНИЯ ФУНКЦИОНАЛЬНЫХ ЗАДАЧ

Microsoft Office для Windows представляет собой интегрированный программный комплекс. Это означает, что входящие в него компоненты могут использоваться как отдельно каждый, так и вместе для решения повседневных деловых задач. Истинная многозадачность Windows позволяет легко переключаться между приложениями Office. В состав Microsoft Office входят следующие прикладные программы:

Microsoft Word — универсальный редактор текстов и средство подготовки оригинал-макетов для печати;

Microsoft Excel — табличный процессор — электронные таблицы с мощными средствами анализа данных и построения диаграмм, а также аналитическими функциями;

Microsoft Access — реляционная система управления базами данных с возможностями создания запросов, отчетов;

Microsoft PowerPoint — программа презентационной графики для создания слайдов и мультимедиа-презентаций;

Microsoft Outlook — программа для управления личной и деловой информацией, адресными книгами, дневником и электронной почтой.

Объединив свои лучшие программы в единый прикладной комплекс, компания Microsoft создала универсальное средство для решения задач обработки данных, возникающих в современном бизнесе.

Во всех приложениях Microsoft Office используются стандартные команды, окна диалога и основные операции, в них используются похожие средства форматирования и макроязыки. Приложения проектировались для совместной работы, так что есть, например, возможность легко объединить текст из Word, диаграмму из Excel, информацию из базы данных Access в одной презентации.

Контрольные вопросы

1. Каково назначение операционной системы?
2. Что означает многооконный интерфейс Windows?
3. Преимущества графического интерфейса.
4. Особенности мультизадачного режима Windows.
5. Для чего предназначен буфер обмена?
6. Объясните метод перетаскивания — drag-and-drop.
7. Что представляет собой принцип WYSIWYG?
8. Особенности среды Windows.
9. В чем отличие динамического обмена данными между приложениями (DDE) и механизма связи и внедрения объектов (OLE)?
10. Что такое папка?
11. Для чего служит ярлык на рабочем столе?
12. Для чего нужна папка *Корзина*?
13. Основное назначение панели задач.
14. Назначения кнопки Пуск.
15. Что такое *Проводник*?
16. В чем отличие работы в *Проводнике* от работы с папкой *Мой компьютер*?
17. Как можно выделить несколько объектов?
18. Как можно отформатировать дискету в Windows?
19. Назовите, какие действия можно производить с папками, файлами?
20. Перечислите и охарактеризуйте основные программы, входящие в комплекс MS Office.

Литература

1. Берлинер Э.М., Глазырин И.Э., Глазырина И.Б. Windows 95. Русская версия. — М.: АБФ, 1997.
2. Евдокимов В.В. и др. Экономическая информатика: Учебник для вузов. — СПб.: Питер, 1997. — С.116–123, 175–240.
3. Информатика. Базовый курс / С.В.Симонович и др. — СПб.: Питер, 2001. — С.98–160.
4. Информатика: Учебник / Под ред. проф. Н.В.Макаровой. — М.: Финансы и статистика, 1998. — С.338–343, 487–500.
5. Майнази Н. Windows 2000. — М.: ЛОРИ, 2000.
6. Практикум по экономической информатике: Учебное пособие. Ч.1 / Под ред. Е.Л.Шуримова, Н.А.Тимаковой, Е.А.Е.А.Мамонтовой. — М.: Перспектива, 2000. — С.17–40.
7. Степанова Т.И. Методические указания по курсу «Информатика». Тема: «Основы операционной системы Windows 95». — Новосибирск: НГАЭиУ, 1997.
8. Фигурнов В.Э. IBM PC для пользователя. — М.: ИНФРА-М, 1995. — С.82–92.
9. Экономическая информатика / Под ред. П.В.Конюховского и Д.Н.Колесова. — СПб.: Питер, 2000. — С.87–99, 110–125.

ТЕМА 1.4. ТЕКСТОВЫЕ РЕДАКТОРЫ. ТЕКСТОВЫЙ ПРОЦЕССОР MS WORD

1.4.1. НАЗНАЧЕНИЕ И КЛАССИФИКАЦИЯ ТЕКСТОВЫХ РЕДАКТОРОВ

Обработка текстов, подготовка различного вида документов составляют значительную часть работ, выполняемых в настоящее время на персональных компьютерах (ПК).

Системы подготовки текстов или текстовые редакторы, или текстовые процессоры представляют собой одну из самых популярных категорий прикладных систем на ПК. Объясняется это, во-первых, появлением дешевых ПК и принтеров, что стимулировало их применение в делопроизводстве, журналистике и т.д. Во-вторых, любая работа так или иначе бывает связана с составлением текстовых документов. Текстовые процессоры служат для подготовки самых разнообразных видов текстовой документации — статей, писем, технических описаний, приказов, отчетов и др. При этом обработка текстовых материалов на ПК выполняется быстрее, эффективнее и предоставляет недоступные ранее возможности. Необходимость работы с текстами столь велика и универсальна для различных категорий пользователей, что текстовые редакторы включаются практически во все интегрированные прикладные системы для ПК.

Текстовые редакторы (процессоры) — это специальные сервисные программы, входящие в состав прикладного программного обеспечения, предназначенные для подготовки, редактирования, оформления и печати документов и текстов различных видов с помощью ЭВМ.

При работе с текстовым процессором возможно многократно исправлять его отдельные фрагменты, не вводя заново весь текст, вносить в текст другие материалы, также подготовленные на компьютере и хранящиеся в других файлах (например, схемы, таблицы), печатать на принтере текст в нужном количестве экземпляров. Можно автоматически составлять оглавление документов, проверять правильность написания слов, составлять по заданным текстам словари и т.д.

Развитые текстовые редакторы выполняют следующие функции:

1) *редактирование текста*, а именно:

- удаление, вставку и пересылку символов, последовательностей символов, строк и целых фрагментов текста;
- поиск и замену цепочек символов;
- одновременную обработку различных фрагментов одного или нескольких файлов в различных окнах;
- создание и использование собственных макрокоманд для обработки текста;
- проверку орфографии;
- поиск синонимов;

2) *форматирование текста*, в том числе:

- оформление текста с использованием различных шрифтов;
- управление делением текста на абзацы;
- автоматический перенос слов;
- выравнивание текста по левой, правой границе или по центру;
- многоколонковый набор;

3) *слияние файлов*, т.е. импорт файлов в различных форматах, подготовленных другими программными продуктами;

4) *экспорт файлов*, т.е. перезапись их в другом формате с целью использования в иных программных продуктах;

5) *подготовку текстов к печати* (функции верстки в усеченном виде), среди которых:

- включение в текст иллюстраций (импорт графики);
- предварительный просмотр сверстанных страниц;
- вычерчивание линий;
- подготовка оглавлений, сносок и индексов;

6) *печать текстов* на различных типах принтеров в различных режимах.

По назначению и функциональным возможностям можно выделить следующие основные типы редакторов:

1. **Встроенные редакторы**, которые являются составной частью более сложных программных систем и выполняют простые функции:

- а) по редактированию текстов небольших объемов. Например, встроенный текстовый редактор Norton Commander не может обрабатывать тексты значительной длины, но зато очень быстро загружается, что удобно для внесения оперативных изменений;
- б) по редактированию текстов программ. Они обычно встроены в систему программирования на конкретном языке программирования (Turbo СИ, Turbo Pascal и др.).

2. **Редакторы текстов и документов**. Среди наиболее распространенных в мире редакторов — Microsoft Word, WordPerfect, MultiEdit, Лексикон. Наиболее популярными редакторами являются Microsoft Word и WordPerfect.

3. **Редакторы научных документов**. Если в тексте встречается много сложных формул, то используют редакторы, относящиеся к группе редакторов научных документов: ChiWriter, TCube, TeX. Наилучшим считается редактор TeX, который позволяет очень быстро вводить математические, физические, химические, многоступенчатые формулы. В этом редакторе набор текста с формулами идет быстрее в несколько раз, чем в любом другом.

4. **Издательские системы**. Издательские системы позволяют подготавливать и печатать на лазерных принтерах сложные документы высокого качества, а также выводить тексты на

фотонаборные автоматы. Они используются для оформления больших документов, книг, газет, буклетов. К самым распространенным издательским системам относятся PageMaker, Ventura Publisher.

Подготовка текстовых документов с помощью ПК заключается в последовательном выполнении ряда этапов:

- набор текста;
- редактирование текста;
- печать текста;
- ведение архива текстов.

Каждый этап состоит из выполнения определенных операций. Состав операций зависит от используемого текстового редактора. Как правило, увеличение числа возможных операций делает текстовый редактор, с одной стороны, более универсальным, а с другой — приводит к усложнению его освоения, уменьшению быстродействия, повышению требований к оборудованию ПК.

Созданный в результате выполнения этапа набора текста документ в большинстве случаев подвергается изменениям, связанным с обнаружением ошибок, корректировкой информации, реорганизацией его фрагментов и т.д.

Получение отпечатанного текста является, как правило, главной целью подготовки документа с помощью текстового редактора на ПК.

Этап печати состоит из операций подготовки текста к печати и собственно печати.

К операциям подготовки текста к печати относятся разделение на страницы, нумерация страниц, изменение шрифта, выделение элементов текста при печати, задание заголовка и подножия страниц.

Собственно печать является заключительной операцией подготовки документа. В процессе печати текстовые редакторы позволяют отказаться от печати отдельных страниц документа, а также приостановить или прервать печать в любой момент времени.

Использование новой технологии подготовки текстовых документов с помощью ПК ставит перед пользователем *задачу ведения архива текстов на магнитных дисках*. Ведение архива включает в себя следующие основные операции: просмотр оглавления текстовых файлов, чтение текстов из архива, запись текстов в архив, копирование текстов, удаление их, переименование.

Операции чтения текстов из архива и запись их в архив выполняются специальными командами редактора. Существует возможность записать на магнитный диск (МД) весь текст или его часть.

Запись создаваемого текстового документа на МД рекомендуется производить периодически, несколько раз в течение сеанса работы, с тем, чтобы избежать больших потерь в случае сбоя ПК, отключения питания и т.д.

1.4.2. ТЕКСТОВЫЙ РЕДАКТОР WORD

Редактор Word относится к текстовым редакторам, ориентированным на работу с документами, состоящими из страниц, абзацев, разделов; в структуру документа могут входить таблицы, графические образы, объекты, созданные другими программами.

Вот некоторые возможности редактора Word:

- использование множества различных шрифтов по размерам и начертаниям;
- наличие орфографического словаря и словаря синонимов;
- набор текста в несколько колонок;
- печать верхнего и нижнего колонтитулов;
- включение рисунков (графических файлов);
- возможность импорта как текстов, так и иллюстраций, подготовленных в разных форматах;
- размещение абзацев (например, рисунков) в любом месте страницы (остальной текст при этом может «огнуться» рисунок);
- использование так называемых стилей. Когда наиболее часто используемые виды оформления текста записываются в так называемую таблицу стилей, тогда любому участку текста можно присвоить один из стандартных видов оформления.

В окне редактора Word содержатся стандартные элементы окон приложений Windows. Например, *строка меню* содержит названия подменю команд, предназначенных для работы с документами в Word. Практически все операции в редакторе Word можно выполнить с помощью меню, но другие способы иногда бывают более удобными.

В *панелях инструментов* расположены кнопки, предназначенные для быстрого выполнения некоторых команд. Панель инструментов включает в себя несколько кнопок, каждая из которых соответствует какой-либо команде.

При работе с редактором Word можно устанавливать несколько режимов просмотра окон: режим устанавливается с помощью *кнопки установки режимов просмотра окна* или команды *Вид*.

Режим *Обычный* — позволяет просматривать текст в виде непрерывной длинной колонки, содержащей текст, таблицы и пр. Отображается упрощенный макет документа, поэтому данный режим рекомендуется использовать только при первоначальном вводе текста, но не при окончательном формировании.

Режим *Структура* предназначен для работы со структурой документа, т.е. с заголовками и фрагментами текста. Имеется возможность просматривать только заголовки документа, свернув документ. Часто при редактировании текста возникает необходимость переставлять местами разделы, параграфы и т.п. Делать это очень удобно в режиме просмотра структуры документа.

Режим *Разметка страницы* показывает документ в таком виде, как он выглядел бы распечатанным на принтере. Этот режим удобен для оформления документа, так как в нем видны не только все элементы документа, но и их расположение.

Режим *Главный документ* дает возможность просматривать документ в виде структуры, которая составлена из имен других документов, входящих в него.

Режим *разделения окна документа на две части*, каждая из которых может прокручиваться самостоятельно, позволяет просматривать одновременно две части одного и того же документа. Режим устанавливается с помощью команды *Окно/Разделить*.



1.4.3. РЕДАКТИРОВАНИЕ И ФОРМАТИРОВАНИЕ ТЕКСТА

Редактирование состоит из операций просмотра текста, вставки, замены, удаления и перемещения символов, строк, фрагментов. В процессе редактирования текстовый документ может существенно измениться. В результате вставок и удалений нарушается выравнивание границ текста. С помощью команд форматирования можно выровнять левые и правые границы текста или переформатировать текст в новых границах.

Форматирование — это формирование внешнего вида документа с помощью выбора различных шрифтов, установление размеров букв, интервалов, отступов и полей, создание списков. Работа с редактором и заключается в умелом применении возможностей по форматированию текста.

Рекомендуют сначала ввести большой текст, а затем отформатировать его в едином стиле.

При работе с документом различают форматирование символов, абзацев, страниц.

Форматирование символов позволяет изменить шрифт, начертание, размер символов и расстояние между ними.

Внешний вид букв определяется шрифтом, который выбирается из списка доступных в Windows шрифтов. При выборе шрифта нужно иметь в виду, что шрифты бывают *пропорциональные* (буквы могут занимать разное место, поэтому и строки из равного количества букв могут иметь разную длину), к таким шрифтам относятся Times New Roman Cyr и Arial Cyr, и *непропорциональные* (все буквы занимают одинаковое место, поэтому в разных строках буквы находятся строго друг под другом), к таким шрифтам относится Courier New Cyr, похожий на шрифт печатной машинки.

Размер шрифта задается в пунктах. Один пункт равен $1/72$ дюйма, или 0,353 мм. Допускается размер от 1 до 1638 пункта, при этом можно указать любой размер шрифта в этом диапазоне, даже если такого в списке предлагаемых размеров нет.

При *форматировании абзацев* параметры абзацев устанавливаются командой *Формат/Абзац* или с помощью панели инструментов.

Рассмотрим некоторые приемы форматирования абзацев.

Выравнивание. Невыровненного текста не бывает, но можно задать несколько типов выравнивания: при выравнивании по левому или правому краю начало или конец строки текста прижимается соответственно к левой или правой границе области ввода, а противоположный край получается не выровненным; при выравнивании по центру текст располагается по центру, что применяется при вводе заголовков; при выравнивании по ширине выравниваются оба края текста по границам области ввода. Чтобы длина строк при этом была одинаковой, пробелы могут увеличиваться. Последняя строка выравнивается по левому краю. При форматировании абзацев нужно иметь в виду, что рекомендуется использовать выравнивание абзацев по ширине, так как это придает документам профессиональный вид.

Группа полей *Отступ* позволяет установить отступы абзаца от границ текста.

Группа полей *Интервал* предназначена для задания величины интервалов между строками и между абзацами. Величина интервалов задается в пунктах.

Вкладка *Положение на странице* задает положение абзаца относительно других абзацев и относительно страницы:

- флажок *Запрет висячих строк* запретит присутствие на странице одной строки из абзаца, если в абзаце больше чем одна строка, т.е. как в начале страницы, так и в конце должны будут находиться хотя бы две строки;
- флажок *Не разрывать абзац* запретит размещать абзац на разных страницах;
- флажок *Не отрывать от следующего* заставляет помещать текущий абзац на одной странице с началом следующего.

Форматирование страниц. К форматированию страницы относится: установка полей, размера бумаги, ориентация страницы, создание колонтитулов.

Верхнее, нижнее, левое, правое поля устанавливают расстояние между соответствующими краями бумаги и текста. Изменить поля можно двумя способами: с помощью линейки или с помощью команды *Файл/Параметры страницы*.

По мере увеличения объема текста Microsoft Word автоматически разбивает документ на страницы. По умолчанию Word разбивает документы на страницы после каждого изменения документа. Word определяет, сколько текста поместится на страницу, и вставляет *мягкий разрыв* страницы — *метки-разделители* страниц. При необходимости пользователь может в любом месте страницы вставить жесткий разделитель страницы — *Разрыв страницы*.

Microsoft Word позволяет вставлять в документ так называемые *колонтитулы* — дополнительную информацию, которая помещается в верхнее или нижнее поле на каждой странице. Это могут быть дата, номера страниц, название документа или раздела (номер главы, раздела), рисунки. В верхней части страницы размещается верхний колонтитул, а внизу — соответственно нижний.

В верхний колонтитул обычно включается информация о названии документа или раздела, рисунки.

В обычном режиме просмотра колонтитулы на экране не видны, а в режиме разметки они отображаются светло-серым тоном. Чтобы создать или отредактировать колонтитул, нужно вывести на экран так называемую область колонтитулов. Для этого следует выбрать команду *Вид/Колонтитул*. Появившаяся панель инструментов позволяет вносить необходимые изменения.

Помимо нижнего колонтитула внизу страницы могут размещаться еще и сноски. Сноски применяются в книгах для ссылки на поясняющие сведения или на документ-источник.

Сноска состоит из двух частей — *значка*, вставленного в текст документа, и *текста сноски*. Этот текст может быть помещен в конце страницы (обычные сноски) или в конце документа (концевые сноски).

В обычном режиме просмотра сноски не видны, они отображаются в режиме разметки и в распечатанном документе. Вставляют сноски командой *Вставка/Сноска*.

1.4.3.1. Разбиение текста на разделы

Редактор Word позволяет распространять установки разметки страницы не на весь документ, а на отдельные его части. Чтобы определенные части документа выглядели по-разному, его следует разбить на разделы и отформатировать каждый раздел по-своему. В разных разделах можно по-разному определять число колонок текста, размер полей, расположение и содержание колонтитулов. Нумерация страниц при этом может быть сквозной, а можно пронумеровать страницы в каждом разделе отдельно.

До тех пор, пока в документ не вставлен разрыв раздела, документ будет форматироваться как единый раздел.

Новый раздел оформляется командой *Вставка/Разрыв* и далее опцией *Новый раздел* устанавливают где должен начаться новый раздел — *Со следующей страницы* или *На текущей странице*.

1.4.3.2. Создание оглавления

Редактор Word позволяет упростить процесс работы с заголовками и кроме этого автоматически оформлять оглавление документа. Word имеет специальный режим, в котором при перемещении любого из заголовков будет перемещен и текст, который ему соответствует, а также изменится нумерация всех заголовков. Это же относится и к нумерации рисунков.

Для создания оглавления документа используют понятие — уровни заголовков. Например, первый уровень — это заголовки глав, второй — заголовки разделов в главах, третий — заголовки параграфов в разделах и т.д. Тогда нумерация заголовков будет многоуровневой, например, если глава будет иметь номер «1», то разделы в ней будут пронумерованы «1.1», «1.2», а параграфы в разделах — «1.1.1», «1.1.2» и т.д.

Каждому уровню заголовков ставится в соответствие *стиль заголовка* — *формат, примененный к заголовку*. В Word входит 9 разных встроенных стилей заголовков — «Заголовок 1» — для 1-го уровня, «Заголовок 2» — для 2-го уровня и т.д. При желании эти встроенные стили можно изменять, как обычно (т.е. начертание, размер, шрифт, цвет и пр.) командой *Формат/Стиль/Изменить...*

Команда *Формат/Список* позволяет задать формат автоматической нумерации заголовков.

В оглавление можно включать не все уровни заголовков, для этого, выбрав команду *Вставка/Оглавление и указатели*, в поле *Уровни* нужно указать количество уровней заголовков, включаемых в оглавление. По умолчанию, как правило, в этом поле указано 3 уровня, поэтому будьте внимательны: если у вас есть уровни заголовков больше третьего, то чтобы эти заголовки были включены в оглавление, поле *Уровни* нужно корректировать.

Если после того, как оглавление было уже создано, вы внесли в свой документ коррективы (добавлен текст и в результате изменились номера страниц, указанные в оглавлении, или в документ был вставлен новый заголовок), необходимо обновить оглавление. Для этого, выделив оглавление и нажав функциональную клавишу F9 или щелкнув правой кнопкой мыши, в открывшемся контекстном меню выберите команду *Обновить поле* и в появившемся диалоговом окне *Обновление оглавления* отметьте флажком опцию *Обновить только номера страниц* или опцию *Обновить целиком* (чтобы в оглавление был добавлен новый заголовок).

1.4.3.3. Использование стилей

Стиль — это набор форматизирующих команд, сохраняемый под своим именем для многократного использования, т.е. стиль — это то, как выглядит текст.

Стиль предназначен для внешнего оформления содержимого документа и его абзацев как основной структурной единицы документа.

Стиль имеет *имя*, задавая которое пользователь задает определенный набор параметров форматирования содержимого абзацев.

Стили упрощают задачу оформления документа, обеспечивают согласованность представления отдельных частей документа или документов одного типа.

Стиль используют для форматирования слов или абзацев. Соответственно в редакторе Word различают стили символов и стили абзацев.

Стиль символов включает в себя шрифт, размер и начертание символов. Все это задается параметрами вкладки *Шрифт* команды *Формат/Шрифт*. Стиль символов может быть применен к отдельным символам, словам и выделенным участкам текста.

Стиль абзацев — это стиль символов в абзаце плюс отступы, межстрочное расстояние и выравнивание, а это задается параметрами вкладки *Отступы и интервалы* команды *Формат/Абзац*.

Редактор Word содержит определенный набор стандартных встроенных стилей, перечисленных в окне списка *Стиль* панели инструментов *Форматирование*, которыми можно пользоваться и при необходимости изменять, но любой пользователь может создать свой собственный стиль для символов или абзацев.

Возможности стандартных стилей шире, чем просто форматирование. Например, используя автоматические стили заголовков, потом можно поместить эти заголовки в оглавление.

По умолчанию во всех новых документах, основанных на шаблоне Normal, используется стиль *Обычный*. Текущий стиль виден в окне *Стиль*.

1.4.3.4. Работа со списками

Многие документы, с которыми приходится работать, содержат так называемые списки-перечисления. Для оформления абзацев в таких списках можно использовать цифровую или буквенную нумерацию, а также специальные значки (например, •, ♦) в начале каждого пункта.

Список можно оформлять при вводе текста, а также в уже введенном тексте. Для оформления списка в уже введенном тексте нужно предварительно выделить нужный фрагмент текста.

Для оформления списка используют команду меню *Формат/Список*. А в появившемся выбирается нужная вкладка: *Маркированный*, *Нумерованный*.

Если необходимо поставить в соответствие одному из пунктов списка собственное перечисление, т.е. перечисление второго уровня или аналогично третьего, четвертого и т.д., то используют вкладку *Многоуровневый*. Для понижения уровня можно нажать кнопку *Увеличить отступ* или клавишу **Tab**, а для повышения уровня — кнопку *Уменьшить отступ* или клавиши **Shift+Tab**.

При использовании команды можно менять размер и расстояние символа до текста при создании маркированного списка, начальный номер, формат нумерации и расстояние до текста при создании нумерованного списка, начальный номер, формат нумерации и отступ при создании многоуровневого.

Примеры списков:

Нумерованный	Маркированный	Многоуровневый
1. Швеция	Принтеры бывают:	1. Пользовательский интерфейс
2. Норвегия	• матричные;	1.1. Окна
3. Германия	• струйные;	1.1.1. Элементы окна
4. Польша	• лазерные.	1.1.2. Упорядочение окон
5. Италия		1.2. Рабочий стол
		1.2.1. Панель задач
		1.2.2. Корзина
		1.2.3. Работа с ярлыками
		1.2.4. Мой компьютер

1.4.4. ТАБЛИЦЫ В ТЕКСТОВЫХ ДОКУМЕНТАХ

Редактор Word имеет мощные средства работы с таблицами. Таблицы используют для представления списков, расписаний, бланков, различных финансовых документов.

Таблица представляет собой совокупность ячеек, расположенных в строках и столбцах, которые можно заполнять произвольным текстом или графикой.

Отдельная ячейка может рассматриваться как обычный документ, т.е. для нее доступны операции ввода, редактирования, форматирования текста. При вводе текста ячейка автоматически растягивается по вертикали.

Таблицу можно создать в любом месте документа. Она может размещаться на нескольких страницах. При этом можно таблице назначить заголовок таким образом, что при переходе таблицы на новую страницу заголовок будет автоматически появляться наверху новой страницы.

Работа с таблицами производится с помощью меню команды *Таблица*.

Можно преобразовать в таблицу выделенный текст с помощью команды *Преобразовать в таблицу*. В этом случае текст должен иметь разделители: знак конца абзаца, символ табуляции, запятую и т.п., который отделяет содержимое ячеек одной строки.

После того как таблица создана и заполнена, может потребоваться редактирование таблицы. Редактор позволяет перемещать, копировать ячейки, вставлять или удалять строки и столбцы таблицы, настраивать высоту строк и ширину столбцов, объединять несколько ячеек одной строки в одну ячейку, и наоборот, разбивать одну ячейку на несколько.

Для получения профессионально оформленной таблицы текст в ячейках таблицы можно отформатировать аналогично форматированию текста и абзацев в документе. Каждая ячейка таблицы содержит отдельный абзац или несколько абзацев, поэтому каждую ячейку можно форматировать по-своему.

Ячейки таблицы отделяются друг от друга линиями сетки. С помощью команды меню *Таблица* можно убрать с экрана линии или показать их снова. Сетка отображается только на экране, а при печати этих разделительных линий не будет. Чтобы сделать таблицу более читабельной, можно применить обрамление и заливку ко всей таблице или к выделенным ячейкам, прорисовать сетку внутри таблицы, выделить цветом наиболее важные сведения в таблице.

Чтобы отформатировать всю таблицу достаточно быстро, в пакете существует набор готовых форматов, которые пользователь может использовать для конкретной таблицы.

В таблице можно производить вычисления. Встроенные средства вычислений целесообразно использовать в следующих случаях:

- задача очень проста (вычисление сумм, максимумов, средних);
- числовые данные расположены регулярным образом в строках или столбцах таблицы.

Вычисления производятся с помощью математических формул. Формулы вставляются при помощи команды *Таблица/Формула* и могут содержать числовые константы, имена встроенных функций, адреса ячеек таблицы, соединенных знаками математических операций.

1.4.5. ТЕХНОЛОГИЯ OLE

Windows позволяет передавать информацию из одной программы в другую. Для этого используют:

- *буфер обмена*. Он используется для временного хранения данных и передачи их из одного приложения в другой;

- *динамический обмен данными (DDE)*. Если нужно создать активную связь, используют команду *Правка/Специальная вставка*. С помощью этой команды связывается объект (источник) и документ (адресат). При установлении связи между объектами данные физически продолжают находиться в программе, где они создавались. В составном документе хранится не сам объект, а только ссылка на другой документ с объектом, обеспечивающая визуальное представление объекта при работе. Связывание не увеличивает размер разрабатываемого документа. При необходимости можно будет корректировать данные в источнике. Изменения будут автоматически переноситься в документ-адресат.

- *связь и внедрение объектов (OLE-технология)*, которая дает возможность эффективного обмена данными, разработанными в различных программах и имеющими различный формат, возможность объединить их в составной документ, не заставляя пользователя заниматься преобразованием форматов.

Технология OLE (Object Linking and Embedding — связывание и внедрение объектов) — это технология, которая позволяет включать в текстовый документ любую информацию (объекты) из других программ, например, рисунки, созданные в графическом редакторе, или электронные таблицы из программы Excel. Достаточно дважды щелкнуть мышью на такой таблице в тексте документа Word, чтобы запустилась программа Excel с автоматической загрузкой указанной таблицы для ее редактирования.

В редакторе Word для Windows возможен импорт различных графических объектов.

Для вставки в текст уже существующего рисунка (в виде графического файла в библиотеке рисунков, находящейся в папке **MSOffice\Clipart**) курсор устанавливают в нужное место текста,

далее нужно выбрать команду *Вставка/Рисунок* и, указав в поле *Тип* формат файла, в котором хранится нужный рисунок, выбрать имя файла с этим рисунком. При этом можно предварительно просмотреть рисунок в диалоговом окне команды. При использовании этого метода наличие программы, с помощью которой создавался рисунок не обязательно.

В редакторе Word для ввода формул используют так называемый *Редактор Формул*. *Редактор Формул* — это отдельная программа (Equation), которая может работать совместно с редактором Word. С помощью *Редактора Формул* можно построить сложные формулы, нажимая кнопки с символами на панели инструментов этого редактора и вводя с клавиатуры числа и имена переменных. При этом *Редактор Формул* автоматически настраивает размеры шрифта и форматирование в соответствии с общепринятыми нормами написания формул. Но в процессе работы можно переопределить стандартные стили различных частей формул.

Организационная диаграмма (или оргдиаграмма) изображает иерархию некоторой организации, используя совокупность блоков и соединительных линий. Организационная диаграмма может отображать структуру компании, отдела, нескольких групп организаций и т.п., которые подчинены одна другой по принципу сверху вниз.

Для работы с организационными диаграммами предназначен модуль MS Organization Chart, входящий в комплект Microsoft Office.

Редактор Word позволяет на основе содержащихся в таблице числовых данных построить диаграмму. Для этого используется программа Microsoft Graph. Этот модуль позволяет построить 14 различных типов диаграмм.

1.4.6. СОЗДАНИЕ И СОХРАНЕНИЕ ДОКУМЕНТА

При создании нового документа Word всегда опирается на *шаблон*. Шаблон называют специальный вид документа, представляющий основные средства форматирования создаваемого документа. Шаблон можно считать своего рода пустым документом с заданными свойствами, который «накладывается» на создаваемый документ или на основе которого строится новый документ. Шаблоны используются для упрощения подготовки документов определенного типа (счета, заказы, планы, деловые письма, факсы и т.д.). Шаблоны позволяют придать документам единый формат и вид.

Шаблон как понятие включает в себя ряд элементов:

- текст или форматирование, которые одинаковы в каждом документе определенного типа;
- стили;
- панели инструментов.

Шаблоны бывают следующих типов:

- Normal, содержащий установки по умолчанию для стандартных документов. Кнопка *Создать* на панели инструментов *Стандартная* создает новый документ на основе шаблона Normal;

- шаблоны, поставляемые с Word или создаваемые пользователем. На основе созданного шаблона можно подготовить новый документ. Выполните команду *Файл/Создать* и на вкладке *Общие* выберите нужный шаблон. Откроется новое окно документа, причем имя этого документа будет — **Документ**, а не имя шаблона (так как файл шаблона остается без изменения). После загрузки шаблона его можно изменять, дополнять и сохранять полученный документ как обычный документ Word.

При работе с любым документом на основе шаблона все содержание и средства переносятся в новый документ, а сам шаблон, который хранится как файл, остается без изменения. Word предлагает несколько шаблонов деловых документов, (в том числе Изысканные, Стандартные, Современные — письма, служебные записки, факсы, заявления, резюме и т.д.). Если ни один из шаблонов не устраивает, вы можете создавать новый шаблон.

Сохранение является завершающей стадией основной работы по подготовке текстового документа. Пока документ не сохранен, все изменения находятся только в оперативной памяти компьютера. Все документы сохраняются как файлы определенного формата. Для сохранения нового документа или повторного сохранения документа с тем же именем используется команда *Файл/Сохранить*. При сохранении версий одного документа используется «инкрементный» метод,

когда в файл записываются только отличия от базовой версии. Для сохранения документа с новым именем или в другом месте (на новом диске, в новой папке) используется команда *Файл/Сохранить как*, которая дает возможность изменить параметры хранения.

Контрольные вопросы

1. Что такое текстовый процессор?
2. Какие функции выполняют развитые текстовые редакторы?
3. Перечислите и охарактеризуйте типы текстовых редакторов.
4. Назовите и охарактеризуйте режимы просмотра окон в редакторе Word.
5. Что такое абзац?
6. Для чего нужна операция откатки?
7. Для каких целей производится выделение фрагментов текста?
8. Что означает форматирование текста?
9. Что входит в понятие «форматирование абзаца»?
10. Что означает стиль абзаца?
11. Можно ли создать свой стиль абзаца?
12. Какие виды списков вы знаете?
13. Как изменяются уровни в многоуровневых списках?
14. Расскажите о возможностях работы с таблицами в редакторе Word.
15. Опишите структуру страницы.
16. Что значит форматирование страниц?
17. Для чего и как разделяют текст на разделы?
18. Как создать оглавление в редакторе Word?
19. Перечислите виды объектов, которые можно импортировать в документ Word.
20. Сформулируйте правила сохранения и открытия документов в Word.

Литература

1. Берлинер Э.М. Word 97. — М.: АБФ, 1997.
2. Евдокимов В.В. и др. Экономическая информатика: Учебник для вузов. — СПб.: Питер, 1997. — С.311–351.
3. Информатика: Базовый курс / С.В.Симонович и др. — СПб.: Питер, 2001. — С.249–295.
4. Информатика: Сборник лабораторных работ в текстовом процессоре Word. Ч.1, 2. — Новосибирск: НГАЭиУ, 2000.
5. Информатика: Учебник / Под ред. проф. Н.В.Макаровой. — М.: Финансы и статистика, 1998. — С.507–527.
6. Практикум по экономической информатике: Учебное пособие. Ч.1 / Под ред. Е.Л.Шурмова, Н.А.Тимаковой, Е.А.Мамонтовой. — М.: Перспектива, 2000. — С.55–124.
7. Фигурнов В.Э. IBM PC для пользователя. — М.: ИНФРА-М, 1995. — С.271–281.
8. Экономическая информатика / Под ред. П.В.Конюховского и Д.Н.Колесова. — СПб.: Питер, 2000. — С.149–209.

ТЕМА 1.5. ТАБЛИЧНЫЙ ПРОЦЕССОР MS EXCEL

1.5.1. НАЗНАЧЕНИЕ ЭЛЕКТРОННЫХ ТАБЛИЦ

В процессе решения расчетных задач часто требуется представлять данные в виде таблиц. Документы табличного типа (сводки, ведомости и т.п.) являются одними из основных информационных единиц, данные из которых постоянно используются, обновляются, пополняются и обрабатываются.

Именно для проведения расчетов на компьютере для данных, представленных в табличной форме, были созданы специальные пакеты прикладных задач, получившие название *табличные процессоры*. В последнее время табличный процессор стал обязательным элементом автоматизации учрежденческой и управленческой деятельности.

Электронные таблицы — это область экрана дисплея с сеткой, которая делит ее на столбцы и строки. Программные средства для проектирования электронных таблиц называют также табличными процессорами.

Табличный процессор (синоним — *электронная таблица*) — это пакет прикладных программ, обеспечивающий автоматизированную обработку информации, представленной в табличной форме.

С помощью электронных таблиц (ЭТ) можно выполнять различные экономические, бухгалтерские и инженерные расчеты, строить разного рода диаграммы, оптимизировать решение различных ситуаций. Табличный процессор является очень эффективным средством проведения численного моделирования ситуации или объекта. Меняя во всевозможных сочетаниях значения исходных параметров, можно наблюдать за изменением расчетных параметров и анализировать получаемые результаты. Таким образом, в считанные минуты получают множество вариантов решения поставленной задачи, на основании анализа которых выбирают наиболее приемлемое.

Табличные процессоры различаются, в основном, набором выполняемых функций и удобством интерфейса.

Наиболее популярными табличными процессорами для ПК в свое время являлись SuperCalc, Quattro Pro (фирмы WordPerfect), Lotus-1-2-3 (фирмы Lotus). В России в настоящее время лидирующие позиции занимает табличный процессор Excel фирмы Microsoft, входящий в пакет программ Microsoft Office.

Возможности табличных процессоров:

- формирование таблиц любого вида, содержащих информацию разного типа: текстовую, числовую, формулы, по которым осуществляется расчет. При этом формулы могут быть взаимосвязанными, т.е. результат вычисления в одной клетке таблицы может зависеть от результата, получаемого в другой клетке;
- для автоматического пересчета всей таблицы достаточно поменять только значения исходных данных, а расчет происходит каждый раз автоматически;
- оформление таблицы в удобном для пользователя виде;
- печать таблиц в виде, удобном для непосредственного использования в делопроизводстве.
- создание многотабличных документов, объединенных формулами;
- представление информации в графическом виде, т.е. автоматическое построение диаграмм, их модификация;
- сортировка таблиц;
- выборка данных, удовлетворяющих некоторым критериям, по запросу пользователя;
- создание итоговых и сводных таблиц;
- статистическая обработка информации;
- решение оптимизационных задач;
- хранение ЭТ на дисках для многократного использования;
- разработка макрокоманд, настройка среды под потребности пользователя и т.д.

1.5.2. ОБРАБОТКА ДАННЫХ В EXCEL

Электронная таблица состоит из столбцов и строк.

Заголовки столбцов — содержат буквы латинского алфавита, обозначающие столбцы ЭТ (A, B, C, ..., AA, AB, ..., IV), всего 256 столбцов.

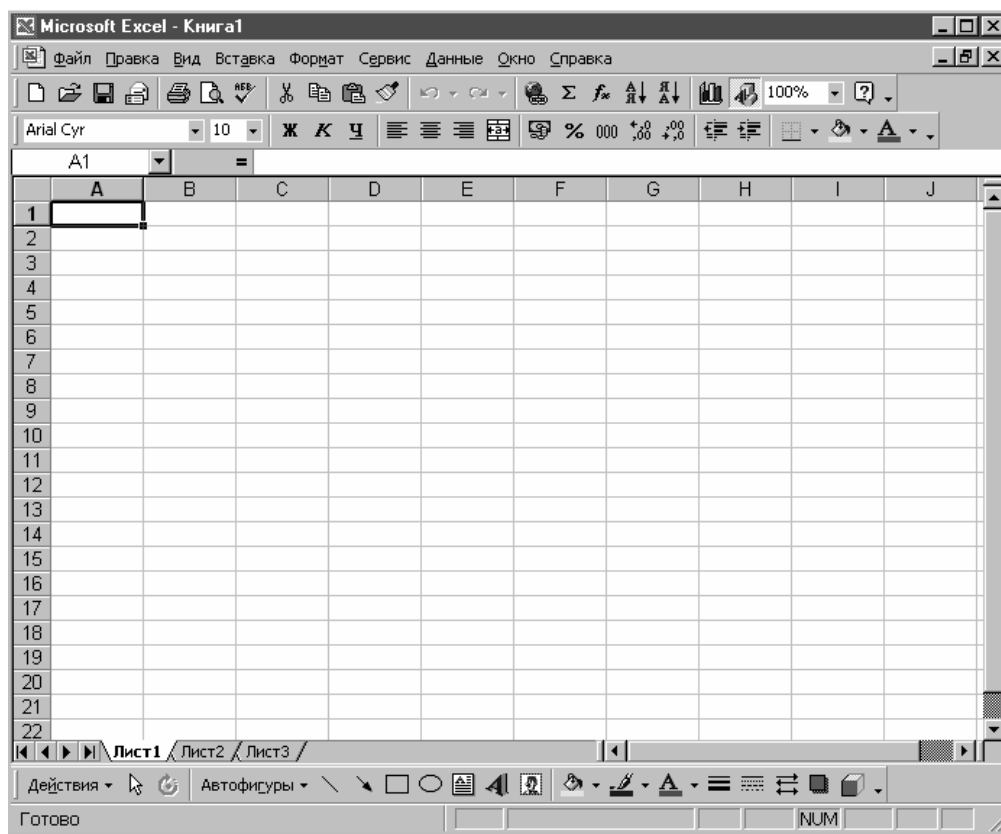
Заголовки строк — расположены в первом столбце и идентифицируют строки арабскими цифрами.

Ячейка — место пересечения столбца и строки. Каждая ячейка имеет свой адрес, состоящий из имени столбца и строки, например, A1 (но не 1A), E89, AC23. Выделенную ячейку называют *активной* или *текущей*. В ЭТ можно работать как с отдельной ячейкой, так и с группами ячеек, которые образуют блок.

Блоки ячеек — это прямоугольная группа смежных ячеек, имена ячеек в блоках разделяются двоеточием. Поэтому адрес блока — это адреса любых противоположных угловых ячеек блока, разделенных двоеточием, например, A1:A6, A1:C8, A1:E1.

Рабочий лист служит для ввода и анализа данных. Имена листов выводятся на ярлычках в нижней части экрана, щелкнув соответствующий ярлычок, можно перейти к нужному листу. С помощью кнопок-стрелок в левом нижнем углу экрана рядом с именами ярлычков можно прокручивать ярлычки горизонтально, чтобы найти нужный лист.

Рабочая книга — основное рабочее пространство ЭТ. Рабочая книга состоит из нескольких листов. По умолчанию книга открывается с рабочими листами — Лист1, Лист2 и т.д., число которых



можно увеличить или уменьшить (командой *Сервис/Параметры/вкладка Общие*, поле *Листов в новой книге*).

Любому листу можно присвоить другое название, для этого нужно дважды щелкнуть по ярлычку листа и ввести новое имя.

Рабочие листы можно вставлять (команда *Вставка/Лист*), удалять (команда *Правка/Удалить лист*), переименовывать (команда *Формат/Лист/Переименовать*), перемещать и копировать (команда *Правка/Переместить/Скопировать лист...*). Эти же команды можно выполнять, используя контекстное меню листов книги.

1.5.2.1. Создание, сохранение и чтение рабочих книг

При запуске Excel автоматически создается новая рабочая книга — Книга 1.

Создать новую рабочую книгу в процессе работы можно командой *Файл/Создать* или кнопкой на панели инструментов *Создать книгу*.

Чтобы рабочую книгу загрузить из внешней памяти (МД, CD-диска и т.п.), нужно воспользоваться командой *Файл/Открыть*.

Сохранить новую рабочую книгу первый раз или сохранить ее с тем же именем, с которым и загрузили (со всеми рабочими листами), можно с помощью команды *Файл/Сохранить*.

Сохранить рабочую книгу под новым именем нужно командой *Файл/Сохранить как*.

По умолчанию Excel автоматически устанавливает стандартное расширение имени файла — **.xls**.

1.5.2.2. Ввод и редактирование данных

После набора информации на клавиатуре записать ее в текущую ячейку и перейти к следующей ячейке можно тремя способами:

- нажатием клавиши <Enter> или <Tab>;
- нажатием клавиши управления курсором;
- щелчком левой кнопки мыши по другой ячейке.

В процессе ввода данных MS Excel автоматически распознает, что вводится — числа, текст или формулы.

В ячейки можно вводить 2 типа данных: константы и формулы.

Константы можно разделить на три основные категории:

- числовые значения;
- текстовые значения;
- значения дат и времени;

и специальные типы констант:

- логические значения;
- ошибочные значения.

Значения, которые хранятся в ячейках и появляются в строке формул, называются *храняемыми значениями*.

Значения, которые появляются в ячейках, называются *выводимыми* или *отображаемыми значениями*.

Числа. При вводе чисел нужно иметь в виду, что в десятичных числах дробная часть отделяется от целой запятой.

Если ширины столбца недостаточно для вывода числа, Excel может вывести либо округленное значение, либо строку символов # (это значит, что нужно увеличить ширину соответствующего столбца).

Числовой формат определяет внешнее представление числа в ячейке. Изменить числовой формат можно командой *Формат/Ячейки* (вкладка *Число*).

Текст. Текст может содержать практически любые символы.

При вводе длинного текста он будет показан на соседних клетках, если они пустые, и будет обрезан границей клетки, если она не пустая.

Чтобы показать весь длинный текст в ячейке, можно расширить столбец или вывести этот текст в несколько строк в одной ячейке, используя команду *Формат/Ячейки/Выравнивание/флажок Переносить по словам*.

Чтобы ввести как текст числовое выражение, нужно использовать апостроф ('). Например, номер телефона '24-24-00.

Даты. При вводе дат число, месяц и год отделяются знаком слэш (/), или дефис (-), или точкой (.). Например, 1.1.99 или 1/1/99 или 1-1-99.

При необходимости формат представления даты можно изменить командой *Формат/Ячейки/Число*.

Независимо от формата, используемого для представления дата, в Excel все даты сохраняются в памяти как последовательные числа. Система отсчета дат в Excel — 1.01.1900. Это первый день, т.е. запоминается как число 1. Благодаря этому даты можно складывать, вычитать и др. Например, записав формулу =5.1.2004-1.1.2004 получим результат 4.

Формулы. Все формулы в Excel должны начинаться со знака =. При вводе формулы в ячейке электронной таблицы отображается значение, вычисленное по этой формуле, а в строке формул — сама формула. Приоритет арифметических операций [(+) — сложение, (-) — вычитание, (*) — умножение, (/) — деление, (^) — возведение в степень] — обычный, для изменения приоритета операций используют круглые скобки.

В формулах можно использовать адреса ячеек или блоков (диапазонов) ячеек, константы и функции.

При вводе формул рекомендуется адреса ячеек не вводить с клавиатуры, а выделять соответствующие ячейки в электронной таблице, чтобы избежать ошибок некорректного ввода адресов (например, на русском языке).

Адреса ячеек в формулах могут быть относительные, абсолютные или смешанные.

Относительные ссылки (адреса) будут изменяться при копировании формул в другие ячейки.

Абсолютные ссылки (адреса) при копировании формул в другие ячейки остаются неизменными. Для того чтобы адрес сделать абсолютным, нужно записать его, используя символ \$. Например, \$B\$3.

Смешанные ссылки (адреса) могут иметь часть адреса неизменной. Например, \$B3 или B\$3.

Смешанные ссылки (адреса) могут иметь часть адреса неизменной. Например, \$B3 (не меняется столбец) или B\$3 (не меняется номер строки).

Примечание. Чтобы изменить тип ссылки (адреса) в формуле, можно установить курсор на ссылку или рядом и нажимать клавишу F4. Каждый раз при нажатии этой клавиши тип ссылки будет циклически изменяться.

Пример 1. Рассчитать сумму дохода семьи за три месяца, если известен доход за каждый из этих месяцев, и определить, какую долю из общего дохода составляет доход каждого месяца.

Чтобы рассчитать долю дохода за январь, введем в ячейку D4 формулу =C4/C7. Если эту формулу скопируем в ячейки D5:D6, то получим следующий результат:

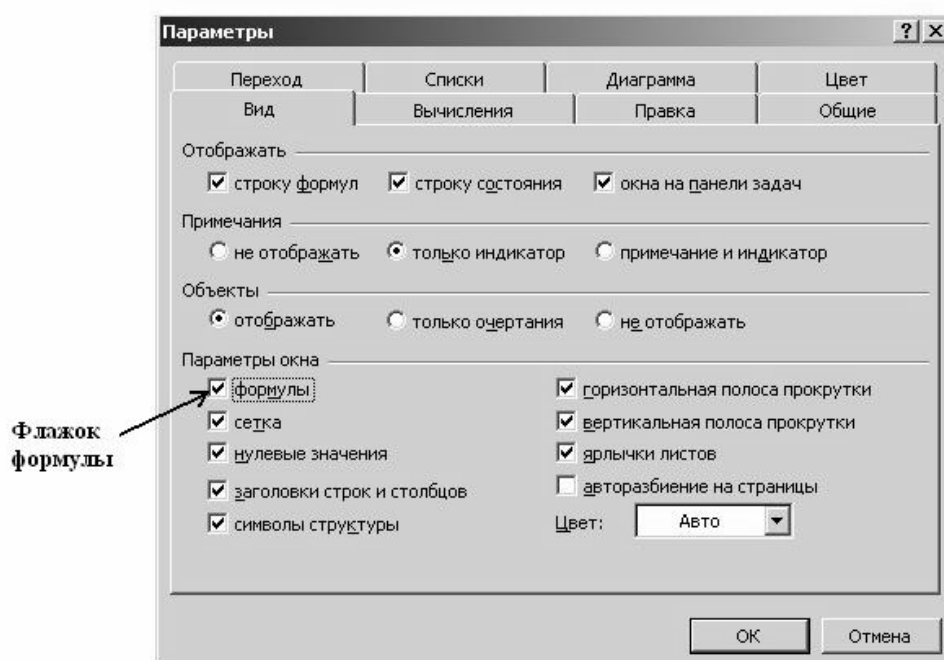
	А	В	С	Д	Е
1					
2					
3		Месяц	Доход	Доля за квартал	
4		январь	12000	0,28	
5		февраль	15400	#ДЕЛЮ!	
6		март	16010	#ДЕЛЮ!	
7		Итого за квартал	43410		
8					

Проанализируем формулы, полученные после копирования:

	А	В	С	Д	Е
1					
2					
3		Месяц	Доход	Доля за квартал	
4		январь	12000	=C4/C7	
5		февраль	15400	=C5/C8	
6		март	16010	=C6/C9	
7		Итого за квартал	=СУММ(C4:C6)		
8					
9					

В данном примере необходимо, чтобы адрес в знаменателе не менялся. Для того чтобы знаменатель при копировании формулы не менялся, используем при записи формулы смешанный адрес C\$7 (так как копирование формулы производится вниз — на новые строки, то достаточно зафиксировать в адресе номер строки).

Примечание. Чтобы показать на листе электронной таблицы все формулы, нужно в окне диалога команды *Сервис/Параметры* на вкладке *Вид* установить флажок формулы:



Теперь формулы после копирования примут вид:

	А	В	С	Д	Е
1					
2					
3		Месяц	Доход	Доля за квартал	
4		январь	12000	=С4/С\$7	
5		февраль	15400	=С5/С\$7	
6		март	16010	=С6/С\$7	
7		Итого за квартал	=СУММ(С4:С6)		
8					

Результат вычислений:

	А	В	С	Д	Е
1					
2					
3		Месяц	Доход	Доля за квартал	
4		январь	12000	0,28	
5		февраль	15400	0,35	
6		март	16010	0,37	
7		Итого за квартал	43410		
8					

В формулах вместо адресов могут быть использованы имена ячеек или блоков ячеек. Имена ячеек делают формулы более удобными для понимания, а при копировании формул заменяют абсолютные адреса. Присвоить имя блоку ячеек можно с помощью команды *Вставка/Имя/Присвоить*.

Для нашего примера выполним еще один вариант: введем формулу с использованием имени блока ячеек. Выделим ячейку С7 (одна ячейка — частный случай блока ячеек) и присвоим ей имя *Итого_за_квартал*. При вводе формулы в ячейку D4 после выделения ячейки С7 ее адрес автоматически заменится на имя.

	А	В	С	Д
1				
2				
3		Месяц	Доход	Доля за квартал
4		январь	12000	=С4/Итого_за_квартал
5		февраль	15400	=С5/Итого_за_квартал
6		март	16010	=С6/Итого_за_квартал
7		Итого за квартал	=СУММ(С4:С6)	
8				

Для ввода в ячейки последовательности данных:

- можно воспользоваться инструментом *Автозаполнения*, протаскивая мышью маркер заполнения, находящийся на рамке ячейки;
- с помощью команды *Правка/Заполнить/Прогрессия* можно быстро создать ряд чисел или дат.

Редактировать данные можно двумя способами:

- выделить ячейку и редактировать данные в строке формул или прямо в ячейке;
- дважды щелкнув на ячейке, а затем поместить текстовый курсор на место, куда нужно внести изменения.

Основными текущими операциями редактирования являются удаление, вставка, перемещение, копирование ячеек, столбцов и строк.

Для удаления ячеек, столбцов или строк выполняют команду *Правка/Удаление*.

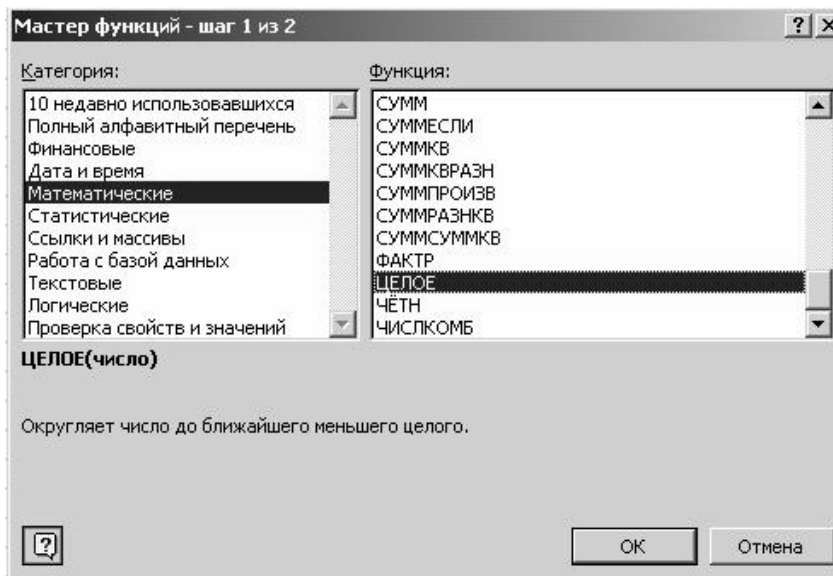
Для вставки блока ячеек, столбца или строки их предварительно выделяют, тем самым задавая исходный объект, по отношению к которому определяется направление вставки. Далее выполняется команда *Вставка*, в которой выбирается один из вариантов вставки.

При копировании ячеек они выделяются, затем сначала выполняется команда *Правка/Копировать*, затем указывается ячейка, которая станет верхним левым углом скопированного диапазона, и выполняется команда *Правка/Вставить*.

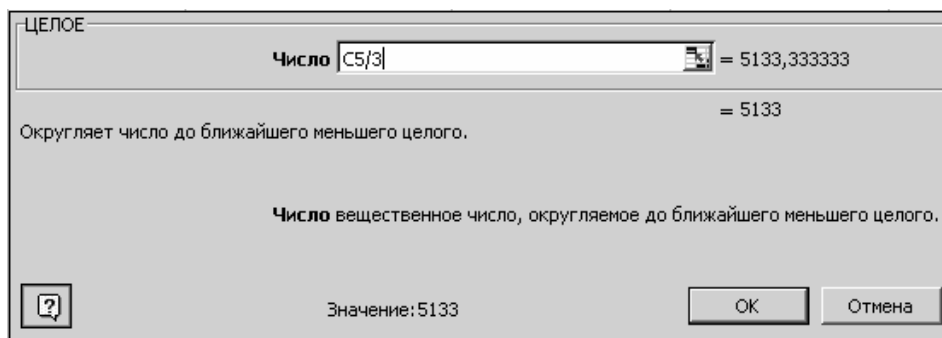
1.5.2.3. Функции

Очень часто в вычислениях при записи формул используют функции, которые оперируют с одним или несколькими значениями — аргументами. Аргументы при записи функций отделяются точкой с запятой. Excel содержит более 400 так называемых *встроенных функций*. Каждая функция имеет имя и аргументы в круглых скобках. Самый простой и удобный способ использования встроенных функций в Excel — использование *Мастера функций*. В *Мастере функций* все функции разделены на 10 категорий, например, математические, статистические, логические и т.д. *Мастер функций* можно вызвать, нажав кнопку на панели инструментов, или выполнив команду *Вставка/Функция*.

В появившемся диалоговом окне выбирают категорию функции и в списке справа нужную функцию из этой категории.

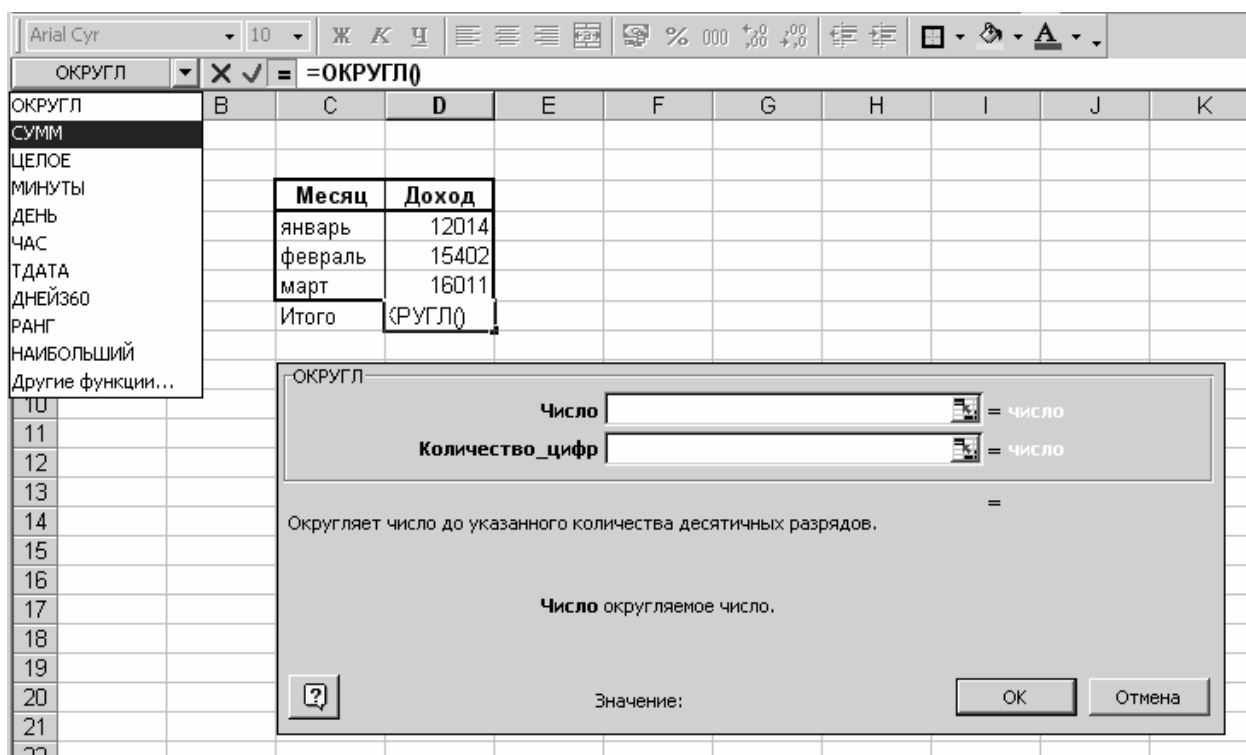


Следующее окно содержит поля для каждого из аргументов выбранной функции. Здесь же выводится описание для каждого текущего аргумента. Как в любую формулу, в функцию можно вводить ссылки на ячейки, причем для этого достаточно выделить эти ячейки в электронной таблице. Справа от каждого поля аргумента отображается текущее значение аргумента.



Примечание. Если в качестве аргумента функции используется также функция, то внутреннюю (вложенную) функцию следует выбирать с помощью списка функций, который находится слева от поля ввода. После ввода аргументов внутренней функции не щелкайте кнопку **Ок** (Готово), так как это досрочно завершит ввод формулы. Вместо этого установите курсор в поле ввода на имени внешней функции. Тогда окно *Мастера функций* внутренней функции заменится на окно *Мастера функций* внешней функции.

Например, в ячейку D7 требуется ввести формулу =ОКРУГЛ(СУММ(D4:D6);-1).



Рассмотрим одну из самых часто используемых функций — математическую функцию СУММ. Эта функция может иметь до 30 аргументов, причем каждый аргумент может быть числом, адресом ячейки или блока ячеек, содержащих число или формулу, возвращающую числовое значение. Например, функция СУММ(A2; B2:K2; 500) имеет три аргумента (первый аргумент — адрес ячейки A2, второй — адрес блока ячеек B2:K2, третий — числовая константа 500). Функция СУММ игнорирует аргументы, которые ссылаются на пустые ячейки, текстовые или логические значения. Так как СУММ является очень часто используемой функцией, то на панели инструментов для ввода этой функции есть специальная кнопка — **Автосуммирование** (Σ).

Хотелось бы обратить внимание на несколько математических функций:

ОКРУГЛ — округляет число до указанного количества десятичных знаков;

ОТБР — отбрасывает дробную часть числа;

ЦЕЛОЕ — округляет число до ближайшего меньшего целого.

Х	ОКРУГЛ(Х;0)	ОТБР(Х;0)	ЦЕЛОЕ(Х)
-138,59	-139	-138	-139
-138,4	-138	-138	-139
138,59	139	138	138
138,4	138	138	138

Примечание. Для того чтобы правильно использовать функцию, необходимо внимательно прочитать справку по назначению и использованию этой функции.

В табл. 1 и 2 приведены некоторые часто используемые функции двух категорий — математические и статистические функции.

Математические функции

Синтаксис функции	Назначение функции	Пример	
		Запись формулы	Результат
ABS (число)	Возвращает модуль (абсолютную величину) числа или формулы	=ABS(-5,2) =ABS(5-D3)	5,2
EXP (число)	Возвращает экспоненту заданного числа, т.е. вычисляет результат возведения числа $e=2,71828$ в степень, равную аргументу – число	=EXP(2) =EXP(C2)	7,389056
LN(число)	Возвращает натуральный логарифм числа	=LN(2,5) =LN(-2) =LN(0)	0,916291 Ошибка:#ЧИСЛО Ошибка:#ЧИСЛО
LOG10(число)	Возвращает десятичный логарифм числа	=LOG10(100)	2
КОРЕНЬ	Возвращает значение квадратного корня	=КОРЕНЬ(25) =КОРЕНЬ(54,2) =КОРЕНЬ(-4)	5 7,362065 Ошибка:#ЧИСЛО
ОКРУГЛ(число; число_цифр)	Округляет число, задаваемое ее аргументом, до указанного количества десятичных знаков	=ОКРУГЛ(123,45;-2) =ОКРУГЛ(123,45;0) =ОКРУГЛ(123,45;2)	100 123 123,46
ОСТАТ(число; делитель)	Возвращает остаток от деления числа на делитель	=ОСТАТ(19;5) =ОСТАТ(6;10) =ОСТАТ(19,6;5,1)	4 6 4,3
ОТБР(число; число_цифр)	Отбрасывает дробную часть числа. Число_цифр – число, определяющее точность усечения. По умолчанию берется значение 0	=ОТБР(13,978) =ОТБР(-13,978) =ОТБР(193,458;2) =ОТБР(193,458;-2)	13 -13 193,45 100
ПРОИЗВЕД(число1; число2;...)	Возвращает произведение аргументов	=ПРОИЗВЕД(0;D3;456,1)	0
СТЕПЕНЬ(число; степень)	Возвращает результат возведения числа в степень	=СТЕПЕНЬ(3;2) =СТЕПЕНЬ(25;0,5) =СТЕПЕНЬ(25;-0,5)	9 5 0,2
СУММ(число1; число2;...)	Возвращает сумму аргументов	=СУММ(6;10;50,5) =СУММ(A1:A5;C1:C5)	66,5
СУММПРОИЗВ(массив1; массив2;...)	Возвращает сумму произведений соответствующих элементов массивов (массив – блок клеток или массив чисел)	=СУММПРОИЗВ(C2:C4; D2:D4) =СУММПРОИЗВ({2;3;4}; {5;10;5})	Сумма произведений: C2*D2+C3*D3+C4*D4 60
ЦЕЛОЕ(число)	Округляет число до ближайшего меньшего целого	=ЦЕЛОЕ(20,65) =ЦЕЛОЕ(-20,65)	20 -21

Таблица 2

Статистические функции

Синтаксис функции	Назначение функции	Пример	
		Запись формулы	Результат
МАКС(число1; число2;...)	Возвращает максимальное число из списка аргументов. Логические значения или текст игнорируются	=МАКС(5,9;8;4,1) =МАКС(C3:D6;A2;E2)	8
МИН(число1; число2;...)	Возвращает минимальное число из списка аргументов. Логические значения или текст игнорируются	=МИН(9;-15;0) =МИН(D4:F7; G7)	-15
НАИБОЛЬШИЙ (массив;k)	Возвращает k-е наибольшее значение из множества данных	=НАИБОЛЬШИЙ (D2:D13;2) =НАИБОЛЬШИЙ ({5;7;2;6};3)	5
РАНГ(число; ссылка; порядок)	Возвращает для числа его ранг – порядковый номер относительно других чисел в списке. Если порядок равен 0 или опущен, то как бы отсортирован в порядке убывания, если любое ненулевое число, то – в порядке возрастания	=РАНГ(D2;D2:D8)	
СРЗНАЧ(число1; число2;...)	Возвращает среднее (арифметическое) значение аргументов	=СРЗНАЧ(4;5;3) =СРЗНАЧ(4;5;-3) =СРЗНАЧ(A1;A4:A6)	4 2
СЧЕТ(значение1; значение2;...)	Подсчитывает количество чисел в списке аргументов	=СЧЕТ(A3:A9; C3:C8)	

Пример 2. Бригаде начислен фонд заработной платы, который распределяется между членами бригады в соответствии с индивидуальными коэффициентами, определяемыми с учетом ответственности и качества работы. Определить зарплату, начисленную каждому члену бригады.

Приведем два варианта расчета зарплат.

Вариант 1. Зарплата, начисленная одному члену бригады, вычисляется как общий фонд зарплат бригады, деленный на сумму индивидуальных коэффициентов и умноженный на индивидуальный коэффициент этого работника. Чтобы при копировании формулы из ячейки С5 в ячейки С6:С9 адрес ячейки В2 (фонд зарплат) и адрес блока ячеек В5:В9 (сумма индивидуальных коэффициентов) не менялись, достаточно в этих адресах зафиксировать номер строки.

	А	В	С	Д
1	Начисление зарплат			
2	Фонд з/плат	20000		
3				
4	Ф.И.О.	Индивидуальный коэффициент	Начислено	
5	Абдулов С.А.	1,2	=В\$2/СУММ(В\$5:В\$9)*В5	
6	Борцов П.И.	1,1	=В\$2/СУММ(В\$5:В\$9)*В6	
7	Игнатов А.А.	1,5	=В\$2/СУММ(В\$5:В\$9)*В7	
8	Киселев К.В.	1,3	=В\$2/СУММ(В\$5:В\$9)*В8	
9	Сазонов В.Н.	1,4	=В\$2/СУММ(В\$5:В\$9)*В9	
10	Всего		=СУММ(С5:С9)	
11				

Примечание. В данной задаче можно использовать абсолютные адреса \$В\$2 и СУММ(\$В\$5:\$В\$9), результат от этого не изменится.

Результат вычисления формул:

	А	В	С	Д
1	Начисление зарплат			
2	Фонд з/плат	20000		
3				
4	Ф.И.О.	Индивидуальный коэффициент	Начислено	
5	Абдулов С.А.	1,2	3692	
6	Борцов П.И.	1,1	3385	
7	Игнатов А.А.	1,5	4615	
8	Киселев К.В.	1,3	4000	
9	Сазонов В.Н.	1,4	4308	
10	Всего		20000	
11				

Вариант 2. Добавим в условие задачи требование округлить суммы в графе «Начислено» до двух знаков после запятой. В этом варианте в ячейке В10 подсчитаем сумму индивидуальных коэффициентов и присвоим этой ячейке имя *ВсегоИндКозф*. Напоминаем, что имя блока ячеек заменяет собой абсолютную адресацию, поэтому формулы приобретут следующий вид:

	А	В	С	Д
1	Начисление зарплат			
2	Фонд з/плат	20000		
3				
4	Ф.И.О.	Индивидуальный коэффициент	Начислено	
5	Абдулов С.А.	1,2	=ОКРУГЛ(В\$2/ВсегоИндКозф*В5;2)	
6	Борцов П.И.	1,1	=ОКРУГЛ(В\$2/ВсегоИндКозф*В6;2)	
7	Игнатов А.А.	1,5	=ОКРУГЛ(В\$2/ВсегоИндКозф*В7;2)	
8	Киселев К.В.	1,3	=ОКРУГЛ(В\$2/ВсегоИндКозф*В8;2)	
9	Сазонов В.Н.	1,4	=ОКРУГЛ(В\$2/ВсегоИндКозф*В9;2)	
10	Всего	=СУММ(В5:В9)	=ОКРУГЛ(СУММ(С5:С9);2)	
11				

Пример 3. Известна зарплата, начисленная каждому члену бригады за второй квартал, т.е. за апрель, май и июнь. Определить общую сумму, начисленную каждому члену бригады за квартал, и среднюю зарплату этого сотрудника за второй квартал. Кроме того, требуется подсчитать среднемесячные зарплаты в бригаде, а также фонд зарплаты бригады за каждый месяц и за квартал в целом.

	A	B	C	D	E	F
1	Начисление зарплаты					
2						
3	Ф.И.О.	апрель	май	июнь	Итого	Средняя з/плата
4	Абдулов С.А.	3690	3950	3860	=СУММ(B4:D4)	=СРЗНАЧ(B4:D4)
5	Борцов П.И.	3385	3155	3500	=СУММ(B5:D5)	=СРЗНАЧ(B5:D5)
6	Игнатов А.А.	4615	4650	4765	=СУММ(B6:D6)	=СРЗНАЧ(B6:D6)
7	Киселев К.В.	4000	4465	4750	=СУММ(B7:D7)	=СРЗНАЧ(B7:D7)
8	Сазонов В.Н.	4310	3980	4125	=СУММ(B8:D8)	=СРЗНАЧ(B8:D8)
9	Всего	=СУММ(B4:B8)	=СУММ(C4:C8)	=СУММ(D4:D8)	=СУММ(E4:E8)	
10	Среднемесячная з/плата	=СРЗНАЧ(B4:B8)	=СРЗНАЧ(C4:C8)	=СРЗНАЧ(D4:D8)		
11						

1.5.2.4. Логические функции

В процессе обработки данных нередко ситуации, когда в зависимости от каких-либо условий следует выполнять либо одну, либо другую операцию.

Условия могут быть простые или сложные. Для записи условий используются соответственно простые или сложные логические выражения.

Простое условие (простое логическое выражение) представляет собой отношение вида $A*B$, где * — одна из операций отношения (=, <, >, >=, <=, <>);

A и B — сравниваемые значения, которые могут быть числами, формулами, текстовыми или логическими значениями.

Результатом логического выражения является логическое значение «истина» (1) или «ложь» (0). Например, результатом логического выражения $15>7$ является «истина», а результат логического выражения $F5>7$ зависит от содержимого ячейки F5.

Для построения сложных логических выражений в Excel имеется набор логических функций.

Сложное условие (сложное логическое выражение) представляет собой два или несколько простых условий, являющихся аргументами логических функций И, ИЛИ, НЕ. Аргументами функций И, ИЛИ и НЕ могут быть логические выражения либо ссылки на ячейки, содержащие логические выражения. Функции И и ИЛИ могут иметь до 30 логических аргументов.

Функция И. Эта функция имеет следующий синтаксис:

=И(логическое_выражение1; логическое_выражение2;...)

Функция И возвращает логическое значение ИСТИНА, если только все аргументы (логические значения) имеют значение ИСТИНА, т.е. если все логические выражения — истинны (т.е. если все условия выполняются).

Пример 1. Условие «сданы ли все предметы только на 5» (оценки записаны в ячейках B4;C4;D4), можно записать:

И(B4=5; C4=5; D4=5)

Если в ячейку E4 записать формулу с использованием этой логической функции, то в ячейке E4 получим результат ИСТИНА или ЛОЖЬ:

	A	B	C	D	E	F
1						
2						
3						
4		5	4	5	ЛОЖЬ	
5						

формула

Пример 2. Мы имеем три ячейки A1, B1 и C1. Составить условие, которое позволит определить, является ли значение ячейки A1 больше остальных. Чтобы значение ячейки A1 было наибольшим, необходимо чтобы оно было больше значения ячейки B1 и больше значения ячейки C1. То есть нужно проверить условие:

=И(A1>B1; A1>C1)

Функция ИЛИ. Эта функция имеет следующий синтаксис:

=ИЛИ (логическое_выражение1; логическое_выражение2;...)

Функция ИЛИ возвращает логическое значение ИСТИНА, если хотя бы один аргумент (логическое значение) имеет значение ИСТИНА, т.е. если хотя бы одно логическое выражение — истинно (т.е. если хотя бы одно условие выполняется).

Пример 3. Условие «если хотя бы одна оценка ниже 4» (оценки записаны в ячейках B4;C4;D4) можно записать:

ИЛИ(B4<4; C4<4; D4<4)

Пример 4. Выяснить, есть ли среди ячеек A1, B1, C1 ячейки с отрицательным результатом. То есть нам требуется выяснить, есть ли хотя бы в одной ячейке отрицательное значение. Для этого запишем в ячейке E1 логическое выражение:

=ИЛИ(A1<0;B1<0;C1<0)

E1	=ИЛИ(A1<0;B1<0;C1<0)				
	A	B	C	D	E
1	521	-4,5	3		ИСТИНА
2					

Функция НЕ. Эта функция имеет следующий синтаксис:

=НЕ(логическое_значение)

Функция НЕ меняет значение своего аргумента на противоположное логическое значение и обычно используется в сочетании с другими функциями. Эта функция возвращает логическое значение ИСТИНА, если аргумент имеет значение ЛОЖЬ, и логическое значение ЛОЖЬ, если аргумент имеет значение ИСТИНА.

Пример 5. Логическое выражение =НЕ(7<10) имеет значение ЛОЖЬ.

Пример 6. Выражение =НЕ(F1>=13) имеет значение ИСТИНА, если F1<13:

B2	=НЕ(F1>=13)					
	A	B	C	D	E	F
1						8
2		ИСТИНА				
3						

Для записи результата после анализа значения логического выражения (т.е. после анализа выполнения условия) используется логическая функция ЕСЛИ.

Функция ЕСЛИ. Эта функция имеет следующий синтаксис:

ЕСЛИ(логическое_выражение; значение_если_истина; значение_если_ложь)

значение_если_истина — результат, который должен возвращаться функцией, если логическое выражение — истинно (условие выполняется).

значение_если_ложь — результат, который должен возвращаться функцией, если логическое выражение — ложно (условие не выполняется).

Пример 7. Записать результат вступительных экзаменов абитуриента. Если абитуриент набрал 13 баллов и больше (набранный балл введен в ячейку C2), то он зачислен в институт.

D2	=ЕСЛИ(C2>=13;"Зачислен";"Не зачислен")				
	A	B	C	D	E
1					
2			14	Зачислен	
3					

Пример 8. Записать в ячейку B7 формулу, которая анализирует значение суммы ячеек B1:B6 — если значение суммы положительное, то в ячейку B7 записывается значение этой суммы, иначе записывается 0:

	A	B	C	D	E	F
1		10				
2		-3				
3		6				
4		5				
5		-9				
6		2				
7		11				
8						

Пример 9. Студенту начисляется стипендия, если все экзамены в сессии сданы на 4 и 5. Записать в ячейку F3 информацию — начислена ли стипендия студенту, который сдал 3 экзамена; результаты экзаменов введены в ячейки C3, D3, E3. В ячейку F3 вводим формулу:

	A	B	C	D	E	F	G	H
1								
2	Ф.И.О.	Группа	История	В.матем.	Ин.яз.			
3	Седов В.А.	ЭМ-01	4	4	5	Начислена		
4								

Вложенная функция ЕСЛИ. Иногда после проверки одного какого-то условия требуется проверка дополнительных условий. В этом случае используют вложенные функции ЕСЛИ (можно использовать до 7 уровней вложения функции ЕСЛИ, но необходимо помнить, что максимальная длина записи в ячейке — 255 символов). Вложенная функция ЕСЛИ в качестве одного из аргументов — значение_если_истина или значение_если_ложь использует опять же функцию ЕСЛИ.

Примечание. Внутреннюю (вложенную) функцию ЕСЛИ следует выбирать, как любую вложенную функцию, в списке функций, который находится слева от строки формул.

Пример 10. Вычислить значение функции Y в зависимости от значения аргумента X:

$$Y = \begin{cases} 5 + X & \text{при } X < 0 \\ X - 10 & \text{при } X > 10 \\ 5 & \text{при остальных } X \end{cases}$$

Значения X будем вводить, например, в ячейку A2, а значение Y получим в ячейке B2. Тогда в B2 запишем формулу:

	A	B	C	D	E	F
1						
2	17	7				
3						

Пример 11. Предположим, что банковский процент зависит от величины вклада следующим образом: для вкладов размером до 10 тыс. руб. процент составляет 7%, для вкладов свыше 10 тыс. и до 30 тыс. — 8%, а свыше 30 тыс. — 9%. Записать выражение для вычисления размера вклада, записанного в ячейке A3, по истечении года хранения вклада. Формулу для вычисления размера вклада запишем в ячейку B3.

	A	B	C	D	E	F
1						
2	Размер вклада	Вклад с начислением годовых процентов				
3	21000	22680				
4						

Мы намеренно усложнили формулу, чтобы еще раз показать использование функции И, в данном примере можно ввести такую формулу:

=A3*ЕСЛИ(A3<=10000;107%;ЕСЛИ(A3>30000;109%;108%)).

1.5.3. ВИДЫ ЭКОНОМИЧЕСКОЙ ИНФОРМАЦИИ. СПРАВОЧНЫЕ ТАБЛИЦЫ

Относительно процесса обработки различают следующие виды экономической информации:

- *входная (или оперативная) информация* — данные, необходимые для решения конкретной задачи, причем частота их обновления определяется периодичностью решения задачи;
- *нормативно-справочная (или условно-постоянная) информация* — информация, которая остается неизменной в течение длительного периода времени и используется многократно для решения одной или нескольких задач. Изменение справочной информации осуществляется по мере необходимости;
- *выходная информация* представляет собой результат решения задачи.

Поэтому при решении экономических задач средствами табличных процессоров, как правило, создается несколько таблиц. Если задача простая, то допускается объединение входной и выходной информации в одну таблицу. Но для справочной информации создаются отдельные таблицы, данные из которых могут использоваться в разных таблицах и разных задачах. Любое изменение справочной информации должно оперативно отражаться во всех таблицах, где эта информация используется. В табличных процессорах существуют средства, позволяющие работать со справочной информацией.

Для поиска информации в справочнике он должен быть правильно организован.

- Справочник в ЭТ должен содержать не менее двух строк и столбцов.
- Каждая строка (для вертикально ориентированных справочников) или столбец (для горизонтально ориентированных справочников) в этой таблице называется *записью*. А каждая клетка в записи, содержащая определенную категорию информации, называется *полем*.
- При создании справочника необходимо определить *ключ*, т.е. такое данное, значение которого будет однозначно указывать на конкретную запись в справочнике.
- Ключевое поле обязательно должно быть первым столбцом (для вертикальных справочников) или первой строкой (для горизонтальных справочников).
- Значение ключей в справочнике должно быть *уникально*, т.е. в справочнике не должно быть двух записей с одним значением ключа.
- Записи могут сортироваться по нескольким полям. При этом сначала записи сортируются по первому полю, затем записи, имеющие одинаковое значение ключа по этому полю, сортируются по второму указанному полю и т.д.
- Записи справочника должны располагаться по возрастанию ключа. Поэтому при создании справочника его записи сортируют *по возрастанию*. (Это требование в последних версиях Excel не является обязательным.)

Как правило, для работы со справочной информацией используют встроенные функции ВПР или ГПР.

Функция ВПР, входящая в категорию функций *Ссылки и массивы*, используется для поиска информации в таблицах. При этом для определения позиции конкретной ячейки используются индексы строки и столбца. Синтаксис этой функции в общем виде:

ВПР(искомое значение; табл_массив; номер индекса столбца; диапазон просмотра),

где *искомое значение* — значение, которое должно быть найдено в первом столбце таблицы массива;

табл_массив — таблица, в которой ищутся данные;

номер индекса столбца — номер столбца в табл_массиве, в котором должно быть найдено соответствующее значение. Первый столбец имеет номер 1;

диапазон просмотра — логическое значение, определяющее точно или приблизительно должно производиться сопоставление.

Первый индекс (строки) определяется по результату поиска значения в первом столбце *табл_массива*, которое меньше или равно заданному аргументу *искомое значение*. В качестве второго индекса используется *номер индекса столбца*.

Функция ГПР аналогична функции ВПР, но используется для поиска информации в горизонтально ориентированных таблицах, т.е. искомое значение ищется в первой строке *табл_массива*.

Функция ВПР используется для поиска информации в таблицах-справочниках. Эта функция записывается в клетку выходного документа, куда следует поместить искомое данное из справочника.

Например, имеется таблица-справочник:

	А	В	С	Д
1				
2	Табельный №	Ф.И.О.	Разряд	Тарифная ставка
3	101	Алексеев П.И.	5	75,5
4	102	Васин С.В.	4	65,0
5	104	Колобов А.А.	5	75,5
6	105	Судаков И.К.	3	55,5
7	107	Старков Р.Л.	4	65,0

Для включения в выходную таблицу информации из справочника запишем в соответствующие ячейки формулы с функцией ВПР:

	А	В	С
21			
22	Табельный №	Ф.И.О.	Тарифная ставка
23	102	=ВПР (A23;\$A\$3:\$D\$7;2;0)	=ВПР (A23;\$A\$3:\$D\$7;4;0)
24	105	=ВПР (A24;\$A\$3:\$D\$7;2;0)	=ВПР (A24;\$A\$3:\$D\$7;4;0)

1.5.4. ГРАФИЧЕСКИЕ ВОЗМОЖНОСТИ EXCEL

Особое место в экономике занимают графические методы представления информации, которые помогают анализу данных. С помощью диаграмм легко наглядно представить закономерности, которые трудно бывает уловить в больших статистических таблицах и расчетах, увидеть тенденции развития какого-либо явления, взаимосвязь показателей.

Диаграмма состоит из графического образа и вспомогательных элементов.

Графический образ — это совокупность точек, линий и фигур, с помощью которых изображаются данные.

По характеру графического образа различают: графики, гистограммы, круговые, точечные диаграммы и т.д.

Существуют два варианта размещения диаграмм:

- *внедренные диаграммы* — хорошо выглядят в отчетах, когда надо, чтобы данные и диаграммы были представлены рядом;
- *диаграммный лист* — для диаграммы выделяется отдельный рабочий лист. Этой возможностью следует воспользоваться, если Вы хотите выполнить диаграмму в виде слайда. Такая диаграмма будет представлять данные, находящиеся на другом листе. Диаграммные листы будут обозначены: *Диаграмма 1*, *Диаграмма 2* и т.д.

В любом из этих вариантов диаграммы сохраняются вместе с книгой, в которой они находятся. Поэтому, чтобы сохранить диаграмму, достаточно сохранить книгу командой *Файл/Сохранить* или *Файл/Сохранить как*.

При открытии книги командой *Файл/Открыть* открываются и все диаграммы этой книги.

1.5.4.1. Основные понятия деловой графики

Каждая диаграмма может и должна иметь название.

В большинстве диаграмм данные размещаются между вертикальной линией (осью Y) и горизонтальной линией (осью X). Координатные оси: две оси представляют категории и значения. Как правило, ось категорий — горизонтальная ось, а ось значений — вертикальная, но для некоторых типов диаграмм (например, линейчатая диаграмма) может быть наоборот.

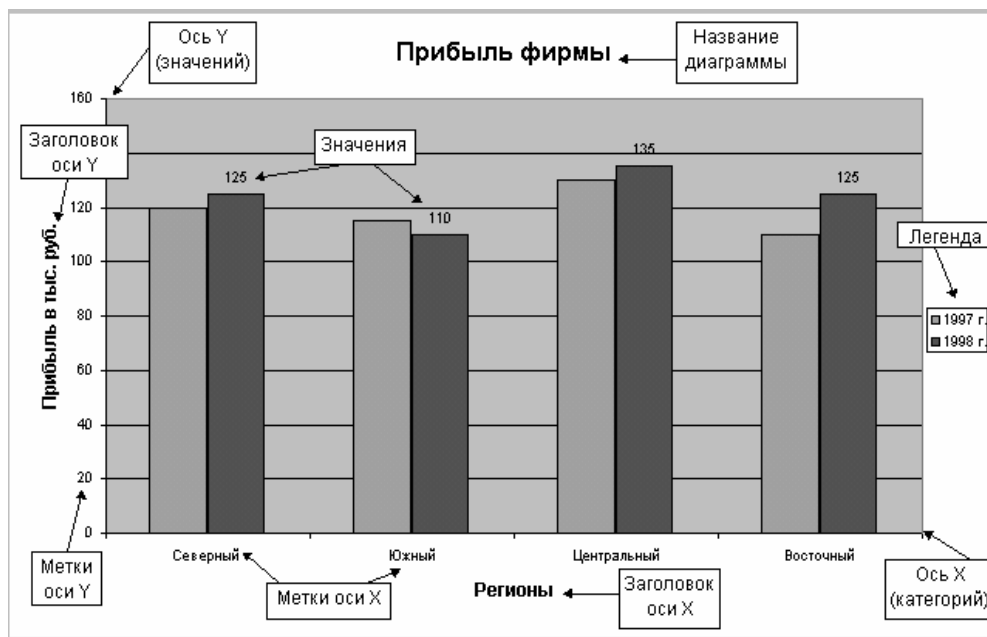


Рис.5. Основные элементы диаграммы

Ось категорий — ось X. Категории задают положение конкретных значений в ряде данных — это метки на оси X. Для некоторых типов диаграмм (например, точечной диаграммы) эта ось также является осью значений.

Числовая ось — ось Y, ось значений. Метки располагаются на осях координат через равные интервалы и помогают идентифицировать данные на диаграмме.

Названия осей — оси X и Y могут и должны иметь названия для лучшего понимания диаграмм.

Точка данных — отдельное значение, взятое из электронной таблицы и представленное на диаграмме.

Ряд данных — отдельная строка (столбец) таблицы, т.е. это то множество значений, которые вы хотите отобразить на диаграмме и которые соответствуют, например, какому-то показателю. Каждый ряд может иметь до 4000 значений или точек данных.

Например, отображаем прибыль фирмы по каждому из регионов за 2000 и 2001 годы. Категориями являются регионы. *1 ряд данных* — множество значений прибыли фирмы по всем регионам за 2000 г. *2 ряд данных* — множество значений прибыли фирмы по всем регионам за 2001 г.

Маркер данных — это отметка на диаграмме конкретного значения данных. Все значения одного ряда данных изображаются на диаграмме одинаковыми маркерами.

Легенда — прямоугольник, в котором указывается, каким цветом или типом линий отображаются на графике или диаграмме данные из того или иного ряда данных. Это необязательный параметр.

1.5.4.2. Типы диаграмм

В Excel 97 *Мастер диаграмм* позволяет строить диаграммы 14 базовых типов. Причем каждый тип имеет несколько подтипов — видов. Пользователь, выбирая определенный тип и вид диаграммы, может получить вариант, наилучшим образом отображающий данные. Поэтому пользователю нужно хорошо ориентироваться в том, какие типы диаграмм предоставляет Excel. Подробно рассмотрим все виды только для одного типа диаграмм — *График*, чтобы показать, какие возможности предлагает один тип диаграмм.

График. Тип *График* используется для отображения динамики изменений ряда значений. Графики наиболее подходят для иллюстраций изменения с течением времени одной или нескольких величин, поэтому их чаще всего используют для представления временных тенденций.

Этот тип диаграммы имеет 7 видов.

1-й вид — каждый ряд данных представляется на диаграмме отдельной ломаной линией. Легенда указывает, какой тип линии используется для каждого ряда данных.

2-й вид — график с накоплением, где также каждый ряд данных представляется на диаграмме отдельной ломаной линией, но в отличие от первого вида значения каждого нового ряда откладываются не от оси X, а от значения предыдущего ряда, таким образом отображается изменение общей суммы значений для нескольких рядов данных. То есть показывается тенденция суммарных значений и в то же время дается представление о вкладе каждого ряда. Причем нужно отметить, что подводя курсор к точкам на графиках, можно увидеть значение конкретного ряда в соответствующей точке.

3-й вид — нормированный график. Показывает в процентах вклад каждой точки данных в итоговую сумму для данной категории. Поэтому последний ряд всегда вырождается на диаграмме в прямую, параллельную оси X, соответствующую 100%, так как сумма всех значений в любой точке есть 100%.

4-й, 5-й, 6-й виды аналогичны 1-му, 2-му и 3-му, но с выводом маркеров данных на линиях графиков.

7-й вид представляет собой объемный вариант графика, где каждый ряд изображается не ломаной линией, а ломаной лентой.

Гистограмма. Гистограммами называются вертикально ориентированные столбчатые диаграммы. Гистограммы удобны для сравнения дискретных значений из нескольких рядов данных. Так как точки в рядах данных не соединены линиями, то эти диаграммы менее удобны для представления тенденций, чем графики. Гистограммы используют для сравнения различных величин за один период или для прослеживания изменения отдельного показателя с течением времени.

Аналогично типу *График* гистограммы также имеют виды «с накоплением» и «нормированный», которые показывают вклад каждой точки данных в итоговую сумму для данной категории.

Линейчатая. Линейчатыми диаграммами называются горизонтально ориентированные столбчатые диаграммы. Для этого типа диаграмм ось X становится вертикальной, а ось Y — горизонтальной. Эти диаграммы хорошо иллюстрируют различные величины за один и тот же период времени.

Круговая. Круговые диаграммы показывают относительный вклад каждой точки данных в общий итог для этого ряда данных, т.е. показывает соотношение между целым и его частями. Например, общий бюджет и бюджетные статьи. Особенностью круговых диаграмм является то, что они отображают только один ряд данных. Сектор, соответствующий первой точке ряда, отображается от вертикальной оси, соответствующей 12 часам. Остальные сектора располагаются по часовой стрелке от него.

Кольцевая. Кольцевая диаграмма аналогично круговой показывает, как соотносятся части с целым, маркеры данных отображаются в виде сегментов кольца. Но в отличие от круговой диаграммы кольцевая позволяет отобразить несколько рядов данных, при этом каждый новый ряд отображается в собственном кольце, внешнем по отношению к предыдущему ряду.

Точечная. В точечной диаграмме в отличие от других диаграмм обе оси являются осями значений. Поэтому обычно точечные диаграммы используются для определения типа зависимости между двумя рядами данных, в частности, точечные диаграммы используют в статистике. Для изображения каждой точки используется пара координат, одна для ряда X и одна для ряда Y. В выделенном диапазоне клеток левый столбец (или верхняя строка) данных представляет *ряд X*, а каждый последующий столбец (строка) — *ряд значений Y*.

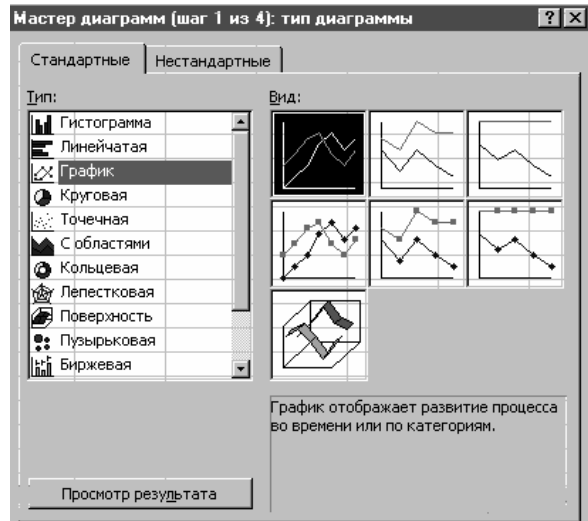
Биржевая. Биржевая диаграмма может использоваться для слежения за ценой акций, отсюда и название этого типа диаграмм. Этот тип диаграмм имеет 4 вида. Остановимся на первом типе подробнее.

1-й вид — «*мини-макс-закрытие*». Данные в исходном диапазоне должны быть расположены в строго определенном порядке: 1-й столбец (строка) — максимальные цены, 2-й столбец (строка) — минимальные цены, 3-й столбец (строка) — цены закрытия. Этот тип диаграммы отображает индекс Доу-Джонса. На диаграмме данные выводятся в виде вертикальных отрезков: верхний конец отрезка — наибольшее значение, нижний конец — наименьшее значение, горизонтальная метка — заключительное значение.

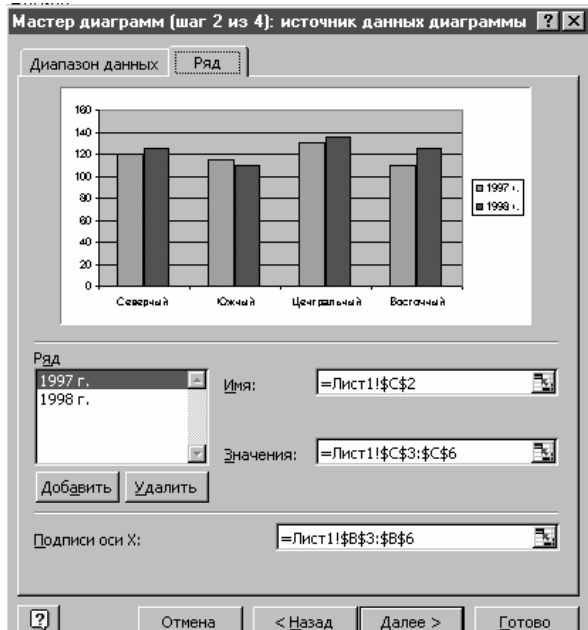
1.5.4.3. Построение и редактирование диаграммы

Для построения диаграммы в Excel используется *Мастер диаграмм*, вызвать который можно, нажав кнопку панели инструментов или используя команду *Вставка/Диаграмма*. Мастер диаграмм строит диаграммы за 4 шага:

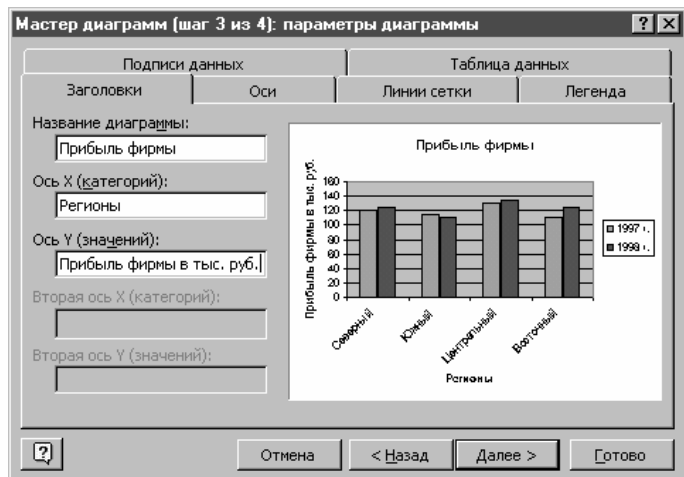
1-й шаг — выбор типа и вида диаграммы;



2-й шаг — определение источника данных диаграммы (указание диапазона данных и размещения рядов данных);



3-й шаг — задание параметров диаграммы (заголовков, легенды, подписей данных и т.д.);



4-й шаг — размещение диаграммы (на отдельном листе или на имеющемся).

В Excel достаточно легко редактировать диаграммы, если что-то в полученной диаграмме вас не устраивает.

1.5.4.4. Форматирование любого элемента диаграммы

1. Двойной щелчок на любом элементе диаграммы вызывает соответствующее диалоговое окно форматирования.

2. Выберите соответствующий объект из списка *Элементы диаграммы* на панели инструментов *Диаграммы* (вывести эту панель инструментов на экран можно командой *Вид/Панели инструментов/Диаграммы*), а затем щелкните на кнопке, расположенной правее, чтобы вызвать окно форматирования выбранного объекта.

3. Щелкните на элементе диаграммы правой кнопкой мыши и выберите нужный пункт из контекстного меню.

Контрольные вопросы

1. Что такое табличный процессор или электронная таблица?
2. Как сохранить и открыть книгу ЭТ?
3. Опишите структуру экрана табличного процессора.
4. Перечислите возможности табличных процессоров.
5. Что такое ячейка в ЭТ, блок ячеек?
6. Объясните основные понятия — книга, лист ЭТ.
7. Какие типы данных вы знаете?
8. Что такое относительный и абсолютный адрес ячейки?
9. Каковы особенности данных типа дата?
10. Что такое формула в ЭТ?
11. Как вводятся функции в формулы?
12. Какие типы диаграмм вы знаете?
13. Назовите основные элементы диаграмм.
14. Какие 4 шага нужно выполнить, чтобы построить диаграмму?
15. Объясните назначение логических функций И, ИЛИ, ЕСЛИ.
16. Назовите правила организации справочников в Excel.
17. Для чего предназначена функция ВПР?

Литература

1. Microsoft Excel 97. Шаг за шагом: Практическое пособие. — М.: ЭКОМ, 1997.
2. *Евдокимов В.В. и др.* Экономическая информатика: Учебник для вузов. — СПб.: Питер, 1997. — С.352–391.
3. Информатика: Базовый курс / С.В.Симонович и др. — СПб.: Питер, 2001. — С.302–327.
4. Информатика: Учебник / Под ред. проф. Н.В.Макаровой. — М.: Финансы и статистика, 1998. — С.528–558.
5. *Лавренов С.М.* Excel: Сборник примеров и задач. — М.: Финансы и статистика, 2000.
6. *Додж М., Кината К., Стинсон К. и др.* Эффективная работа с Excel 7.0 для Windows 95. — СПб.: Питер, 1997.
7. Практикум по экономической информатике: Учебное пособие. Ч.1 / Под ред. Е.Л.Шуримова, Н.А.Тимаковой, Е.А.Мамонтовой. — М.: Перспектива, 2000. — С.125–239.
8. Экономическая информатика / Под ред. П.В.Конюховского и Д.Н.Колесова. — СПб.: Питер, 2000. — С.210–258.
9. *Гарнаев А.Ю.* Использование MS Excel и VBA в экономике и финансах. — СПб.: БХВ, 1999.
10. Информатика: Сборник лабораторных работ в табличном процессоре Excel. — Новосибирск: НГАЭиУ, 2000.

ТЕМА 1.6. СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

1.6.1. ОСНОВНЫЕ ПОНЯТИЯ БАЗ ДАННЫХ

В настоящее время управление организацией, компанией или предприятием немыслимо без обработки информационных данных, которыми владеет это предприятие. При решении экономических, хозяйственных и финансовых задач зачастую приходится обрабатывать большие специфически структурированные и взаимозависимые массивы данных. Такие сложные массивы данных называют базами данных.

База данных (БД) — это поименованная и организованная (структурированная) совокупность взаимосвязанных данных, которые отражают состояние объектов конкретной предметной области и находятся под центральным программным управлением. База данных удерживает такой минимальный излишек информации, который позволяет удовлетворить запросы каждого из многочисленных пользователей.

Предметная область — часть реальной среды, которая отражается и описывается в базе данных.

Создание баз данных и обеспечение доступа к ней пользователей осуществляется с помощью систем управления базами данных.

Система управления базами данных (СУБД) — это комплекс программных и языковых средств, необходимых для создания баз данных, поддержания их в актуальном состоянии и организации поиска в них необходимой информации.

По технологии обработки данных базы данных делятся на централизованные БД и распределенные БД.

Централизованная БД хранится в памяти одной вычислительной системы. Эта вычислительная система может быть компонентом сети ЭВМ. Системы централизованных баз данных с сетевым доступом организуются по двум архитектурам: «файл-сервер» и «клиент-сервер».

Распределенная БД состоит из нескольких частей, которые хранятся в разных ЭВМ вычислительной сети.

Теория управления базами данных как самостоятельная дисциплина начала развиваться приблизительно с начала 50-х годов двадцатого столетия. За это время в ней сложилась определенная система фундаментальных понятий. Приведем некоторые из них.

Объектом называется элемент информационной системы, сведения о котором хранятся в базе данных. Иногда объект также называют сущностью (от англ. entity).

Поле — это элементарная единица логической организации данных, которая соответствует отдельной, неделимой единице информации — реквизиту. Поле может иметь следующие характеристики: имя, тип, длина, точность.

Запись — это совокупность полей, соответствующих логически связанным реквизитам. Структура записи определяется составом и последовательностью входящих в нее полей.

Экземпляр записи — реализация записи, содержащая конкретные значения полей.

Файл — это множество одинаковых по структуре экземпляров записей.

Первичный ключ — это одно или несколько полей, однозначно идентифицирующих запись. Если первичный ключ состоит из одного поля, он называется простым, если из нескольких полей — составным ключом. По первичному ключу можно найти один экземпляр записи.

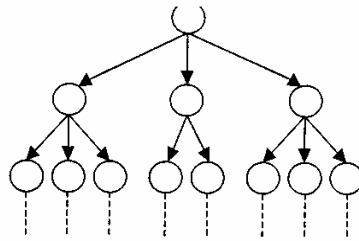
Вторичный ключ — это поле, значение которого может повторяться в нескольких записях. По вторичному ключу можно найти несколько экземпляров записи. Вторичные ключи исполняют роль поисковых или группировочных признаков.

Ядром любой базы данных является модель данных.

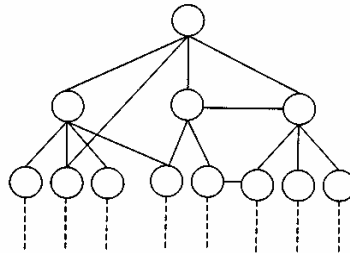
Модель данных — совокупность структур данных и операций их обработки.

В теории систем управления базами данных выделяют модели трех основных типов: иерархическую, сетевую и реляционную.

В иерархической модели данных элементы связаны отношениями подчиненности и при этом любой элемент может подчиняться только одному какому-нибудь другому элементу. Такую форму зависимости удобно изображать с помощью древовидного графа.

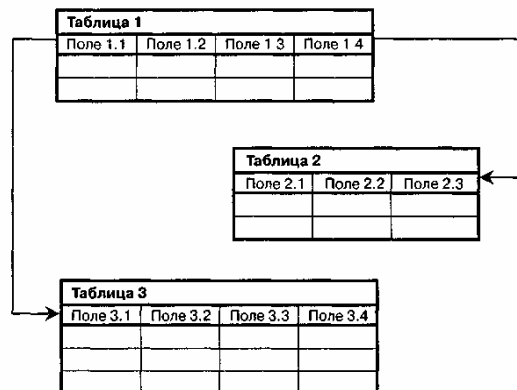


Сетевой подход к организации данных является расширением иерархического. В иерархических структурах запись-потомок должна иметь в точности одного предка; в сетевой структуре данных потомок может иметь любое число предков.



Среди достоинств систем управления данными, основанных на иерархической или сетевой моделях, могут быть названы их компактность и, как правило, высокое быстродействие, а среди недостатков — неуниверсальность, высокая степень зависимости от конкретных данных.

В реляционной модели объекты и взаимосвязи между ними представляются с помощью таблиц.



Информация в реляционных базах данных сохраняется в таблицах, связь между которыми осуществляется путем совпадения значений одного или нескольких полей. В реляционной базе данных каждая таблица должна иметь первичный ключ — поле или комбинацию полей, которые единственным образом идентифицируют каждую строку в таблице.

Предметную область в реляционных моделях данных представляют в виде соответствующего числа таблиц.

Таблицы находятся между собой в соответствующих отношениях (связях). Собственно говоря, термин «реляционная» происходит от английского relation — отношение.

Благодаря своей простоте и естественности представления реляционной модели на ПК она в настоящее время получила наибольшее распространение в системах автоматизированной обработки данных. Одной из самых популярных систем управления базами данных, построенных на основе реляционных моделей, является СУБД Microsoft Access.

1.6.2. РАБОТА СО СПИСКАМИ ДАННЫХ В EXCEL

При работе с таблицами большую помощь могут оказать содержащиеся в Excel средства работы с базой данных. По сути, таблица в Excel представляет собой однотабличную базу данных. В Excel базы данных называются списками.

Список — это набор строк с постоянным количеством и заголовками столбцов и переменным количеством строк. В каждом столбце должна содержаться однотипная информация, и недопустимы пустые ячейки.

Основными задачами при работе с базами данных являются:

- организация ввода данных;
- просмотр данных;
- сортировка данных;
- фильтрация данных;
- подведение итогов.

Весь инструмент работы со списками (базами данных) в Excel сосредоточен в одном пункте меню — *Данные*.

Табличный процессор Excel предоставляет пользователю удобное средство упорядочивания (сортировки), поиска (фильтрации) данных списка и создания отчетов на их основе. Сортировка данных может осуществляться в алфавитном, числовом и хронологическом порядке. Фильтрация позволяет осуществлять быстрый поиск и работу с определенными подмножествами данных без их перемещения или сортировки. То есть возможности, которые предоставляет Excel для работы со списками, это есть функции работы с базами данных.

База данных (БД) — это таблица, организованная следующим образом:

- столбец — поле записи;
- первая строка — имена полей;
- остальные строки — записи БД.

Если заголовки граф расположены в нескольких строках электронной таблицы, то в качестве имен полей используется только последняя строка заголовка, именно эта строка и включается в диапазон списка (базы данных). Например, в таблице на рис.6 ячейки B2:E2 не включены в диапазон списка.

	A	B	C	D	E	F
1						
2				Адрес		
3		Ф.И.О.	Год рождения	Улица, № дома	№ кв.	
4		Сомов А.П.	1954	Титова 12	4	
5		Чащина О.И.	1962	Кирова 17	35	
6		Панин Е.В.	1948	К.Маркса 23	58	
7						

Поле записи

Имя поля записи

Запись

Список (база данных) - диапазон B3:E6

Рис.6. Структура списка (базы данных)

Все поля базы данных должны иметь уникальные имена, т.е. не может быть в одной базе данных два поля с одинаковыми именами. Например, в таблице на рис.7 ячейки B3:D3 не могут быть строкой имен полей, так как имена полей в ячейках C3 и D3 совпадают.

	A	B	C	D	E
1					
2		Наименование	Оптовая	Розничная	
3		товара	цена в руб.	цена в руб.	
4					
5					
6					
7					

Рис.7. Пример неправильного оформления списка

Не рекомендуется объединять ячейки, расположенные в разных строках заголовков граф таблицы (как указано на рис.8), так как имя поля должно занимать только одну ячейку.

	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						
7						

Рис.8. Пример неправильного форматирования шапки таблицы-списка

Организация ввода и просмотра данных:

- Новую запись в БД можно ввести обычным образом, введя новую строку таблицы.
- Ввести и просмотреть информацию можно при помощи форм данных. Каждая форма данных настроена на выделенный список и содержит поля для ввода данных во все поля списка. Чтобы вызвать форму данных, нужно выделить диапазон списка и вызвать команду *Данные/Форма*.
- Задание *Критерия* в форме позволяет просматривать через окно формы только записи, удовлетворяющие условиям критерия.

Сортировка всех данных списка производится с помощью команды *Данные/Сортировка*. При выполнении этой команды появляется диалоговое окно, в котором пользователь может выбрать поле, по которому производится сортировка, а также установить для различных полей разные приоритеты (по какому полю сортировать в первую, вторую и в последнюю очередь), выбрать критерии сортировки (по возрастанию или убыванию). Пользователь должен быть внимательным при определении диапазона списка, данные которого необходимо сортировать: сортировать весь список или определенный диапазон списка.

Фильтрация данных. Excel дает возможность выборочно работать с данными, удовлетворяющими определенным условиям. Отфильтровать список — значит, скрыть все строки за исключением тех, которые удовлетворяют заданным условиям отбора. *Фильтрация списка* дает возможность выводить на экран необходимое подмножество элементов списка. Закончив работу с ограниченным числом записей, можно снова вывести на экран весь список. Для фильтрации списков Excel предоставляет две команды: *Автофильтр* и *Расширенный фильтр*.

Если при обращении к команде *Данные/Фильтр* в меню выбрать подкоманду *Автофильтр*, то в метке каждого столбца появляется кнопка окна списка. При раскрытии списка в каком-либо столбце выводится список всех уникальных элементов этого столбца и некоторые критерии отбора (Первые 10, Все, Пустые, Непустые), в том числе пользовательские критерии отбора (*Условие*). При выборе определенного критерия на экране остаются только те элементы списка, которые соответствуют выбранному критерию.

Примечание: После фильтрации списка в информации, расположенной слева или справа от списка, также будут скрыты те же строки, что и в списке. Поэтому лучше всего другую информацию помещать ниже или выше списка.

Если нужно вывести отфильтрованные строки в другую часть рабочего листа или на другой лист, можно использовать команды *Копировать* и *Вставить* из меню *Правка* или команду *Расширенный фильтр*.

Команда *Данные/Фильтр/Расширенный фильтр* предоставляет пользователю более широкие возможности по созданию сложных критериев фильтрации. При работе с расширенным фильтром в специальном диапазоне условий создается критерий по определенным правилам самим пользователем. Результат фильтрации может быть показан на месте исходного списка или скопирован в новое место.

При работе расширенный фильтр опирается на три области.

Область данных — база данных или список.

Область критериев поиска (диапазон условий). Диапазон условий должен содержать по крайней мере две строки: строку заголовков полей, которые будут ключевыми при отборе записей, т.е. используются в условиях отбора (для обеспечения точности эти заголовки лучше копировать

из области данных), и строку (или строки) критериев. При записи критериев нужно соблюдать следующие правила:

- критерии могут содержать константы, операции отношения, формулы;
- в критериях могут применяться шаблоны (символы ? и *);
- условия можно записывать в виде: <знак отношения><значение>. В критериях можно использовать все знаки отношения (>, >=, <, <=, <>), кроме знака = (равно), так как с этого символа в Excel начинаются формулы;
- в диапазоне критериев можно ввести любое количество условий. Если условия находятся в одной строке, то они считаются соединенными логическим оператором И, если в разных — то логическим оператором ИЛИ.

Целевая область (область вывода). Ее задание необязательно (так как можно оставить результаты отбора на месте). В целевую область копируют имена тех полей записей, которые требуется вывести. Количество и порядок этих полей могут быть любые. Необходимо позаботиться о том, чтобы ниже имен полей в целевой области было достаточно пустых строк для вывода результата.

Ниже приводятся примеры запросов (критериев поиска) для списка (базы данных), представленного на рис.9.

Пример 1. Отобрать записи и вывести фамилию, возраст и оклад тех сотрудников, у кого оклад больше 3000.

В критерий данного примера запишем условие в виде отношения: >3000.

Область данных — A3:D12.

Диапазон условий (F3:F4).

Целевая область (A15:C15).

	A	B	C	D	E	F	G	H
1								
2								
3	Ф.И.О.	Телефон	Возраст	Оклад		Оклад		
4	Белова С.А.	46-78-91	43	4500		>3000		
5	Буров А.В.	22-33-44	39	4500				
6	Буров С.Н.		24	2100				
7	Жуков П.Г.	45-65-83	34	3800				
8	Киселев В.П.		52	3800				
9	Котов Р.Л.	43-44-45	31	4500				
10	Петрова Р.И.		29	2100				
11	Рыжова А.В.	21-34-56	24	2500				
12	Суворова С.Г.	46-45-44	46	3200				
13								
14								
15	Ф.И.О.	Возраст	Оклад					
16								
17								

Рис.9. Область данных, диапазон условия и область вывода для примера 1

После того, как сформированы все области, выбираем команду *Данные/Фильтр/Расширенный фильтр*. В появившемся диалоговом окне заполняем все поля, не вводя адреса диапазонов с клавиатуры, а выделяя соответствующие диапазоны на электронной таблице.

Наша задача — научиться правильно записывать критерии поиска. Поэтому в следующих примерах мы разберем, как формируются диапазоны критериев.

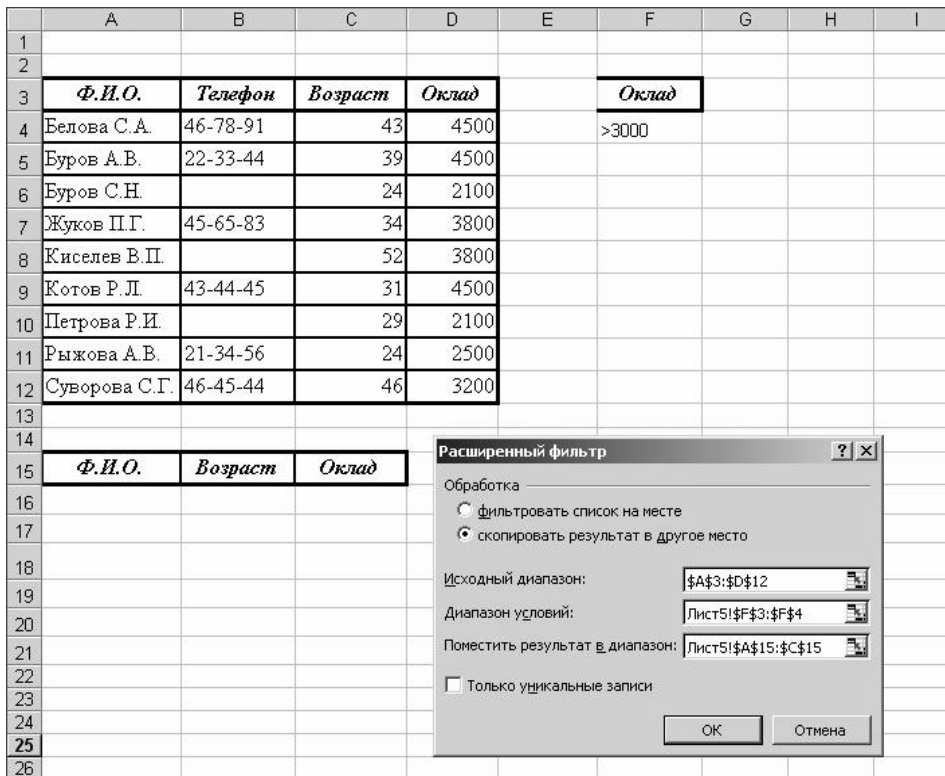


Рис.10. Пример работы с окном Расширенный фильтр

Пример 2. Отобрать записи сотрудников, возраст которых 24 года.

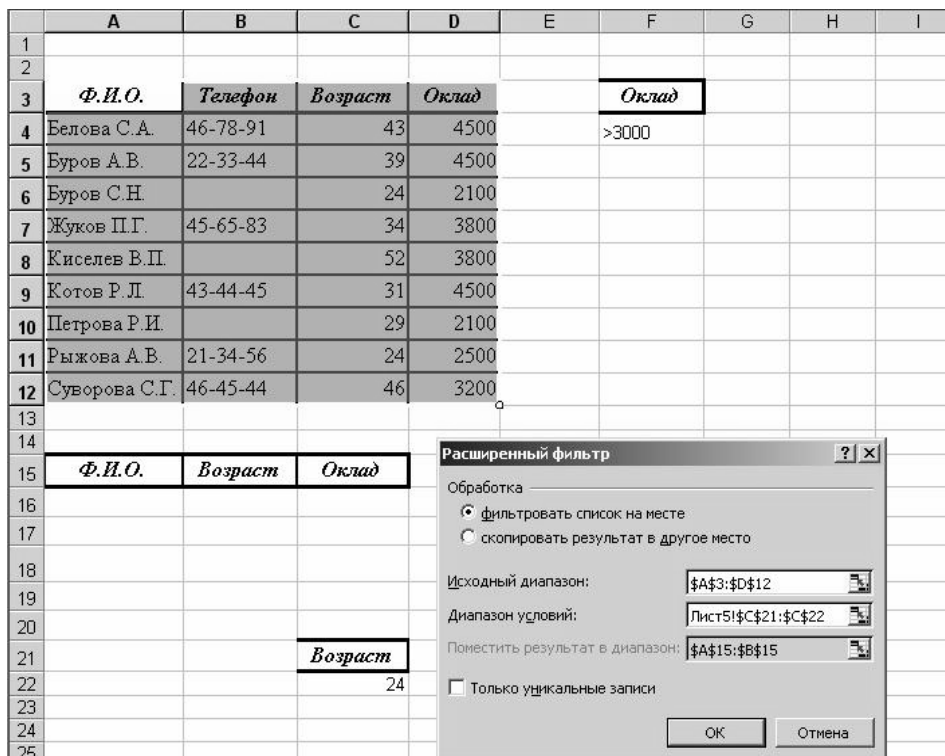
В этом условии знак равно не ставится.

Диапазон условий (C21:D22):

	С
21	Возраст
22	24

Для данного запроса список мы отфильтруем на месте.

Чтобы вернуть исходную таблицу после фильтрации на месте, необходимо выбрать команду *Данные/Фильтр/Показать все.*



Пример 3. Отобрать записи тех сотрудников, у кого оклад больше 3000 и возраст меньше 35 лет.

В этом примере два условия соединяются логической операцией И (И *Оклад*>3000, И *Возраст*<35), следовательно, в диапазоне критерия они располагаются в одной строке.

Диапазон условий — четыре ячейки (C24:D25):

	С	D
24	Оклад	Возраст
25	>3000	<35

Пример 4. Отобрать записи сотрудников младше 25 и старше 50 лет.

В этом примере два условия соединяются логической операцией ИЛИ (ИЛИ *Возраст*<25, ИЛИ *Возраст*>50), следовательно, в диапазоне критерия они располагаются в разных строках одного столбца (одного поля — *Возраст*).

Диапазон условий — три ячейки (C26:C28):

	С
26	Возраст
27	<25
28	>50

Пример 5. Отобрать записи сотрудников, у которых телефон начинается на цифры 46.

В этом примере при записи условия используем шаблон 46* (* заменяет любое количество любых символов). То есть будут отобраны записи, в которых значение поля *Телефон* начинается на 46.

Диапазон условий (C30:C31):

	С
30	Телефон
31	46*

Вычисляемые условия отличаются от обычных условий тем, что позволяют использовать значения, возвращаемые формулой.

Три правила записи вычисляемых условий:

- Заголовок над вычисляемым условием должен отличаться от любого имени поля в списке, он может быть пустым или содержать произвольный текст. Если заголовок пустой, он все равно включается в диапазон условия.
- Ссылки на ячейки, находящиеся вне списка, как правило, должны быть абсолютными.
- Ссылки на ячейки, находящиеся в списке, как правило, должны быть относительными.

Пример 6. Отобрать записи сотрудников, у которых оклад больше среднего оклада.

В этом примере используем вычисляемое условие. В левой части этого условия указываем адрес ячейки D4, которая является первой ячейкой с данными в столбце *Оклад*. В процессе фильтрации вместо этого адреса подставляются по порядку адреса всех остальных ячеек поля *Оклад* до конца списка, т.е. до ячейки D12.

Диапазон условий (C33:C34):

	С
33	
34	=D4>CP3НАЧ(\$D\$4:\$D\$12)

или диапазон условий (C33:C34):

	С
33	Оклад выше среднего
34	=D4>CP3НАЧ(\$D\$4:\$D\$12)

Встроенные функции для работы с базами данных. Excel имеет категорию функций, предназначенных для работы с базами данных (списками). Эти функции являются аналогами обычных статистических функций, но оперируют только с элементами списка, которые удовлетворяют заданным условиям.

Синтаксис таких функций в общем виде:

<функция>(база данных; поле; диапазон условия)

где *поле* — имя поля (столбца) в кавычках или его порядковый номер в списке, в котором должны производиться вычисления.

Пример 7. В ячейке A21 определить минимальный возраст сотрудников, оклад у которых больше 4000.

	А	С
21	=ДМИН (A3:D12; 3; B21:B22)	Оклад
22		>4000

Или (второй вариант)

	А	С
21	=ДМИН (A3:D12; «Возраст»; B21:B22)	Оклад
22		>4000

Пример 8. Подсчитать количество сотрудников старше 40 лет.

	А	С
24	=БСЧЕТА(A3:D12;1;B24:B25)	Возраст
25		>40

Пример 9. Определить сумму окладов сотрудников младше 30 лет.

	А	С
27	=БДСУММ(A3:D12;4;B27:B28)	Возраст
28		<30

Подведение итогов. Агрегирование данных списков состоит в формировании промежуточных итогов, создании консолидированных и сводных таблиц.

Часто требуется знать промежуточные суммы (или среднее значение, количество значений, максимум, минимум и др.) для тех или иных таблиц. Можно, конечно, вычислить результат по диапазону ячеек, но удобнее воспользоваться инструментом *Промежуточные итоги*. Для формирования промежуточных итогов требуется предварительная сортировка списка по полям группировки записей. Команда меню *Данные/Итоги* позволяет сформировать итоги определенного вида по указанным (отсортированным) полям при изменении значения поля группировки.

Консолидация. Еще одним полезным средством обработки данных в Excel является консолидация. Консолидация — особый способ вычисления итогов для диапазона ячеек. Консолидируемые данные могут находиться на одном и том же или нескольких листах одной рабочей книги, а также различных рабочих книг. Все такие таблицы должны иметь абсолютно идентичную структуру. Соединение не является механическим. Итоговая таблица будет содержать только одну строку с ключевым полем, а числовые данные в ней будут суммами (или другими функциями — при консолидации доступны все функции статистических итогов: сумма, среднее, максимум, минимум и т.д.) всех строк объединенного подмножества.

Ключевым полем считается самое левое поле (колонка) таблицы или ее выделенной части. Консолидация осуществляется командой *Данные/Консолидация*.

Сводные таблицы являются весьма мощным и гибким инструментом анализа и трансформации данных в среде Microsoft Excel. С его помощью можно сделать практически любой «разрез» таблицы, получить итоги по любым данным. *Сводная таблица* во многом объединяет возможности других обобщающих инструментов, таких как *Итоги*, *Фильтр*, *Консолидация*. С помощью *Сводной таблицы* можно произвольным образом изменить структуру исходных данных, располагая заголовки строк и столбцов удобным образом, а также получить все необходимые

промежуточные итоги и сортировки. Построение сводной таблицы осуществляется с помощью *Мастера сводных таблиц*, вызываемого командой *Данные/Сводная таблица*.

В результате ее выполнения создается настраиваемая таблица для организации полей на листе в новых сочетаниях. Положение всех элементов сводной таблицы пользователь в процессе анализа может изменять, располагая их в желаемом порядке.

Для формирования сводных таблиц Excel предлагает использовать *Мастер сводных таблиц*, который за четыре шага формирует нужную сводку:

1-й шаг — определение типа источника данных.

2-й шаг — определение адреса источника данных.

3-й шаг — определение структуры сводной таблицы. Это самый важный шаг. На этом шаге предлагается определить структуру формируемого документа. В правой части окна отображаются кнопки-названия всех полей, участвующих в сводке, которые с помощью мыши можно перетаскивать в свободные области структуры документа.

В области *Столбец* и *Строка* переносятся поля, сведения о которых будут находиться на соответствующих осях формируемой таблицы. Здесь имеет значение относительный порядок расположения полей: поле, находящееся первым (выше или левее), является старшим по отношению к следующему. Это означает, что одному значению первого поля будут соответствовать все значения второго.

В поле *Данные* будут отображаться собственно данные — суммы, среднее, минимум, доли и т.д.

4-й шаг — указание местоположения готовой сводной таблицы.

Контрольные вопросы

1. Что такое база данных?
2. Для чего предназначены СУБД?
3. Что такое запись, поле записи в базе данных?
4. Что такое первичный и вторичный ключи?
5. Охарактеризуйте каждый из типов моделей данных.
6. Что такое база данных в Excel? Перечислите основные задачи при работе с базами данных.
7. Что такое фильтрация списка, и какие способы фильтрации Вы знаете?
8. Каковы правила составления критерия при фильтрации списков?
9. Каковы особенности работы с функциями категории «Работа с базой данных»?
10. Что такое промежуточные итоги и как их получить?
11. Охарактеризуйте операцию консолидации.
12. Что такое сводные таблицы?

Литература

1. Microsoft Excel 97. Шаг за шагом: Практическое пособие. — М.: ЭКОМ, 1997.
2. *Евдокимов В.В. и др.* Экономическая информатика: Учебник для вузов. — СПб.: Питер, 1997. — С.352–391.
3. Информатика: Базовый курс / С.В.Симонович и др. — СПб.: Питер, 2001. — С.302–327.
4. Информатика: Учебник / Под ред. проф. Н.В.Макаровой. — М.: Финансы и статистика, 1998. — С.528–558.
5. *Лавренов С.М.* Excel: Сборник примеров и задач. — М.: Финансы и статистика, 2000.
6. Практикум по экономической информатике: Учебное пособие. Ч.1 / Под ред. Е.Л.Шуримова, Н.А.Тимаковой, Е.А.Мамонтовой. — М.: Перспектива, 2000. — С.125–239.
7. Экономическая информатика / Под ред. П.В.Конюховского и Д.Н.Колесова. — СПб.: Питер, 2000. — С.210–258.
8. *Гарнаев А.Ю.* Использование MS Excel и VBA в экономике и финансах. — СПб.: БХВ, 1999.
9. Информатика: Сборник лабораторных работ в табличном процессоре Excel. — Новосибирск: НГАЭиУ, 2000.

ТЕМА 1.7. МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ. ЛОКАЛЬНЫЕ И ГЛОБАЛЬНЫЕ СЕТИ ЭВМ

1.7.1. ОСНОВЫ И МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

Информация является изначально слабозащищенным ресурсом и поэтому представляется чрезвычайно важным принимать повышенные меры ее защиты. Защита информации являлась проблемой на разных этапах развития человечества. Бурное развитие информационных технологий сделали эту проблему особенно актуальной и острой.

Одна из тенденций, характеризующая развитие современных информационных технологий, — рост числа компьютерных преступлений и связанных с ними материальных потерь. Сегодня появилась новая современная технология — технология защиты информации в компьютерных информационных системах и в сетях передачи данных.

Среди причин активизации компьютерных преступлений отмечают:

- переход от традиционной «бумажной» технологии хранения и передачи сведений на электронную и недостаточное при этом развитие технологии защиты информации;
- объединение вычислительных систем, создание глобальных сетей и расширение доступа к информационным ресурсам;
- увеличение сложности программных средств и связанное с этим уменьшение их надежности и увеличение числа уязвимых мест.

Под угрозой безопасности информации понимается действие или событие, которое может привести к разрушению, искажению или несанкционированному использованию информационных ресурсов, включая хранимую, передаваемую и обрабатываемую информацию, а также программные и аппаратные средства.

Среди основных угроз безопасности информации можно выделить:

- раскрытие конфиденциальности информации (несанкционированный доступ к базам данных, прослушивание каналов и т.д.);
- компрометация информации (например, несанкционированные изменения базы данных и т.п.);
- несанкционированное использование информационных ресурсов;
- несанкционированный обмен информацией и т.д.

Наиболее распространенными путями несанкционированного доступа к информации, являются:

- перехват электронных излучений;
- применение подслушивающих устройств;
- хищение носителей информации и документальных отходов;
- чтение остаточной информации в памяти системы после выполнения санкционированных запросов;
- копирование носителей информации с преодолением мер защиты;
- маскировка под зарегистрированного пользователя;
- использование программных ловушек;
- незаконное подключение к аппаратуре и линиям связи;
- злоумышленный вывод из строя механизмов защиты;
- внедрение и использование компьютерных вирусов и т.д.

Для решения проблемы защиты информации основными средствами защиты информации принято считать:

1. *Технические средства* — реализуются в виде электрических, электромеханических, электронных устройств. Всю совокупность технических средств принято делить на:

- *аппаратные* — устройства, встраиваемые непосредственно в аппаратуру, или устройства, которые сопрягаются с аппаратурой систем обработки данных по стандартному интерфейсу (схемы контроля информации по четности, схемы защиты полей памяти по ключу, специальные регистры);
- *физические* — реализуются в виде автономных устройств и систем (электронно-механическое оборудование охранной сигнализации и наблюдения и т.п.).

2. *Программные средства* — программы, специально предназначенные для выполнения функций, связанных с защитой информации.

3. *Организационные средства* — организационно-правовые мероприятия, осуществляемые в процессе создания и эксплуатации систем обработки данных для обеспечения защиты информации.

4. *Законодательные средства* — законодательные акты страны, которыми регламентируются правила использования и обработки информации ограниченного доступа и устанавливаются меры ответственности за нарушение этих правил.

Однако эти способы не обеспечивают надежной защиты информации, поэтому рекомендуется использовать дополнительные способы защиты информации.

Управление доступом, включающее следующие функции защиты:

- идентификацию пользователя (присвоение персонального имени, кода, пароля и опознание пользователя по предъявленному идентификатору);
- проверку полномочий, соответствие дня недели, времени суток, запрашиваемых ресурсов и процедур установленному регламенту;
- разрешение и создание условий работы в пределах установленного регламента;
- регистрацию обращений к защищаемым ресурсам;
- реагирование (задержка работ, отказ, отключение, сигнализация) при попытках несанкционированных действий.

Маскировка — способ защиты информации в системах обработки данных путем ее криптографического шифрования. При передаче информации по линиям связи большой протяженности криптографическое закрытие является единственным способом ее надежной защиты.

Механизм цифровой подписи, основывающийся на алгоритмах шифрования и включающий две процедуры: формирование подписи отправителя и ее распознавание (верификацию) получателем и др.

1.7.2. РЕЗЕРВИРОВАНИЕ ФАЙЛОВ

Одним из методов защиты информации является резервирование файлов.

При работе с файлами на МД может произойти порча или потеря информации по различным причинам: физическая порча МД, неправильная корректировка файлов, случайное уничтожение информации и т.д.

Резервированием файлов называется создание их копий на машинных носителях информации и систематическое их обновление в случае изменения резервируемых файлов.

Различают несколько способов резервирования файлов:

- *простое копирование* одного или нескольких файлов (или файловой структуры, т.е. папок с входящими в них файлами) на таком же или другом носителе информации (например, на диске, стримере). В этом случае резервные копии занимают столько же места, сколько и исходные файлы. Копия может быть использована в дальнейшем без каких-либо преобразований;
- *резервирование файлов*, т.е. сохранение файлов на другом носителе информации в одном файле. В дальнейшем для использования этих файлов их предварительно необходимо извлечь из этого файла (такой процесс называется восстановлением файлов). Для резервирования и восстановления файлов применяют специальные утилиты;
- *архивное копирование файлов*, когда при переносе программ или данных с одного носителя информации на другой используют упаковку данных, т.е. сжатие, уплотнение информации.

Сжатие (упаковка) информации — это процесс преобразования информации, хранящейся в файле, к виду, при котором уменьшается избыточность в ее представлении и соответственно требуется меньший объем памяти для хранения.

Целью сжатия файлов обычно является не только обеспечение более компактного размещения информации на диске, но и сокращение времени и соответственно стоимости передачи информации по каналам связи в компьютерных сетях, упрощение переноса группы файлов с одного ПК на другой, сокращение времени копирования файлов.

Для архивации и упаковки файлов предназначен специальный вид сервисных программ — архиваторы.

Программа, осуществляющая сжатие и упаковку данных, называется *архиватором*. Все архиваторы позволяют осуществлять упаковку и распаковку файлов любого типа, простейшие операции с архивом: просмотр, обновление, удаление и добавление файлов в архив.

Архивный файл — это специальным образом организованный файл, содержащий в себе один или несколько файлов в сжатом (или несжатом) виде, из которого в случае необходимости их можно извлечь в первоначальном виде. Кроме того, архивный файл содержит и служебную информацию: оглавление имен файлов, информацию о размерах файлов, дате их создания, специальный код циклического контроля файлов для проверки целостности архива и т.д.

Степень сжатия файлов зависит от используемой программы, метода сжатия и типа исходного файла. В зависимости от реализованного способа уплотнения информации одни программы лучше архивируют текстовые файлы, другие — графические и т. д. При добавлении файлов в архив, как правило, указывается, на сколько процентов его удалось сжать при помещении в архив. Существуют методы, позволяющие уменьшить объем дискового пространства, занимаемого файлами, на 30–50%.

При выборе программ-архиваторов ориентируются чаще всего на скорость работы программы, удобство, коэффициент сжатия информации.

Наиболее популярными являются следующие программы-архиваторы: ARJ, WINZIP, WINRAR и др.

Грубых ошибок при работе архиваторов не отмечено. Большинство программ имеют сходные возможности архивирования, изучив которые, можно легко создать архив.

Большие по объему архивные файлы могут быть размещены на нескольких дисках (томах). Такие архивы называются *многотомными*.

Программы-архиваторы позволяют создавать и такие архивы, для извлечения из которых содержащихся в них файлов не требуются какие-либо программы, так как сами архивные файлы могут содержать программу распаковки. Такие архивные файлы называются *самораспаковывающимися*.

1.7.3. КОМПЬЮТЕРНЫЕ ВИРУСЫ И АНТИВИРУСНЫЕ СРЕДСТВА

Компьютерным вирусом называется специально написанная программа, способная самопроизвольно присоединяться к другим программам, создавать свои копии и внедрять их в файлы, системные области компьютера и в вычислительные сети с целью нарушения работы программ, порчи файлов и каталогов, создания всевозможных помех в работе на ПК.

Количество новых программных вирусов постоянно растет. Основными путями проникновения вирусов являются съемные диски (гибкие и лазерные), а также компьютерные сети. Заражение жесткого диска может произойти при загрузке дискеты, содержащей вирус.

Основные признаки проявления вирусов (правда, не всегда это вирус, могут быть и другие причины):

- медленная работа компьютера;
- невозможность загрузки операционной системы;
- исчезновение или искажение файлов и каталогов;
- изменение размеров файлов;
- неожиданное значительное увеличение количества файлов на МД;
- существенное уменьшение размера свободной оперативной памяти;
- частые зависания и сбои в работе компьютера;
- прекращение работы или неправильная работа ранее успешно функционировавших программ.

Существуют различные классификации вирусов.

В зависимости от среды обитания:

- *сетевые вирусы* — распространяются по различным компьютерным сетям. В частности сетевые вирусы распространяются, используя возможности электронной почты;

- *файловые вирусы* — используют особенности файловой организации системы. Такие вирусы внедряются главным образом в файлы с расширением com, exe. Файловые вирусы заражают файлы, внедряясь в начало, в конец или в середину файла. Некоторые вирусы вместо исходного

содержимого файла записывают свой код. Такие файлы уже не могут быть использованы по своему прямому назначению. Некоторые вирусы «присваивают» имя заражаемого файла, а тому дают новое имя. Тогда при запуске «зараженного» файла сначала отработает вирус, а потом сам файл (это делается для того, чтобы скрыть факт работы вируса);

- *загрузочные вирусы* — внедряются в загрузочный сектор диска (Boot-сектор) или в сектор, содержащий программу загрузки системного диска;

- *макровирусы* — это файловые вирусы, использующие особенности файлов документов, созданных популярными редакторами или табличными процессорами.

По способу заражения:

- *резидентные вирусы* — при заражении ПК оставляет в оперативной памяти свою резидентную часть, которая потом перехватывает обращение операционной системы к объектам заражения (файлам, загрузочным секторам дисков и т.д.) и внедряются в них. Резидентные вирусы находятся в оперативной памяти и являются активными вплоть до выключения или перезагрузки ПК;

- *нерезидентные вирусы* — не заражают память ПК и являются активными ограниченное время.

По особенностям алгоритма:

- *паразитические вирусы* — изменяют содержимое файлов и секторов диска и могут быть легко обнаружены и уничтожены;

- *вирусы-репликаторы*, называемые червями — распространяются по компьютерным сетям, вычисляя адреса сетевых компьютеров и записывают по этим адресам свои копии;

- *вирусы-невидимки*, называемые стелс-вирусами, — очень трудно обнаружить и обезвредить, так как они перехватывают обращение операционной системы к пораженным файлам и подставляют вместо своего тела незараженные участки диска;

- *вирусы-мутанты* — содержат алгоритмы шифровки-расшифровки, благодаря которым копии одного и того же вируса не имеют ни одной повторяющейся цепочки байтов;

- *квазивирусы* (тройские программы) — маскируясь под полезную программу, разрушают загрузочный сектор и файловую систему дисков.

Для обнаружения, удаления и защиты от компьютерных вирусов создаются специальные **антивирусные программы**. Все антивирусные программы делятся на следующие виды:

- *программы-детекторы* — осуществляют поиск характерной для конкретного вируса последовательности байтов (сигнатуры вируса) в оперативной памяти и в файлах и при обнаружении выдают об этом сообщения. Такие программы находят те вирусы, которые известны разработчикам программ-антивирусов;

- *программы-доктора*, или фаги, программы-вакцины — не только находят зараженные файлы, но и «лечат» их, т.е. удаляют из файлов тело программы-вируса, возвращая файлы в исходное состояние. Программы-полифаги уничтожают большое количество вирусов. К таким программам относятся Aidstest, Scan, Norton Antivirus, Doctor Web. Версии их постоянно обновляются, так как эти антивирусные программы быстро устаревают;

- *программы-ревизоры* — запоминают исходное состояние программ, каталогов и системных областей диска, когда ПК не заражен вирусом, а затем периодически (или по желанию пользователя) сравнивают текущее состояние с исходным. Примером такой программы является программа Adinf фирмы «Диалог-наука». Эти программы относятся к самым надежным средствам защиты от вируса;

- *программы-«сторожа»*, или фильтры — представляют собой небольшие резидентные программы, предназначенные для обнаружения подозрительных действий при работе ПК, характерных для вирусов. Они способны обнаружить вирус на самой ранней стадии его существования до размножения, но эти программы не «лечат» файлы и диски (для этого существуют другие программы, например, фаги). Примером таких программ является программа Vsafe;

- *программы-вакцины*, или иммунизаторы — предотвращают заражение файлов, причем вакцина возможна только от известных вирусов. Программа-вакцина модифицирует программу или диск таким образом, чтобы это не отразилось на их работе, а вирус будет воспринимать эти объекты зараженными и поэтому не будет внедряться. Такие программы имеют ограниченное применение.

Основные меры по защите от компьютерных вирусов:

- регулярное обновление версий программ-антивирусов;
- обязательная проверка перед работой дискет с информацией, записанной на других ПК;
- использование антивирусных программ для входного контроля всех исполняемых файлов, получаемых из компьютерных сетей;
- установка защиты от записи на дискеты при работе с ними на других компьютерах, если не нужно производить запись;
- проверка файлов после их разархивации на жестком диске при переносе на свой ПК архивированных файлов;
- периодическая проверка на наличие вирусов жестких дисков ПК (запуская антивирусные программы с защищенной от записи системной дискеты);
- создание архивных копий ценной информации на дискетах и др.

1.7.4. ВОЗНИКНОВЕНИЕ И НАЗНАЧЕНИЕ КОМПЬЮТЕРНЫХ СЕТЕЙ

Современное производство требует высоких скоростей обработки информации, удобных форм ее хранения и передачи. Для решения задач управления становятся важными и актуальными не только скорость и удобство обмена информацией, а также возможность тесного взаимодействия всех участвующих в процессе выработки управленческих решений.

В эпоху использования ЭВМ с пакетной обработкой информации в основном использовались компьютеры, на которых можно было решать почти все классы задач. При этом господствовал принцип централизованной обработки данных.



Рис.11. Система централизованной обработки данных

Принцип централизованной обработки данных не отвечал требованиям к надежности процесса обработки, затруднял развитие систем. Кратковременный выход из строя центральной ЭВМ нарушал функционирование системы в целом.

Появление малых ЭВМ, микроЭВМ и ПЭВМ способствовало появлению нового подхода к организации систем обработки информации. Возникло логически обоснованное требование перехода от использования отдельных ЭВМ в системах централизованной обработки данных к другой системе обработки данных. И в конце 70-х годов возникла тенденция распределенной обработки данных.

Распределенная обработка данных — обработка данных, выполняемая на независимых, но связанных между собой компьютерах, представляющих распределенную систему.

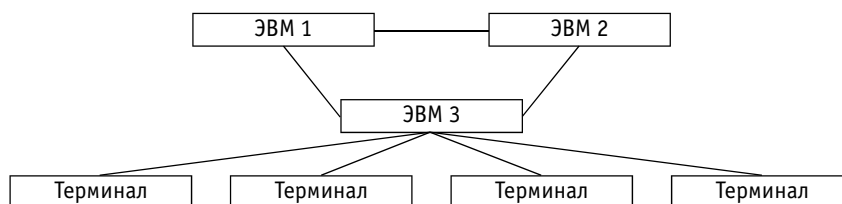


Рис.12. Система распределенной обработки данных

Для реализации распределенной обработки данных (РОД) были созданы *многомашинные ассоциации*, структура которых разрабатывается по одному из следующих направлений:

- многомашинные вычислительные комплексы (МВК). *Многомашинный вычислительный комплекс* — группа установленных рядом вычислительных машин, объединенных с помощью специальных средств сопряжения и выполняющих совместно единый информационно-вычислительный процесс;
- компьютерные (вычислительные) сети — высшая форма многомашинных ассоциаций.

Компьютерная (вычислительная) сеть — совокупность компьютеров и терминалов, соединенных с помощью каналов связи в единую систему, удовлетворяющую требованиям распределенной обработки данных.

В настоящее время создание вычислительных сетей становится наиболее важным применением компьютеров.

- Объединение компьютеров в сеть позволяет совместно использовать дорогостоящее оборудование — магнитные диски большой емкости, принтеры, основную память, иметь общие программные средства и данные.
- Глобальные сети предоставляют возможность использовать аппаратные ресурсы удаленных компьютеров.
- Глобальные сети полностью изменили процесс распространения и восприятия информации, сделали обмен информацией через электронную почту самой распространенной услугой сети, а основным ресурсом — информацию.
- Глобальные сети обеспечивают единое информационное пространство для миллионов пользователей ЭВМ.

Основным назначением компьютерной сети является обеспечение простого, удобного и надежного доступа пользователя к распределенным общесетевым ресурсам и организация их коллективного использования при надежной защите от несанкционированного доступа, а также обеспечение удобных и надежных средств передачи данных между пользователями сети.

Существует множество задач, нуждающихся в централизованных общих данных, удаленном доступе к базам данных, передаче данных на расстояние и их распределенной обработке. Примерами являются банковские и другие финансовые структуры; коммерческие системы, отражающие состояние рынка («спрос-предложение»); налоговые службы; системы резервирования авиабилетов; системы социального обеспечения и т.д.

Отличия компьютерных сетей от многомашинных вычислительных комплексов:

- *размерность*. В состав многомашинного вычислительного комплекса входят обычно две-три ЭВМ, расположенные преимущественно в одном помещении. Вычислительная сеть может состоять из десятков и сотен ЭВМ, расположенных на расстоянии друг от друга от нескольких метров до сотен и тысяч километров;

- *разделение функций между ЭВМ*. Если в многомашинном вычислительном комплексе функции обработки, передачи данных и управления системой могут быть реализованы в одной ЭВМ, то в вычислительных сетях эти функции распределены между различными ЭВМ;

- *необходимость решения в сети задачи маршрутизации сообщений*. Сообщение от одной ЭВМ к другой в сети может быть передано по различным маршрутам в зависимости от состояния каналов связи, соединяющих ЭВМ друг с другом.

Элементами многомашинной ассоциации являются:

- *абоненты сети* — объекты, генерирующие или потребляющие информацию в сети. Абонентами сети могут быть отдельные ЭВМ, комплексы ЭВМ, терминалы, промышленные роботы, станки с числовым программным управлением и т.д. Любой абонент сети подключается к станции;

- *станция* — аппаратура, которая выполняет функции, связанные с передачей и приемом информации. Совокупность абонента и станции принято называть *абонентской системой*.

Для организации взаимодействия абонентов необходима физическая передающая среда.

Физическая передающая среда — линии связи или пространство, в котором распространяются электрические сигналы, и аппаратура передачи данных.

На базе физической передающей среды строится *коммуникационная сеть*, которая обеспечивает передачу информации между абонентскими системами.

Любая компьютерная сеть — это совокупность абонентских систем и коммуникационной сети.

1.7.5. КЛАССИФИКАЦИЯ КОМПЬЮТЕРНЫХ СЕТЕЙ

Компьютерные сети можно классифицировать по ряду признаков.

В зависимости от территориального расположения абонентских систем вычислительные сети можно разделить на три основных класса.

Локальные сети (LAN — Local Area Network) связывают абонентов одного или нескольких близлежащих зданий одного предприятия, учреждения. Расстояние между ЭВМ в локальной сети небольшие — до 10 км, а при использовании радиоканалов связи — до 20 км. К классу локальных вычислительных сетей относятся сети отдельных предприятий, фирм, банков, офисов и т.д.

Региональные сети (MAN — Metropolitan Area Network) объединяют пользователей города, региона, отдельной небольшой страны. Расстояние между абонентами региональной вычислительной сети — десятки-сотни километров.

Глобальные сети (WAN — Wide Area Network) объединяют пользователей, расположенных по всему миру. Взаимодействие между абонентами такой сети может осуществляться на базе телефонных линий связи, радиосвязи и систем спутниковой связи.

Объединение глобальных, региональных и локальных сетей позволяет создавать *многосетевые иерархии*. Они обеспечивают мощные, экономически целесообразные средства обработки огромных информационных массивов и доступ к неограниченным информационным ресурсам.

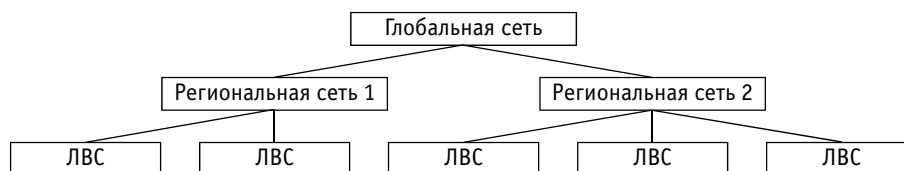


Рис.13. Иерархия компьютерных сетей

1.7.6. ХАРАКТЕРИСТИКИ КОММУНИКАЦИОННОЙ СЕТИ

Для оценки качества коммуникационной сети можно использовать такие характеристики, как:

- *скорость передачи данных по каналу связи*, которая измеряется количеством битов информации, передаваемой в секунду. Часто используется единица измерения скорости — бод. *Бод* — число изменений состояния среды передачи в секунду. Так как каждое изменение состояния может соответствовать нескольким битам данных, то реальная скорость в битах в секунду может превышать скорость в бодах;

- *пропускная способность канала связи* — количество знаков, передаваемых по каналу за единицу времени — секунду;

- *достоверность передачи информации* — отношение количества ошибочно переданных знаков к общему числу переданных знаков. Единица измерения достоверности: количество ошибок на знак — ошибок/знак. Для вычислительных сетей этот показатель должен лежать в пределах: одна ошибка на миллион или на десять миллионов переданных знаков;

- *надежность каналов связи и модемов* определяется либо долей времени исправного состояния в общем времени работы, либо средним временем безотказной работы. Единица измерения надежности: среднее время безотказной работы — час. Для вычислительных сетей среднее время безотказной работы должно быть достаточно большим и составлять, как минимум, несколько тысяч часов.

1.7.7. СЕТЕВОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Функциональные возможности сети определяются теми услугами, которые она предоставляет пользователю. Для реализации каждой из услуг сети и доступа пользователя к этой услуге разрабатывается программное обеспечение.

Программное обеспечение, предназначенное для работы в сети, должно быть ориентированным на одновременное использование многими пользователями. В настоящее время получили распространение две основные концепции построения такого программного обеспечения.

1. **Файл-сервер.** В этой концепции сетевое программное обеспечение ориентировано на предоставление многим пользователям ресурсов некоторого общедоступного главного компьютера сети, называемого файловым сервером. *Сервер* — компьютер, подключенный к сети и обеспечивающий ее пользователей определенными услугами. Это название он получил потому, что основным ресурсом главного компьютера являются файлы. Это могут быть файлы, содержащие программные модули или данные. Очевидно, емкость дисков файлового сервера должна быть больше, чем на обычном компьютере, так как он используется многими компьютерами. В сети может быть несколько файловых серверов.

Файловый сервер работает под управлением специальной операционной системы, которая обеспечивает одновременный доступ пользователей сети к расположенным на нем данным.

Сетевое программное обеспечение, управляющее ресурсами файлового сервера и предоставляющее к ним доступ многим пользователям сети, называется *сетевой операционной системой*.

Рабочая станция — персональный компьютер, подключенный к сети, через который пользователь получает доступ к ее ресурсам.

Основная часть сетевой операционной системы размещается на файловом сервере; на рабочих станциях устанавливается только небольшая оболочка, выполняющая роль интерфейса между программами, обращающимися за ресурсом, и файловым сервером. Вся обработка данных, даже если они являются общим ресурсом и хранятся на файловом сервере, производится на компьютере пользователя. Очевидно, что для этого файлы, в которых хранятся данные, должны быть перемещены на компьютер пользователя.

2. **Архитектура «клиент-сервер».** В этой концепции программное обеспечение ориентировано не только на коллективное использование ресурсов, но и на их обработку в месте размещения ресурса по запросу пользователей. Программные системы архитектуры клиент-сервер состоят из двух частей: программного обеспечения сервера и программного обеспечения пользователя-клиента. Работа этих систем организуется следующим образом: программы-клиенты выполняются на компьютере пользователя и посылают запросы к программе-серверу, которая работает на компьютере общего доступа. Основная обработка данных производится мощным сервером, а на компьютер пользователя посылаются только результаты выполнения запроса.

В приложениях глобальных сетей архитектура клиент-сервер является основной. Широко известны Web-серверы, обеспечивающие хранение и обработку гипертекстовых страниц, серверы электронной почты и др. Программы клиентов этих служб позволяют сформулировать запрос на получение услуги со стороны таких серверов и принять от них ответ.

1.7.8. ЛОКАЛЬНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

Основное назначение любой компьютерной сети — предоставление информационных и вычислительных ресурсов подключенным к ней пользователям.

С этой точки зрения локальную вычислительную сеть (ЛВС) можно рассматривать как совокупность серверов и рабочих станций.

Серверы могут осуществлять хранение данных, управление базами данных, удаленную обработку заданий, печать заданий и ряд других функций. Сервер — источник ресурсов сети.

Рабочая станция сети функционирует как в сетевом, так и в локальном режиме. Она оснащена собственной операционной системой (MS DOS, Windows и т.д.), обеспечивает пользователя всеми необходимыми инструментами для решения прикладных задач.

Как уже отмечалось выше, файл-сервер хранит данные пользователей сети и обеспечивает им доступ к этим данным. Файл-сервер выполняет следующие функции: хранение данных, архивирование данных, синхронизацию изменений данных различными пользователями, передачу данных.

Компьютерные сети реализуют распределенную обработку данных: обработка данных в этом случае распределена между двумя объектами — клиентом и сервером.

Архитектура клиент-сервер может использоваться в одноранговых локальных вычислительных сетях и в сетях с выделенным сервером.

Одноранговая сеть. В такой сети нет единого центра управления взаимодействием рабочих станций и нет единого устройства хранения данных. Сетевая операционная система распределена

по всем рабочим станциям. Каждая станция может выполнять функции как клиента, так и сервера. Она может обслуживать запросы от других рабочих станций и направлять свои запросы на обслуживание в сеть.

Пользователю доступны все устройства, подключенные к другим станциям (диски, принтеры).

Достоинства одноранговых сетей:

- низкая стоимость;
- высокая надежность.

Недостатки одноранговых сетей:

- зависимость эффективности работы сети от количества станций;
- сложность управления сетью;
- сложность обеспечения защиты информации;
- трудности обновления и изменения программного обеспечения станций.

Сеть с выделенным сервером. В сети с выделенным сервером один из компьютеров (его обычно называют сервером сети) выполняет функции хранения данных, предназначенных для использования всеми рабочими станциями, управления взаимодействием между рабочими станциями и ряд сервисных функций. В сетях с централизованным управлением существует возможность обмена информацией между рабочими станциями, минуя файл-сервер.

Достоинства сети с выделенным сервером:

- надежная система защиты информации;
- высокое быстродействие;
- отсутствие ограничений на число рабочих станций;
- простота управления по сравнению с одноранговыми сетями.

Недостатки сети:

- высокая стоимость из-за выделения одного компьютера под сервер;
- зависимость быстродействия и надежности сети от сервера;
- меньшая гибкость по сравнению с одноранговой сетью.

Сети с выделенным сервером являются наиболее распространенными у пользователей компьютерных сетей. Сетевые операционные системы для таких сетей — LANServer (IBM), Windows NT Server версий 3.51 и 4.0 и NetWare (Novell).

Основные топологии ЛВС. Вычислительные машины, входящие в состав ЛВС, могут быть расположены на территории, где создается вычислительная сеть случайным образом. Однако для способа обращения к передающей среде и методов управления сетью безразлично, как расположены абонентские ЭВМ.

Топология ЛВС — это усредненная геометрическая схема соединений узлов сети.

Любую компьютерную сеть можно рассматривать как совокупность узлов.

Узел — любое устройство, непосредственно подключенное к передающей среде.

Топологии могут быть разными, но типичными являются кольцевая, шинная и звездообразная.

Кольцевая топология предусматривает соединение узлов сети замкнутой кривой — кабелем передающей среды. Информация по кольцу передается от узла к узлу. Кольцевая топология идеальна для сетей, занимающих сравнительно небольшое пространство. В ней отсутствует центральный узел, что повышает надежность сети. Но последовательное соединение узлов сети снижает ее быстродействие, а выход из строя одного из узлов нарушает целостность кольца, т.е. требуются специальные меры для сохранения тракта передачи информации.

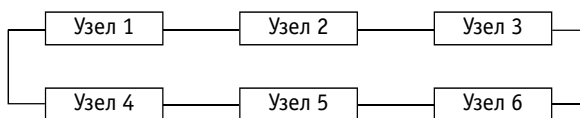


Рис.14. Сеть кольцевой топологии

В *шинной топологии* данные от передающего узла сети распространяются по шине в обе стороны. Информация поступает на все узлы, принимает сообщение лишь тот узел, кому она адресована. Дисциплина обслуживания параллельная, что обеспечивает высокое быстродействие сети. Сеть легко наращивать и конфигурировать. Это наиболее распространенная в настоящее время топология.

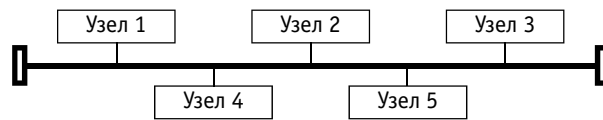


Рис.15. Сеть шинной топологии

В *звездообразной топологии* вся информация передается через центральный узел, который ретранслирует, переключает и маршрутизирует информационные потоки в сети. Такая топология значительно упрощает взаимодействие узлов друг с другом, позволяет использовать более простые сетевые адаптеры. Но работоспособность такой сети целиком зависит от центрального узла.

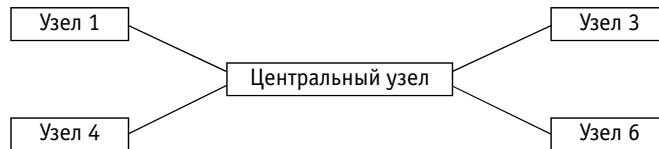


Рис.16. Сеть звездообразной топологии

Созданная на определенном этапе развития системы локальная вычислительная сеть с течением времени перестает удовлетворять потребности всех пользователей, и тогда встает проблема расширения ее функциональных возможностей. Может возникнуть необходимость объединения внутри фирмы различных локальных вычислительных сетей, появившихся в различных ее отделах и филиалах в разное время, хотя бы для организации обмена данными с другими системами. Проблема расширения конфигурации сети может быть решена как в пределах ограниченного пространства, так и с выходом во внешнюю среду.

Стремление получить выход на определенные информационные ресурсы может потребовать подключения локальных вычислительных сетей к сетям более высокого уровня.

В самом простом варианте объединение локальных вычислительных сетей необходимо для расширения сети в целом, но технические возможности существующей сети исчерпаны, новых абонентов подключить к ней нельзя. Можно только создать еще одну ЛВС и объединить ее с уже существующей.

1.7.9. ГЛОБАЛЬНАЯ СЕТЬ ИНТЕРНЕТ

Интернет (Internet) представляет собой глобальную компьютерную сеть. Само ее название означает «между сетей». Это сеть, объединяющая отдельные сети.

Интернет обеспечивает обмен информацией между всеми компьютерами, которые входят в сеть, подключенные к ней. Тип компьютера и используемая им операционная система значения не имеют. Соединение сетей обладает громадными возможностями.

Основные ячейки Интернет — локальные вычислительные сети. Если некоторая локальная вычислительная сеть непосредственно подключена к Интернет, то каждая рабочая станция такой сети также может подключаться к Интернет. Существуют также компьютеры, самостоятельно подключенные к Интернет. Они называются хост-компьютерами. Каждый подключенный к сети компьютер имеет свой адрес, по которому его может найти абонент из любой точки света.

Важной особенностью Интернет является то, что она, объединяя различные сети, не создает при этом никакой иерархии — все компьютеры, подключенные к сети равноправны. Для иллюстрации возможной структуры некоторого участка сети Интернет на рис.17 приведена схема соединения различных сетей.

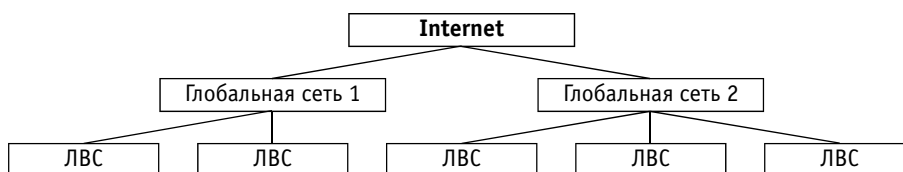


Рис.17. Подключение различных сетей к Интернет

1.7.9.1. Система адресации в Интернет

Интернет самостоятельно осуществляет передачу данных. К адресам станций предъявляются специальные требования. Адрес должен иметь формат, позволяющий вести его обработку автоматически, и должен нести некоторую информацию о своем владельце.

С этой целью для каждого компьютера устанавливаются два адреса: цифровой IP-адрес (IP — Internetnetwork Protocol — межсетевой протокол) и доменный адрес.

Оба эти адреса могут применяться равноценно. Цифровой адрес удобен для обработки на компьютере, а доменный адрес — для восприятия пользователем.

Цифровой адрес имеет длину 32 бита и содержит полную информацию, необходимую для идентификации компьютера. Он разделяется на 4 блока по 8 бит: два блока определяют адрес в сети, а два другие — адрес компьютера внутри этой сети. Цифровой адрес (IP-адрес) включает в себя три компонента: адрес сети, адрес подсети, адрес компьютера в подсети. Например, десятичный код адреса имеет вид: 192.45.9.200. Адрес сети — 192.45; адрес подсети — 9; адрес компьютера — 200.

Доменный адрес определяет область, представляющую ряд хост-компьютеров. В отличие от цифрового адреса он читается в обратном порядке.

Чтобы абонентам Интернет можно было достаточно просто связаться друг с другом, все пространство ее адресов разделяется на области — домены. Возможно также разделение по определенным признакам внутри доменов.

В системе адресов Интернет приняты домены, представленные географическими регионами. Они имеют имя, состоящее из двух букв.

Существуют и домены, разделенные по тематическим признакам. Такие домены имеют трехбуквенное сокращенное название.

Компьютерное имя включает как минимум два уровня доменов. Каждый уровень отделяется от другого точкой. Слева от домена верхнего уровня располагаются другие имена. Все имена, находящиеся слева, — поддомены для общего домена.

Для пользователей Интернет адресами могут быть просто их регистрационные имена на компьютере, подключенном к сети.

Для обработки пути поиска в доменах имеются специальные серверы имен. Они преобразуют доменное имя в соответствующий цифровой адрес.

Локальный сервер передает запрос на глобальный сервер, имеющий связь с другими локальными серверами имен.

Например, victor@tutor.sptu.edu — имя пользователя victor, зарегистрировано на компьютере, имеющем в Интернет имя tutor.sptu.edu.

1.7.9.2. Способы организации передачи информации в Интернет

Электронная почта (e-mail — electronic mail) выполняет функции обычной почты. Электронное письмо приходит сразу же после его отправления и хранится в почтовом ящике до получения адресатом. Кроме текста оно может содержать графические и звуковые файлы. Электронные письма могут отправляться сразу по нескольким адресам.

Для работы с электронной почтой создано большое количество программ.

Эти программы выполняют следующие функции:

- подготовку текста;
- чтение и сохранение корреспонденции;
- удаление корреспонденции;
- ввод адреса;
- комментирование и пересылку корреспонденции;
- импорт (прием и преобразование в нужный формат) других файлов.

Сообщения можно обрабатывать собственным текстовым редактором программы электронной почты.

Формат адреса электронной почты должен иметь вид:

имя пользователя@адрес хост-компьютера.

1.7.9.3. WORLD-WIDE-WEB (Всемирная информационная паутина)

WWW является одной из самых популярных информационных служб Интернет. Две основные особенности отличают WWW: использование гипертекста и возможность клиентов взаимодействовать с другими приложениями Интернет.

Гипертекст — текст, содержащий в себе связи с другими текстами, графической, видео- или звуковой информацией.

Внутри гипертекстового документа некоторые фрагменты текста четко выделены. Указание на них с помощью, например, мыши позволяет перейти на другую часть этого же документа, на другой документ в этом же компьютере или даже на документы на любом другом компьютере, подключенном к Интернет.

Все серверы WWW используют специальный язык HTML (Hypertext Markup Language — язык разметки гипертекста). HTML-документы представляют собой текстовые файлы, в которые встроены специальные команды.

Отображенный на экране гипертекст представляет собой сочетание алфавитно-цифровой информации в различных форматах и стилях и некоторые графические изображения — картинки.

Связь между гипертекстовыми документами осуществляется с помощью ключевых слов. Найдя ключевое слово, пользователь может перейти в другой документ, чтобы получить дополнительную информацию.

1.7.9.4. Телеконференции Usenet

Система Usenet была разработана для перемещения новостей между компьютерами по всему миру. В дальнейшем она практически полностью интегрировалась в Интернет, и теперь Интернет обеспечивает распространение всех ее сообщений. Серверы Usenet имеют средства для разделения телеконференций по темам.

Телеконференции — дискуссионные группы, входящие в состав Usenet.

Телеконференции организованы по иерархическому принципу, и для верхнего уровня выбраны семь основных рубрик. В свою очередь, каждая из них охватывает сотни подгрупп. Образуется древовидная структура, напоминающая организацию файловой системы.

Из числа основных рубрик следует выделить:

- comp — компьютерные темы;
- sci — научные темы;
- news — информация и новости Usenet;
- soc — социальная тематика;
- talk — дискуссии.

Существуют, кроме того, специальные рубрики и региональное разделение телеконференций.

Управляют доступом к службе Usenet специальные программы, позволяющие выбирать телеконференции, работать с цепочками сообщений и читать сообщения и ответы на них.

При участии в какой-либо телеконференции любой абонент может направить свое сообщение по интересующей его теме.

1.7.9.5. Передача файлов с помощью протокола FTP

Для того чтобы обеспечить перемещение данных между различными операционными системами, которые могут встретиться в Интернет, используется протокол FTP (File Transfer Protocol), работающий независимо от применяемого оборудования. Протокол обеспечивает способ перемещения файлов между двумя компьютерами и позволяет абоненту сети Интернет получить в свое распоряжение множество файлов.

Программа, реализующая этот протокол, позволяет установить связь с одним из множества FTP-серверов в Интернет.

FTP-сервер — компьютер, на котором содержатся файлы, предназначенные для открытого доступа.

Программа FTP-клиент не только реализует протокол передачи данных, но и поддерживает набор команд, которые используются для просмотра каталога FTP-сервера, поиска файлов и управления перемещением данных.

Операционная система Windows 95 позволяет работать с программой WS_FTP, что обеспечивает удобный способ работы с серверами FTP. Еще один способ работы основан на использовании приложений-навигаторов WWW, таких как Microsoft Internet Explorer, Netscape Navigator.

Взаимодействие с другим компьютером (Telnet). Telnet обеспечивает взаимодействие с удаленным компьютером. Установив такую связь через Telnet, пользователь получает возможность работать с удаленным компьютером как со «своим», т.е. теоретически получить в свое распоряжение все ресурсы, если к ним разрешен доступ.

Telnet — это программа, которая позволяет использовать мощь Интернет для прямого соединения пользователя непосредственно в момент работы с базами данных, библиотечными каталогами и другими информационными ресурсами по всему миру.

Два вида услуг Интернет требуют подключения к серверам через Telnet: библиотечные каталоги и электронные доски объявлений.

Как любое большое сообщество, компьютерный мир имеет свои библиотеки, куда вы можете пойти для поиска информации или чтобы взять хорошую книгу. Telnet это один из ваших ключей к этим библиотекам. Несколько сотен библиотек по всему миру доступны через Интернет.

Независимо от Интернет существуют маленькие диалоговые службы, предоставляющие доступ к BBS (Bulletin Board System — система электронных досок объявлений).

Это компьютеры, к которым можно присоединиться с помощью модемов через телефонную сеть. BBS содержит файлы, которые можно переписывать, они позволяют проводить дискуссии, участвовать в различных играх и имеют свою систему электронной почты.

1.7.9.6. Глобальные компьютерные сети в финансово-экономической деятельности

Конкурентоспособная экономика базируется на системе финансовых организаций, способных предоставить услуги всем потенциальным клиентам. Глобальные сети, связывающие отдельные автоматизированные комплексы организаций, позволяют проводить расчеты на национальном и международном уровне. Развитие телекоммуникационных услуг является одним из ключевых моментов оживления хозяйственной и коммерческой деятельности.

Какие возможности дает Интернет современному экономисту, управленцу?

- Возможность оперативно получить информацию от непосредственного производителя товара или услуг, минуя посредников.
- Найти новых партнеров по бизнесу.
- Поместить свою рекламу на самых популярных серверах всего мира.
- Заключить контракт с любым партнером или клиентом, находящимся в любой части света.
- Оперативно узнавать о котировке валют и курсе акций любых фирм на самых известных биржах всего мира.
- Следить за изменениями в законодательстве, что поможет своевременно принимать правильные решения.
- В банковской системе глобальные сети используются для проведения различных финансовых операций, используя такие преимущества систем электронных сообщений, как скорость совершения операций и доступность информационных ресурсов. Во всем мире создаются сети, предлагающие услуги по межбанковским операциям и обеспечивающие высокий уровень защиты информации.
- Эффективно работать на мировом фондовом рынке можно, только обладая определенным уровнем компьютеризации и выходами на телекоммуникации. Исключительно большое значение на рынке ценных бумаг придается электронному формированию и поддержке единого информационного пространства участников рынка. Появляются организации, специализирующиеся на координации и систематизации разнородных потоков информации.
- Электронная коммерция подразумевает использование технологий глобальных компьютерных сетей для ведения бизнеса. В настоящее время сеть Интернет превратилась в популярную среду для общения, рекламы, бизнеса. Именно популярность и доступность Интернета сделала возможным широкое использование электронной коммерции. Потребители могут искать, заказывать и оплачивать товары, используя Интернет, обмениваться информацией о товарах и

услугах с другими пользователями. Правительственные организации могут использовать Интернет для сбора налоговых деклараций и распространения официальной информации.

Можно выделить два аспекта электронной коммерции:

- электронная коммерция как торговля в сети Интернет подразумевает продажу товаров и услуг с использованием Интернет;
- электронная коммерция как рынок. То есть виртуальный магазин может не только предлагать свою продукцию пользователям Интернета, но и искать поставщиков продукции, заключать контракты с ними, оплачивать счета, нанимать сотрудников и т.д.

Интернет широко используют как средство электронных платежных средств. Существует три типа электронных платежных средств:

- комбинация обычных и электронных платежей. Например, платеж осуществляется традиционным способом, подтверждение владельцу высылается по электронной почте или наоборот;
- передача номеров кредитных карточек электронным способом и использование так называемых смарт-карт (небольшие устройства, которые могут хранить электронную наличность, электронные ключи и др.), которые могут хранить и передавать информацию о своем владельце. Вся операция может происходить в электронном виде;
- различные виды цифровой наличности и электронных денег — если сообщение, например, само несет в себе некую сумму денег, а не только номер кредитной карточки.

Контрольные вопросы

1. Что такое сжатие (упаковка) информации?
2. Чем архивный файл отличается от обычного?
3. Что такое многотомные архивы?
4. Что такое компьютерный вирус?
5. В чем состоят проявления компьютерных вирусов?
6. Какие существуют виды программ для обнаружения и защиты от вирусов?
7. Назовите основные меры по защите от компьютерных вирусов.
8. Что такое распределенная обработка данных?
9. Что такое компьютерная (вычислительная) сеть?
10. В чем основное назначение компьютерной сети?
11. Как вы понимаете, что такое коммуникационная сеть? Какие характеристики коммуникационной сети вы можете назвать?
12. Что такое локальная, региональная, глобальная вычислительная сеть?
13. Назовите основные элементы компьютерной сети.
14. В чем заключается концепция файл-сервера? Какие требования предъявляются к файл-серверу, рабочим станциям?
15. Что такое технология клиент-сервер?
16. Что такое одноранговая сеть?
17. Что такое топология сети? Какие топологии вы знаете?
18. Почему возникает проблема объединения компьютерных сетей?
19. Что такое глобальная компьютерная сеть Internet?
20. Какие возможности предоставляет Internet?
21. Что такое протокол?
22. Система адресации в Интернет.
23. Глобальные компьютерные сети в финансово-экономической деятельности.

Литература

1. *Евдокимов В.В. и др.* Экономическая информатика: Учебник для вузов. — СПб.: Питер, 1997. — С.218–310.
2. Информатика: Базовый курс / С.В.Симонович и др. — СПб.: Питер, 2001. — С.198–243, 364–387.
3. Информатика: Учебник / Под ред. проф. Н.В.Макаровой. — М.: Финансы и статистика, 1998. — С.202–242, 418–435, 447–455.
4. *Фигурнов В.Э.* IBM PC для пользователя. — М.: ИНФРА-М, 1995. — С.325–370.
5. Экономическая информатика / Под ред. П.В.Конюховского и Д.Н.Колесова. — СПб.: Питер, 2000. — С.132–142, 394–410, 436–474.

РАЗДЕЛ 2

ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ ПРОДУКТОВ

ТЕМА 2.1. АЛГОРИТМИЗАЦИЯ

Появление первых компьютеров породило программирование как науку. Разрабатывались математические теории обработки информации, средства доказательства правильности программ и т.п. В первые годы программированием занимались профессиональные программисты. С появлением новых языков программирования миллионы людей, не имеющие специального образования, получили возможности применять компьютеры для решения своих прикладных задач.

2.1.1. ЭТАПЫ СОЗДАНИЯ ПРОГРАММНОГО ПРОДУКТА

Программы любого вида характеризуются жизненным циклом — от момента возникновения идеи разработки программы до момента отказа от использования программы. Стадии жизненного цикла программ определяют состав и содержание работ по созданию программных продуктов.

При традиционной разработке программ различают следующие этапы создания программного продукта.

1. **Составление технического задания на разработку программы.** Исходными данными для выполнения работ этого этапа являются постановка задачи и анализ рынка программных продуктов.

На этом этапе требуется:

- разработать обобщенный алгоритм (функциональную структуру алгоритма или состав объектов);
- выбрать платформу программы — тип операционной системы;
- определить необходимость разработки программы, которую можно переносить на другие платформы;
- оценить необходимость сетевого варианта работы программы;
- определить требования к техническим средствам обработки информации;
- специфицировать требования к пользовательскому интерфейсу.

2. **Разработка технического проекта.** На данном этапе выполняется комплекс наиболее важных работ:

- разрабатывается детальный алгоритм решения комплекса задач с учетом принятых решений по составу и структуре данных или уточняется состав объектов и их свойств, методов обработки, событий;
- определяется детализированный состав общего программного обеспечения (операционная система, тип и архитектура сети, модель СУБД, текстовый процессор, табличный процессор, методо-ориентированные и функциональные ППП и т.д.);
- разрабатывается модульная структура алгоритма и детализированные алгоритмы отдельных модулей;
- выбираются инструментальные средства разработки программ.

Работы данного этапа зависят от выбранных инструментальных средств создания программ, технологии работ.

3. **Рабочий проект (рабочая документация).** Данный этап является технической реализацией проектных решений. Основным содержанием работ является разработка программных модулей или методов обработки объектов (собственно программирование), их автономная и комплексная отладка на контрольном примере, испытание работоспособности программного продукта в комплексе с системными и базовыми программными средствами.

Основной результат этого этапа — создание комплекта документации на программный продукт:

- *описание применения*, где дается общая характеристика программного продукта и его сферы применения, требования к базовому программному обеспечению, комплексу технических средств обработки;

- *руководство пользователя* — детальное описание функциональных возможностей и технологии работы с программным продуктом для конечного пользователя (содержит необходимую информацию для самостоятельного освоения и нормальной работы пользователя);
- *руководство программиста (оператора)* — описание особенностей установки (инсталляции) программного продукта и внутренней структуры программного продукта (состав и назначение модулей, правила эксплуатации и обеспечения надежной и качественной работы программного продукта). На данном этапе для программных продуктов создаются, если необходимо, обучающие системы, демоверсии, гипертекстовые системы помощи.

4. **Ввод в действие.** Готовый программный продукт проходит опытную эксплуатацию, а затем сдается в промышленную эксплуатацию. Выведение программного продукта на рынок программных средств заключается в организации продажи программ массовому пользователю, а для заказанного программного продукта — в переходе к эксплуатации.

2.1.2. СТРУКТУРА ПРОГРАММНЫХ ПРОДУКТОВ

Программные продукты имеют внутреннюю структуру (внутреннюю организацию), образованную взаимосвязанными программными модулями.

Модуль — это самостоятельная часть программы, имеющая определенное назначение и обеспечивающая заданные функции обработки независимо от других программных модулей.

Структуризация программ выполняется для удобства разработки, программирования, отладки и внесения изменений в программный продукт. Особенно это важно, когда программный продукт разрабатывается коллективом разработчиков.

Структуризация программных продуктов позволяет:

- распределить работы по исполнителям, обеспечив их загрузку и требуемые сроки разработки программных продуктов;
- построить календарные графики проектных работ и осуществлять их координацию в процессе создания программных продуктов;
- контролировать трудозатраты и стоимость проектных работ и др.

Среди множества модулей различают:

- *головной модуль*, который управляет запуском программного продукта (существует в единственном числе);
- *управляющие модули*, которые задают последовательность вызова и обеспечивают вызов других модулей на обработку;
- *рабочие модули*, выполняющие функции обработки;
- *сервисные модули и библиотеки*, утилиты, осуществляющие обслуживающие функции.

Некоторые программные продукты используют модули из готовых библиотек стандартных подпрограмм, процедур, функций, объектов, методов обработки данных.

Информационная связь модулей обеспечивается за счет использования общей базы данных либо межмодульной передачи данных через переменные обмена.

Каждый модуль может оформляться как самостоятельно хранимый файл и для функционирования программного продукта необходимо наличие программных модулей в полном составе.

Структурно-сложные программные продукты разрабатываются как пакеты программ. А так как они чаще всего имеют прикладной характер, то и называются пакеты прикладных программ, или ППП.

ППП — это система программ, предназначенных для решения задач определенного класса.

Компоненты ППП объединены общими данными (базой данных), информационно и функционально связаны между собой и обладают свойством системности, т.е. объединению программ присуще новое качество, которое отсутствует для отдельного компонента ППП.

2.1.3. ВИДЫ ПРОЕКТИРОВАНИЯ ПРОГРАММНЫХ ПРОДУКТОВ

Проектирование алгоритмов и программ может основываться на различных подходах, среди которых наиболее распространены:

- структурное проектирование программных продуктов;
- информационное моделирование предметной области и связанных с ней приложений;
- объектно-ориентированное проектирование программных продуктов.

В основе **структурного проектирования** лежит последовательная декомпозиция, целенаправленное структурирование на отдельные составляющие.

Алгоритмы большой сложности обычно представляют с помощью схем двух видов:

- *обобщенной*, раскрывающей общий принцип функционирования алгоритма и основные логические связи между отдельными этапами на уровне типовых процессов обработки информации (ввод и редактирование данных, вычисления, печать результатов и т.п.);
- *детальной*, представляющей содержание каждого элемента обобщенной схемы с использованием управляющих структур в виде блок-схем алгоритма, псевдокода либо алгоритма на языках высокого уровня (программы).

Представителями структурного проектирования являются следующие методы:

- нисходящее проектирование;
- модульное программирование;
- структурное программирование.

Метод нисходящего проектирования предполагает последовательное разложение общей функции обработки данных на простые функциональные элементы методом «сверху — вниз», от целого к части.

В результате строится иерархическая схема, отражающая состав и взаимоподчиненность отдельных функций, которая носит название *функциональная структура алгоритма* (ФСА) приложения. Каждый отдельный элемент ФСА описывает законченную содержательную функцию обработки информации, которая предполагает определенный способ реализации на программном уровне. Функции ввода-вывода информации рекомендуется отделять от функций вычисления или логической обработки данных.

Подобная структура приложения отражает наиболее важное — состав и взаимосвязь функций обработки информации для реализации приложений, хотя и не раскрывает логику выполнения каждой отдельной функции, условия или периодичность их вызовов.

Модульное программирование основано на понятии модуля — логически взаимосвязанной совокупности функциональных элементов, оформленных в виде отдельных программ. Принципы модульного программирования программных продуктов во многом сходны с принципами нисходящего проектирования. Сначала определяются состав и подчиненность функций, а затем — набор программных модулей, реализующих эти функции. Однотипные функции реализуются одними и теми же модулями. Функция верхнего уровня обеспечивается главным модулем, который управляет выполнением нижестоящих функций, реализуемых подчиненными модулями.

В результате создается *функционально-модульная схема* (ФМС) алгоритма приложения, которая является основой для программирования.

Структурное программирование основано на модульной структуре программного продукта и типовых управляющих структурах алгоритмов обработки данных различных программных модулей. В любой типовой структуре блок, кроме условного (который имеет два выхода), имеет только один вход и один выход. Виды основных управляющих структур — последовательность, альтернатива (условие выбора), цикл.

Информационное моделирование предметной области — подход, в основе которого положение об определяющей роли данных при проектировании алгоритмов и программ. Этот подход имеет решающее значение для разработки алгоритмов и программ, работающих с базами данных.

В этом подходе выделяют следующие составляющие:

- информационный анализ предметных областей (бизнес-областей);
- информационное моделирование — построение комплекса взаимосвязанных моделей данных;
- системное проектирование функций обработки данных;
- детальное конструирование процедур обработки данных.

Первоначально строятся информационные модели: модели, отражающие интегрированные структуры данных предметной области и не зависящие от средств программной реализации хранения и обработки данных (информационно-логическая модель) и модели, ориентированные на среду хранения и обработки данных (даталогические модели).

Технологии, ориентированные на информационное моделирование, сначала специфицируют данные, а затем описывают процессы, использующие эти данные. Средствами структур данных моделируются функции предметной области, прослеживается взаимосвязь функций обработки данных, уточняется состав входной и выходной информации, логика преобразования входных структур данных в выходные.

Метод объектно-ориентированного проектирования основывается на:

- модели построения системы как совокупности объектов абстрактного типа данных;
- модульной структуре программ;
- нисходящем проектировании, используемом при выделении объектов.

В отличие от традиционного структурного подхода, объектно-ориентированный подход к проектированию программных продуктов основан на:

- выделении классов;
- установлении характерных свойств классов и методов их обработки;
- создании иерархии классов;
- наследовании свойств классов и методов их обработки.

Каждый объект объединяет в себе как данные, так и программу обработки этих данных. Программный продукт, созданный с помощью инструментальных средств объектно-ориентированного программирования, содержит объекты с их характерными свойствами, для которых разработан графический интерфейс пользователя.

2.1.4. ПОНЯТИЕ АЛГОРИТМА

Решение задачи при помощи ЭВМ начинается с составления алгоритма. Что же такое алгоритм?

Происхождение термина «алгоритм» связывают с именем великого математика Мухаммеда аль-Хорезми (763–850 гг.), который разработал правила выполнения четырех арифметических действий. Первоначально алгоритмами называли правила, определяющие последовательность действий для получения суммы чисел, произведения и т.д. И многие годы понятие «алгоритм» использовалось математиками для описания правил решения математических задач.

Однако не следует считать алгоритм чисто математическим понятием. Под алгоритмом понимают совокупность правил выполнения определенных действий, обеспечивающих решение задачи (например, инструкции пользования бытовыми приборами, кулинарный рецепт и др.).

Согласно ГОСТ 19781-74: **Алгоритм** — это точное предписание, определяющее вычислительный процесс, ведущий от варьируемых начальных данных к искомому результату.

То есть алгоритм — это четкое указание исполнителю алгоритма выполнить определенную последовательность действий для решения поставленной задачи и получения результата. При этом под исполнителем алгоритма понимают тот объект, для управления действиями которого составлен алгоритм, например, человек, компьютер и т.д. При составлении алгоритма нужно учитывать возможности исполнителя, который умеет выполнять некоторый, вполне определенный конечный набор действий, поэтому алгоритм формулируется в расчете на конкретного исполнителя.

Разработать алгоритм означает разбить задачу на определенную последовательность шагов. От разработчика алгоритма требуется знание особенностей и правил составления алгоритмов.

Основные особенности алгоритмов:

1. Наличие *ввода* исходных данных.
2. Наличие *вывода* результата выполнения алгоритма, поскольку цель выполнения алгоритма — получение результата, имеющего вполне определенное отношение к исходным данным.
3. Алгоритм должен иметь *дискретную структуру*, т.е. алгоритм представляется в виде последовательности шагов, и выполнение каждого очередного шага начинается после завершения предыдущего.
4. *Однозначность* — каждый шаг алгоритма должен быть четко определен и не должен допускать произвольной трактовки исполнителем.
5. *Конечность* — исполнение алгоритма должно закончиться за конечное число шагов.
6. *Корректность* — алгоритм должен задавать правильное решение задачи.
7. *Массовость* (общность) — алгоритм разрабатывается для решения некоторого класса задач, различающихся исходными данными.

8. *Эффективность* — алгоритм должен выполняться за разумное конечное время. При этом выбирается наиболее простой и короткий способ решения задачи при соблюдении, естественно, всех ограничений и требований к алгоритму.

2.1.4.1. Способы записи алгоритмов

Разработанный алгоритм может быть представлен несколькими способами:

- 1) на естественном языке (словесная запись алгоритма);
- 2) в виде блок-схем (графическая форма);
- 3) на языке программирования.

Словесная запись алгоритма. Словесная форма используется обычно для описания алгоритмов, предназначенных исполнителю — человеку. Команды записываются на обычном языке и выполняются по порядку. В командах могут использоваться формулы, специальные обозначения, но каждая команда должна быть понятна исполнителю. Естественный порядок команд может быть нарушен (если требуется, например, переход к предыдущей команде или требуется обойти очередную команду при каком-то условии), в этом случае команды можно нумеровать и указывать команду, к которой требуется перейти. Например, *перейти к п.3* или *повторить с п.4*.

Графическая форма. Алгоритмы представляются в виде блок-схем. Существуют специальные стандарты для построения блок-схем, где определяются графические изображения блоков. Команды алгоритмов записываются внутри блоков на обычном языке или с использованием математических формул. Блоки соединяются по определенным правилам линиями связи, которые показывают порядок выполнения команд.

На языке программирования. Если алгоритм разработан для решения задачи на ЭВМ, то для того, чтобы он мог выполняться исполнителем — ЭВМ, его необходимо записать на языке, понятном этому исполнителю. Для этого разработано множество языков программирования для решения задач разных классов. Запись алгоритма на языке программирования называется программой.

2.1.4.2. Представление алгоритмов в виде блок-схем

Блок-схемой будем называть такое графическое представление алгоритма, когда отдельные действия (или команды) представляются в виде геометрических фигур — *блоков*. Внутри блоков указывается информация о действиях, подлежащих выполнению. Связь между блоками изображают с помощью линий, называемых *линиями связи*, обозначающих передачу управления.

Существует Государственный стандарт, определяющий правила создания блок-схем. Конфигурация блоков, а также порядок графического оформления блок-схем регламентированы ГОСТ 19.701-90 «Схемы алгоритмов и программ». В табл. 2.1 приведены обозначения некоторых элементов, которых будет вполне достаточно для изображения алгоритмов при выполнении студенческих работ.

Правила составления блок-схем:




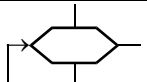


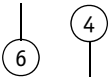


1. Каждая блок-схема должна иметь блок «Начало» и один блок «Конец».
2. «Начало» должно быть соединено с блоком «Конец» линиями потока по каждой из имеющихся на блок-схеме ветвей.
3. В блок-схеме не должно быть блоков, кроме блока «Конец», из которых не выходит линия потока, равно как и блоков, из которых управление передается «в никуда».
4. Блоки должны быть пронумерованы. *Нумерация* блоков осуществляется сверху вниз и слева направо, номер блока ставится вверху слева, в разрыве его начертания.
5. Блоки связываются между собой линиями потока, определяющими последовательность выполнения блоков. Линии потоков должны идти параллельно границам листа. *Если линии идут справа налево или снизу вверх, то стрелки в конце линии обязательны*, в противном случае их можно не ставить.
6. По отношению к блокам линии могут быть *входящими* и *выходящими*. Одна и та же линия потока является выходящей для одного блока и входящей для другого.
7. От блока «Начало» в отличие от всех остальных блоков линия потока только выходит, так как этот блок — первый в блок-схеме.

8. Блок «Конец» имеет только вход, так как это последний блок в блок-схеме.

9. Для простоты чтения желательно, чтобы линия потока входила в блок «Процесс» сверху, а выходила снизу.

10. Чтобы не загромождать блок-схему сложными пересекающимися линиями, линии потока можно разрывать. При этом в месте разрыва ставятся *соединители*, внутри которых указываются номера соединяемых блоков. В блок-схеме не должно быть разрывов, не помеченных соединителями.

11. Чтобы не загромождать блок, можно информацию о данных, об обозначениях переменных и т.п. размещать в *комментариях* к блоку.

Название блока	Обозначение блока	Назначение блока
Терминатор		Начало/Конец программы или подпрограммы
Процесс		Обработка данных (вычислительное действие или последовательность вычислительных действий)
Решение		Ветвление, выбор, проверка условия. В блоке указывается условие или вопрос, который определяет дальнейшее направление выполнения алгоритма
Подготовка		Заголовок счетного цикла
Предопределенный процесс		Обращение к процедуре
Данные		Ввод/Вывод данных
Соединитель		Маркировка разрыва линии потока
Комментарий		Используется для размещения пояснений к действиям
Горизонтальные и вертикальные потоки		Линии связей между блоками, направление потоков

2.1.4.3. Типы алгоритмов

Тип алгоритма определяется характером решаемой в соответствии с его командами задачи. Различают три типа алгоритмов: линейные, разветвляющиеся, циклические.

Линейный алгоритм состоит из упорядоченной последовательности действий, не зависящей от значений исходных данных, при этом каждая команда выполняется только один раз строго после той команды, которая ей предшествует.

Таким, например, является алгоритм вычисления по простейшим безальтернативным формулам, не имеющий ограничений на значения входящих в эти формулы переменных. Как правило, линейные процессы являются составной частью более сложного алгоритма.



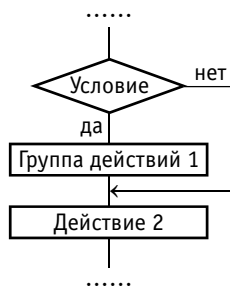
Разветвляющимися называются алгоритмы, в которых в зависимости от значения какого-то выражения или от выполнения некоторого логического условия дальнейшие действия могут производиться по одному из нескольких направлений.

Каждое из возможных направлений дальнейших действий называется *ветвью*.

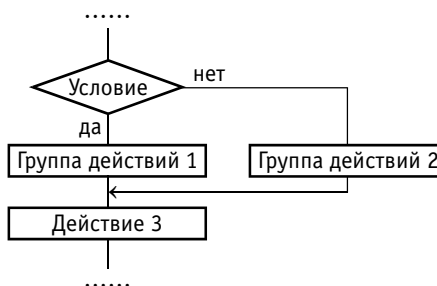
В блок-схемах разветвление реализуется специальным блоком «Решение». Этот блок предусматривает возможность двух выходов. В самом блоке «Решение» записывается логическое условие, от выполнения которого зависят дальнейшие действия.

Различают несколько видов разветвляющихся алгоритмов.

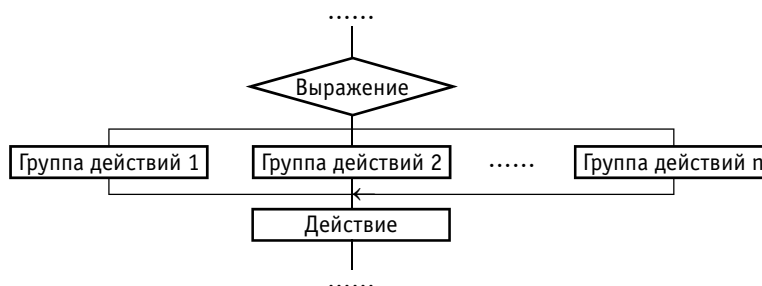
1. «Обход» — такое разветвление, когда одна из ветвей не содержит ни одного оператора, т.е. как бы обходит несколько действий другой ветви:



2. «Разветвление» — такой тип разветвления, когда в каждой из ветвей содержится некоторый набор действий:



3. «Множественный выбор» — особый тип разветвления, когда каждая из нескольких ветвей содержит некоторый набор действий. Выбор направления зависит от значения некоторого выражения:



Циклические алгоритмы применяются в тех случаях, когда требуется реализовать многократно повторяющиеся однотипные вычисления. *Цикл* — это последовательность действий, которая может выполняться многократно, т.е. более одного раза.

Различают:

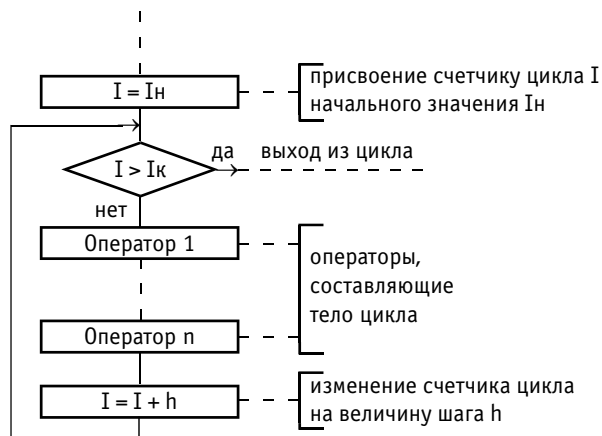
- циклы с известным числом повторений (или со счетчиком);
- циклы с неизвестным числом повторений (циклы с предусловием и циклы с постусловием).

В любом цикле должна быть переменная, которая управляет выходом из цикла, т.е. определяет число повторений цикла.

Последовательность действий, которая должна выполняться на каждом шаге цикла (т.е. при каждом повторении цикла), называется *телом цикла* или *рабочей частью цикла*.

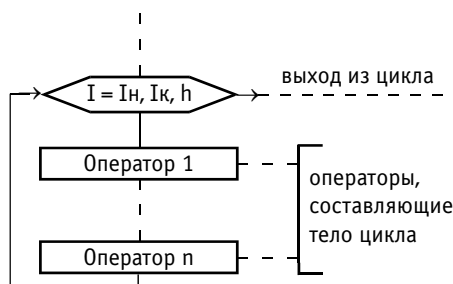
Циклы со счетчиком. В циклах такого типа известно число повторений цикла, т.е. оно является фиксированным числом. В этом случае переменная, которая считает количество повторений (шагов) цикла, называется *счетчиком цикла* (или параметром цикла, или управляющей переменной цикла).

Блок-схема циклического алгоритма в этом случае выглядит так:



Такая блок-схема хорошо иллюстрирует работу цикла со счетчиком. Перед выполнением первого шага цикла счетчику цикла должно быть присвоено начальное значение — любое число в зависимости от алгоритма. Если величина счетчика цикла не превышает конечное значение, то далее будет выполняться группа действий, составляющих тело цикла. После выполнения тела цикла счетчик цикла изменяется на определенную величину — *шаг изменения* счетчика цикла h . Если полученное значение счетчика цикла не превысит конечное значение, то цикл продолжится до тех пор, пока счетчик цикла не станет больше конечного значения — тогда управление передается действию, следующему за циклом.

В блок-схемах для изображения цикла со счетчиком используют блок «Подготовка». В блоке «Подготовка» записывается счетчик цикла (I), далее последовательно указываются начальное значение (I_n), конечное значение (I_k) счетчика цикла и шаг его изменения (h). Если шаг изменения h равен 1, его можно не записывать. Желательно, чтобы линия потока входила в блок сверху, линия потока к телу цикла выходила снизу, слева (или справа) входила линия потока перехода к следующему шагу цикла, а справа (или слева) выходила линия потока — выход из цикла.



При использовании цикла со счетчиком необходимо соблюдать некоторые требования:

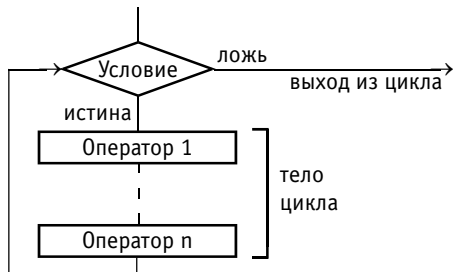
- в теле цикла нельзя принудительно изменять значение счетчика цикла;
- не разрешается передавать управление оператору тела цикла извне, т.е. вход в цикл допускается только через начало цикла.

Циклы с предусловием. Циклы с предусловием чаще всего используют тогда, когда неизвестно число повторений цикла. Циклы с предусловием — это такие циклы, в которых до начала выполнения тела цикла проверяется условие выполнения следующего шага цикла. Если значение этого условия истинно (т.е. условие выполняется), то выполняется тело цикла. В теле цикла должно изменяться значение по крайней мере одной переменной, которая влияет на значение условия (иначе произойдет «зацикливание»). Далее опять проверяется условие выполнения цикла, и если значение условия ложно, то осуществляется выход из цикла.

Можно использовать и еще один вариант этого цикла, когда проверяется не истинность значения условия, а ложность. В этом случае выход из цикла происходит, когда значение условия цикла становится истинным. Тот или иной вариант цикла используется в зависимости от того, какое условие в данном алгоритме программисту удобнее использовать.

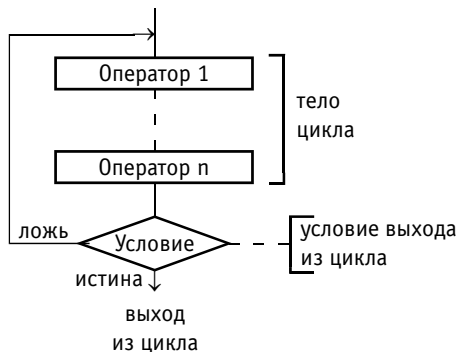
Особенность этого типа цикла в том, что тело цикла может не выполниться ни разу, если условие первоначально ложно в первом варианте (или истинно во втором).

На блок-схеме такой цикл реализуется следующей конструкцией:



Циклы с постусловием. Этот тип цикла также используется при неизвестном заранее количестве повторений цикла, но в отличие от цикла с предусловием здесь условие на выход из цикла проверяется после того, как выполнились операторы тела цикла, поэтому хотя бы один раз тело цикла будет обязательно выполнено.

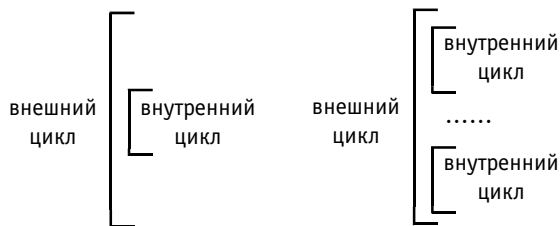
На блок-схеме этот тип цикла изображается следующим образом:



Сложные циклы. Циклы, которые содержат внутри себя (в теле цикла) один или несколько других циклов, называются *сложными* или *вложенными циклами*.

При этом циклы, охватывающие другие циклы, называются *внешними*, а циклы, входящие во внешние, — *внутренними циклами*.

На каждом шаге внешнего цикла внутренний цикл «прокручивается» полностью.



Контрольные вопросы

1. Перечислите этапы создания программного продукта.
2. Для чего выполняется структуризация программ?
3. Что такое модуль, какие виды модулей вы знаете?
4. Назовите и охарактеризуйте методы структурного проектирования.
5. На чем основывается объектно-ориентированный подход?
6. Что такое алгоритм?
7. Перечислите основные особенности алгоритмов.
8. Назовите способы представления алгоритмов.

9. Что такое блок-схема, каковы правила составления блок-схем?
10. Какой алгоритм называется линейным?
11. Перечислите виды разветвляющихся алгоритмов.
12. Какие виды циклов вы знаете?
13. Объясните работу циклов со счетчиком.
14. В чем отличие циклов с предусловием от циклов с постусловием?
15. Что такое сложные циклы?

Литература

1. *Евдокимов В.В. и др.* Экономическая информатика: Учебник для вузов. — СПб.: Питер, 1997. — С.534–549.
2. *Иванова Г.С.* Основы программирования: Учебник для вузов. — М.: Изд-во МГТУ им. Баумана, 2001.
3. Информатика: Учебник / Под ред. проф. Н.В.Макаровой. — М.: Финансы и статистика, 1998. — С.647–665.
4. *Криницкий Н.А., Миронов Г.А., Фролов Г.Д.* Программирование и алгоритмические языки / Под ред. А.А.Дородницына. — М.: Наука, 1979.
5. *Миллс Х., Хьюз Дж., Мичтом Дж.* Структурный подход к программированию. — М.: Мир, 1980.
6. *Прищепов М.А.* Экзамен по информатике. Основы алгоритмизации и программирования: Справочное пособие. — Минск: ТетраСистемс, 2001.
7. *Светозарова Г.И., Мельников А.А., Козловский А.В.* Практикум по программированию на языке Бейсик: Учебное пособие для вузов. — М.: Наука, 1988.
8. *Степанова Т.И.* Основы алгоритмизации и программирования: Учебное пособие. — Новосибирск, НГАЭиУ, 2002.

ТЕМА 2.2. ПРОГРАММИРОВАНИЕ

2.2.1. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ

Любая ЭВМ понимает свой внутренний язык — так называемый «машинный язык», где каждая команда представляет собой длинную последовательность нулей и единиц. Однако составление программ на машинных языках очень трудоемкий процесс, так как требует от программиста запоминания большого количества деталей, без которых невозможна запись программы на языке машинных команд. Причем трудоемкость и сложность возрастают с увеличением программы. Поэтому для облегчения труда программиста используются так называемые алгоритмические языки.

Алгоритмический язык — это определенный набор символов и специальных слов, которые в соответствии со строгими правилами записи команд (синтаксиса языка) описывают алгоритм решения задачи.

Если язык программирования ориентирован на конкретный тип процессора, то он называется *языком программирования низкого уровня*. Таким языком является, например, язык *Ассемблер*, в котором каждая команда машинного кода записывается не в виде чисел, а с помощью условных символьных обозначений. Языки программирования низкого уровня используются чаще всего для разработки системных приложений, так как они позволяют получить доступ ко всем аппаратным ресурсам компьютера.

Алгоритмические языки относятся к *языкам программирования высокого уровня*. Особенности архитектуры конкретного компьютера такие языки не учитывают, поэтому программы, созданные на таких языках, могут выполняться на любой ЭВМ, для которой имеется транслятор этого языка.

Высокоуровневые языки программирования в отличие от языков машинных команд предназначены для написания программ с помощью привычных человеку терминов и оперируют с командами, напоминающими естественный язык.

Все алгоритмические языки высокого уровня используют несколько основных языковых конструкций с несколькими десятками слов (операторов) и строгими правилами синтаксиса.

Языков программирования очень много, но наиболее распространенными стали лишь несколько: Алгол (Algol), Фортран (Fortran), Кобол (Cobol), ПЛ/1 (PL/1), Бейсик (Basic), Паскаль (Pascal), СИ (C) и др.

Любая ЭВМ может выполнять только программы на языке машинных кодов. Поэтому текст программы на алгоритмическом языке должен быть переведен на язык машинных кодов. Процесс равносильного преобразования алгоритма, заданного на языке программирования, в алгоритм на машинном языке называется *трансляцией*. Перевод программ на машинный язык осуществляют специальные программы.

Трансляторы — это специальные программы, которые переводят программы, написанные на языке высокого уровня, на язык машинных кодов.

Причем для каждого алгоритмического языка предназначен свой транслятор.

Транслятор выполняет следующие основные функции:

- производит синтаксический анализ исходного текста программы. При этом выводятся сообщения о синтаксических ошибках в тексте программы на алгоритмическом языке (например, недопустимые символы в операторах, недопустимые операторы или отсутствие знаков препинания и др.);
- производит непосредственную трансляцию исходной программы на язык машинных кодов;
- распределяет память для полученной программы.

Различают два подхода при переводе программы с языка высокого уровня на машинный язык: это интерпретация и компиляция.

Интерпретация — такой процесс, когда каждая команда переводится в машинный код и тут же выполняется, после чего эта процедура совершается над следующей командой. В случае ошибки эту команду можно исправить и продолжить интерпретацию. Такой способ медленный, но зато гибкий, так как обнаруженную ошибку можно исправить немедленно. К концу интерпретации программа готова к работе. Транслятор, который осуществляет пооператорное выполнение программы, называется *интерпретатором*.

Компиляция — процесс, при котором в машинный код переводится вся программа полностью. При этом формируется файл программы, который может быть запущен на выполнение (так называемый *загрузочный файл*). Трансляторы, которые формируют загрузочный файл, называются *компиляторами*. Откомпилированная программа выполняется гораздо быстрее (в десятки и даже сотни раз), но при наличии ошибок каждый раз весь процесс будет повторяться сначала.

Еще одно отличие — файл откомпилированной программы уже готов к выполнению и программа-компилятор во время выполнения алгоритма не нужна, в то время как интерпретируемые программы требуют постоянного присутствия программы-интерпретатора в оперативной памяти ЭВМ во время решения задачи.

Каждый из подходов (компиляция и интерпретация) имеет свои достоинства. В настоящее время для большинства алгоритмических языков созданы и компиляторы, и интерпретаторы, и часто в реальных системах программирования перемешаны технологии интерпретации и компиляции.

2.2.2. ФОРМЫ ПРЕДСТАВЛЕНИЯ ДАННЫХ В ЭВМ

Чтобы понимать, как работает любая ЭВМ, нужно прежде всего знать, как представляется информация в компьютере. Наиболее простым и надежным способом является так называемое «битовое» представление и хранение информации. Необходимо понимать, что любая информация — числовая, текстовая, графическая или звуковая — представляется в ЭВМ в виде двоичных кодов. Все программы при работе оперируют с данными. Основные типы данных в любом алгоритмическом языке — числовые и текстовые. Рассмотрим, как представляются данные в ЭВМ.

Целые числа. Целые числа не имеют дробной части. Для представления целых чисел используется форма с фиксированной точкой (запятой), т.е. для представления целых чисел отводится определенное постоянное количество разрядов. Количество разрядов определяет диапазон значений целых чисел. Например, 2 байта (16 разрядов) позволяют представить числа в диапазоне от 0 до 65535 без знака, или в диапазоне от -32768 до +32767 со знаком (один разряд отводится под знак числа).

Если диапазон целых чисел слишком мал, то следует увеличить количество разрядов, например, воспользоваться длинными целыми числами — 4 байта.

Действительные числа. Действительные числа могут иметь дробную часть и представляются в форме с плавающей точкой (запятой). В форме с плавающей точкой используется экспоненциальная форма записи числа. Например:

$123456.7 = 1.234567 \times 10^5$ можно представить как $1.234567E+5$,

$-0.00032 = -3.2 \times 10^{-4}$ можно представить как $-3.2E-4$.

Символ E в такой записи означает «10 в степени...».

В экспоненциальной форме любое число представляется в виде мантиисы числа, основания системы счисления (в десятичной системе счисления это 10) и целочисленного порядка:

$$N = \pm m E \pm p.$$

Следовательно, при представлении чисел с плавающей точкой необходимо в разрядную сетку записать мантиису m и порядок p со своими знаками. При этом:

- знак мантиисы определяет знак числа;
- число разрядов, выделенных для представления мантиисы, определяет точность представления чисел (чем больше разрядов, тем выше точность);
- число разрядов, выделенных для представления порядка, определяет диапазон чисел (чем больше разрядов, тем шире диапазон чисел).

Текстовые данные. Текстовые данные представляют собой набор любых символов алфавита определенного языка. Таблица, в которой устанавливается соответствие между символом и его кодом (порядковым номером в компьютерном алфавите), называется *таблицей кодировки*. Для представления текстовой информации в компьютерах типа IBM PC используется алфавит из 256 символов таблицы кодировки ASCII. Код ASCII (аббревиатура словосочетания American Standard Code for Information Interchange — стандартный американский код для обмена информацией) является одним из самых распространенных. Расширенная таблица ASCII состоит из двух частей: первая часть (коды 0–127) универсальная, а вторая часть (коды 128–255) предназначена для специальных символов и букв национальных алфавитов (в том числе и русского).

Длиной текста является количество символов, составляющих текст, включая знаки препинания и пробелы. Так как каждый символ таблицы ASCII занимает один байт памяти, следовательно, память, необходимая для хранения текста, определяется количеством символов в тексте, т.е. его длиной. Например, слово «Алгоритм» займет 8 байт памяти.

2.2.3. ОСНОВНЫЕ ПОНЯТИЯ ПРОГРАММЫ НА АЛГОРИТМИЧЕСКОМ ЯЗЫКЕ

Программа — это упорядоченная последовательность команд для ЭВМ, реализующая алгоритм решения какой-либо задачи.

По сути дела, программа — это форма представления алгоритма для исполнения его компьютером, поэтому программа может быть написана на языке машинных кодов, на языке программирования низкого уровня или на языке программирования высокого уровня — алгоритмическом языке.

В программе на алгоритмическом языке действия, которые должны выполняться в программе, задают *операторы*. Поэтому программа представляет собой последовательность операторов.

Язык программирования, как любой язык, имеет свои правила:

- *синтаксические правила*, определяющие допустимые конструкции языка. В синтаксис включается определение алфавита и правила записи конструкций языка (например, имен переменных, операторов);
- *правила, определяющие семантику языка*, т.е. смысл и правила использования его конструкций.

Основой любого языка, в том числе и алгоритмического, является *алфавит*. Элементами алфавита алгоритмического языка являются символы таблицы кодировки. Для записи основных конструкций языка, как правило, используются символы первой универсальной половины таблицы ASCII: буквы латинского алфавита, арабские цифры, знаки арифметических операций и специальные символы, например, знаки препинания, скобки. Русские буквы, как правило, используют в поясняющих комментариях к программе или для вывода информации.

Из символов алфавита строятся различные языковые конструкции. Простейшей языковой конструкцией является *слово*. Различают ключевые слова и слова пользователя. Любой алгоритмический язык имеет определенный набор служебных, так называемых *ключевых слов*. Каждое ключевое слово (например, FOR, IF, DO) несет определенную смысловую нагрузку. Эти слова не допускается изменять.

Любое искажение ключевого слова приводит к ошибке в программе!

Ключевые слова относятся к группе зарезервированных слов, поэтому их нельзя использовать не по назначению, например, для обозначения имен переменных.

Слова пользователя вводятся в программу самим программистом, и хотя эти слова вводятся произвольно, но и они должны записываться в соответствии с определенными правилами.

Более сложной конструкцией является *оператор*, задающий описание некоторого действия программы. Любой алгоритмический язык имеет набор операторов, реализующих операции присваивания, передачи управления, организации циклов и т.д.

2.2.4. ЯЗЫК ПРОГРАММИРОВАНИЯ VISUAL BASIC

С появлением графического интерфейса пользователи ПК получили возможность работать в интуитивно понятной среде. Однако теперь программистам приходилось в каждом приложении программировать прикладные и диалоговые окна, меню, списки с прокруткой, шрифты и т.д.

Решением проблемы стало создание в 1991 году языка Visual Basic (VB). В основе языка Visual Basic лежит алгоритмический язык Basic. В его составе есть специальные библиотеки, средства визуального проектирования и элементы объектно-ориентированного программирования.

В основе объектно-ориентированного программирования лежат понятия: объект, класс, инкапсуляция, наследование, полиморфизм, событие.

Объект — совокупность свойств (параметров) определенных сущностей и методов их обработки. В языках программирования объект понимается как совокупность свойств, методов их обработки (т.е. изменения этих свойств) и событий, на которые данный объект может реагировать. В VBA объектом являются элементы пользовательского интерфейса (это может быть форма пользователя или элементы, расположенные на этой форме, например, кнопки, полосы прокрутки, рисунки), рабочие листы, диапазоны ячеек и т.д.

Объекты могут объединяться в *класс* (группы, наборы) — совокупность объектов с общими методами обработки или свойствами.

Объединение объектов с методами в одном классе называется *инкапсуляцией*. Инкапсуляция — это скрытие информации о внутренней структуре объекта. Доступ к объекту возможен только через его свойства и методы.

Один объект может выступать объединением вложенных в него по иерархии других объектов и иметь образованные от него подклассы, при этом осуществляется наследование данных и методов обработки объектов исходного класса. *Наследование* — это возможность создания на основе одного класса новых классов с наследованием всех его свойств и методов и добавлением новых.

Методы классов-наследников могут переопределяться по сравнению с методами базового класса. Такое свойство объектов переопределять методы наследуемого класса называется *полиморфизмом*.

Система визуального объектно-ориентированного программирования позволяет за счет использования средств визуального проектирования существенно упростить значительную часть программистской работы — создание интерфейса прикладной программы.

Кроме того, Visual Basic поддерживает событийно-управляемое программирование, когда вместо описания каждого шага программы программист лишь указывает, как реагировать на различные события: нажатие кнопки, щелчок кнопкой мыши по объекту и т.п.

Таким образом, работая в Visual Basic, программист создает не одну большую программу, а приложение, состоящее из совокупности взаимодействующих процедур, управляемых пользователем.

При появлении приложений, объединенных в пакет Microsoft Office (Word, Excel для Windows и др.), возникла проблема их интеграции, поскольку все они предназначались для работы с документами и автоматизации экономических расчетов. Во всех этих приложениях есть

возможность заменить повторяющиеся действия пользователя последовательностью машинных команд — *макросов* или *макроопределений*. Основным языком для разработки таких макроопределений стал Visual Basic. Появились его версии: Word Basic, Excel Basic и т.д.

Для последующих версий приложений MS Office фирма Microsoft использует единый и мощный язык программирования Visual Basic, приспособивая его к специфике приложений. Эта версия языка носит название Visual Basic for Applications (Visual Basic для приложений) или VBA.

Решение прикладной задачи с помощью VB выполняется с использованием *проекта*. Проекты содержат *модули*, а модули включают *процедуры* и *функции*. В Visual Basic программный код оформляется в виде процедур и функций. Разработанные отдельные функции или процедуры можно накапливать в библиотеках и в дальнейшем использовать по мере необходимости.

2.2.5. ОБЪЕКТЫ, СВОЙСТВА, МЕТОДЫ

Создание программы на VB включает в себя два этапа.

Этап визуального программирования. На этом этапе не требуется вводить тексты программ на языке программирования — на этом этапе программа проектируется с помощью инструментов VB. То есть производится построение объектов (форм, кнопок и т.п.) и задание свойств этим объектам (цвет, шрифт и др.). На этапе визуального программирования представляется будущий вид создаваемого приложения (программы).

Этап программирования в исходном коде. На этапе программирования вводится текст программы, для этого используется встроенный текстовый редактор. Программа состоит из операторов языка VB. Исходный текст программы называется *исходным кодом* или *программным кодом*.

Программы, созданные в VB, будут являться стандартными приложениями Windows, это означает, что можно использовать буфер обмена, вырезать, копировать и вставлять тексты, графику и т.д.

Объектно-ориентированный подход использует следующие базовые понятия: объект, свойства объекта, метод обработки.

Объект — это то, чем управляют с помощью программы (например, кнопка, полоса прокрутки, рабочий лист, диаграмма, ячейка таблицы и т.д.).

Свойство — характеристика объекта, его параметр. Все объекты наделены определенными свойствами, которые в совокупности выделяют объект из множества других объектов, задают его *качественную определенность*, обуславливают независимость создания и обработки других объектов. Например:

ОБЪЕКТ_А (свойство 1, свойство 2, ..., свойство k)

ОБЪЕКТ_В (свойство m , свойство $m+1$, ..., свойство $m+n$)

Свойства объектов различных классов могут «пересекаться», т.е. возможны объекты, имеющие одинаковые свойства:

ОБЪЕКТ_С (свойство 1, свойство m , ..., свойство k)

Существует определенный формат программного кода, задающего установку свойства:

Объект. Свойство = Значение

Установка значений свойств — это один из способов управления объектами.

Метод — это программа действий над объектом или его свойствами. Метод рассматривается как программный код, связанный с определенным объектом, с помощью которого осуществляется преобразование свойств объекта либо изменение его поведения. То есть методы — это команды, с помощью которых выполняются некоторые преобразования объектов.

Объект может обладать набором заранее определенных встроенных методов обработки, либо созданных пользователем или заимствованных в стандартных библиотеках.

Синтаксис вызова метода объекта:

Объект. Метод [Аргументы]

Событие — изменение состояния объекта.

Программный продукт, созданный с помощью инструментальных средств объектно-ориентированного программирования, содержит объекты с их свойствами, для которых разработан графический интерфейс пользователя. Как правило, работа с программным продуктом осущест-

вляется с помощью экранной формы, с объектами управления, которые содержат методы обработки, вызываемые при наступлении определенных событий. Экранные формы также используются для выполнения заданий и перехода от одного компонента программы к другому. Для объектов управления уточняется перечень событий и создаются пользовательские методы обработки — программный код на языке программирования в виде событийных процедур.

2.2.6. ВИДЫ И ТИПЫ ДАННЫХ

Любая программа предназначена для обработки данных. *Данные* (англ. data, information) — это информация, хранящаяся в ЭВМ и представленная в форме, которая обеспечивает возможность ее хранения, обработки и передачи.

Для обработки в ЭВМ данные представляются в виде величин — элементов данных.

Все данные можно разделить на константы и переменные.

Константа (англ. constant) — величина, значение которой постоянно и не изменяется при выполнении программы.

Переменная (англ. variable) — некоторая величина, которая может изменяться, принимая в процессе выполнения программы различные значения.

Программа предназначена для обработки информации, которая может быть представлена в числовом виде, символьном, графическом, звуковом. Поэтому каждая величина должна быть объявлена или описана, т.е. указан ее тип, какая она — символьная, целое или вещественное число и т.д. Число ячеек, отводимых для переменной, зависит от типа этой переменной.

Тип переменной определяет:

- размер памяти компьютера, необходимый для ее внутреннего представления;
- возможные значения переменной;
- операции, которые могут выполняться над переменной.

Каждый тип описывается своим ключевым словом. В зависимости от объема памяти, отведенного для хранения данных определенного типа (длина в байтах, указанная в таблицах, — длина внутреннего представления данных), значения их должны укладываться в определенный диапазон.

Основные (встроенные) типы данных

Тип	Хранимая информация	Занимаемая память (в байтах)	Диапазон значений
boolean	Логические значения	2	0 — FALSE, 1 — TRUE
byte	Целые числа	1	0...255
integer	Целые числа	2	-32768...32767
long	Длинные целые числа	4	-/+2.1E9
decimal	Десятичное число	14	Целое 29 знаков Вещественное — 27 знаков после запятой
single	Вещественные одинарной точности	4	-3.4E38...-1.4E-45 1.4E-45...3.4E38
double	Вещественные двойной точности	8	-1.7E308...-4.9E-324 4.9E-324...1.7E308
currency	Денежные единицы	8	-922337203685477.5808... +922337203685477.5808
string	Текстовая информация (строка переменной длины)	Количество символов + 10	Количество символов <=2000000000
string*n	Текстовая информация (строка фиксированной длины)	Количество символов	n<=65400
date	Информация о дате и времени	8	От 1 января 1000 г. до 31 декабря 9999 г.
object	Рисунок или ссылка на любой другой объект	4	
variant	Значения любого из перечисленных типов	Не менее 16	Любое числовое или строковое значение

Каждому данному в программе необходимо дать *имя* или *идентификатор* (от лат. identifico — отождествлять).

Правила использования идентификаторов:

- имя или идентификатор — это последовательность, содержащая латинские буквы, цифры, начинающаяся с буквы;
- идентификатор может содержать не более 255 символов, но различаются первые 31 символов;

- в качестве идентификаторов нельзя использовать зарезервированные слова;
- не допускается использовать в идентификаторах пробел, точку, разделительные символы, знаки операций, а также специальные символы.

Пользовательские типы записей. Очень часто в программах требуется хранить и обрабатывать информацию о нескольких объектах одной и той же структуры. Например, информацию о студентах: фамилию, имя, отчество, год рождения и т.д. Можно хранить эту информацию в виде отдельных массивов. Но в современных языках программирования существуют специальные средства, позволяющие работать со сложными типами данных, которые содержат фиксированное число элементов — полей. Каждое поле имеет имя, которое задается ему при описании, и может иметь свой тип.

Структура данных, состоящая из определенного числа разнотипных компонент (полей), называется *записью*.

В Visual Basic для описания записей можно использовать тип, определяемый пользователем, это так называемый *пользовательский тип*, с помощью которого можно описать свой достаточно сложный тип данных и в дальнейшем работать с ним как с базовым типом (т.е. целый, строковый и т.д.).

Создание нового типа осуществляется следующей конструкцией:

```

TYPE <имя типа>
<имя 1 поля> AS <тип 1 поля>
<имя 2 поля> AS <тип 2 поля>
...
<имя n поля> AS <тип n поля>
END TYPE

```

Например, пользовательский тип, предназначенный для хранения фамилии, имени и года рождения студентов, может быть объявлен следующим образом:

```

TYPE Student
fam AS STRING 'Фамилия
name AS STRING 'Имя
bdate AS DATE 'Дата рождения
END TYPE

```

Созданный тип данных может быть использован в программе. Для этого в разделе объявлений переменных надо объявить переменную этого пользовательского типа. Например,

```

DIM starosta AS Student

```

Пользовательский тип данных может, в свою очередь, содержать структурные элементы, тип которых также является пользовательским.

2.2.7. ОПИСАНИЕ ПЕРЕМЕННЫХ

Объявление переменных можно осуществлять на двух уровнях — на уровне процедуры (т.е. только в процедуре, в которой они объявлены) и на уровне модулей (т.е. только в процедурах модуля, в котором они объявлены, но не в других модулях проекта).

Область программы, в которой может быть использована переменная, называется *областью видимости переменной*.

Область видимости переменной задается при ее объявлении одним из ключевых слов:

Dim — объявляет локальные переменные, существующие только во время вызова процедур, в которых они объявлены. Если переменная объявляется в разделе глобальных объявлений модуля, то она доступна для всех процедур этого модуля. Для других модулей будет не видна.

Private — не может объявлять переменную внутри процедуры. При объявлении в разделе глобальных объявлений модуля Dim и Private равнозначны.

Public — объявленная таким образом переменная является глобальной на уровне приложения и доступна из всех его модулей.

Static — для объявления статических переменных, т.е. при выходе из процедуры их значения сохраняются, хотя являются недоступными, пока процедура не будет вызвана вновь.

Перед использованием переменных в программе их требуется *описать* (*объявить, декларировать*). Объявление переменных может быть явным или неявным. Для явного описания переменных существуют специальные операторы описания. Синтаксис оператора описания:

[Static/Private/Public] DIM <имя переменной> [AS <тип>]

Примеры:

DIM a AS string

Private DIM y1, y2 AS integer

DIM x AS integer, s AS long

Другим способом явного объявления переменных является указание типа с помощью *суффикса* — специального символа в конце имени. В этом случае использование ключевого слова *As* не требуется:

[Static/Private/Public] DIM <имя переменной>Суффикс

Ниже в таблице приведены типы переменных и соответствующие им суффиксы.

Суффикс	Тип
%	integer
&	long
!	single
#	double
\$	string
@	currency

Например:

DIM Fam\$

Static N%

Хороший стиль программирования предполагает использование явного объявления переменных, поэтому неявное объявление рассматривать не будем.

Использование констант. Различают именованные и неименованные константы.

Именованные константы должны быть описаны в программе в разделе описания. Обращение к ним в программе осуществляется по имени:

[Public/Private] Const <имя константы> [As <тип>]=<выражение>

Например:

Const Pi As Single = 3.1415

Однажды определив константу, ее нельзя будет случайно изменить.

Неименованная константа (литерал) представляет собой значение константы, записанное непосредственно в операторе программы.

Числовые константы:

- целые со знаком (**-123**) или без знака (**54**);
- вещественные (с десятичной точкой) — **-57.09**.

Символьные константы заключаются в двойные кавычки — **“10 кг”**.

2.2.8. ПРОГРАММИРОВАНИЕ ЛИНЕЙНЫХ ПРОЦЕССОВ

Самые простые алгоритмы представляют собой так называемые линейные процессы. Линейные алгоритмы — наборы команд, выполняемые последовательно во времени друг за другом.

2.2.8.1. Оператор присваивания

Одной из основных операций в таких алгоритмах является *операция присваивания*. Присваивание — одно из основных действий, выполняемых компьютером. С помощью операции присваивания записывают действия, которые позволяют изменять значения переменных. Синтаксис этой операции в общем виде можно записать следующим образом:

<имя переменной>=<выражение>

Знак «=» здесь читается «присвоить» (не путайте со знаком «равно»). Суть этой операции заключается в том, что вычисляется результат выражения, стоящего в правой части от знака «=», и этот результат записывается в ячейку памяти, обозначенную идентификатором переменной,

стоящей слева от знака «=». Поэтому операция $X=X+1$ означает, что к значению переменной X прибавляется 1 и результат снова записывается в ячейку X , т.е. значение переменной X увеличилось на 1.

Для того чтобы правильно записывать выражения в программе, необходимо знать некоторые правила записи выражений:

1. Все выражения записываются в одну строку с учетом порядка выполнения операций.
2. Для изменения порядка выполнения операций используют круглые скобки.
3. Порядок выполнения операций в выражении определяется приоритетом арифметических операций. Ниже в таблице записаны операции в порядке убывания их приоритета и обозначение этих операций в программе. Операции, имеющие равный приоритет, записаны в одной строке таблицы.

Название операции	Обозначение	Пример
Возведение в степень	\wedge	$X^2 \rightarrow X^2$
Остаток от деления (только для данных целого типа)	$X \bmod Y$ где X — делимое, Y — делитель	$7 \bmod 3$ <i>Ответ: 1</i>
Целочисленное деление (целая часть результата деления)	$X \setminus Y$ где X — делимое целого типа, Y — делитель целого типа	$7 \setminus 3$ <i>Ответ: 2</i>
Умножение, деление	$*, /$	$2 * X + Y / 3$
Сложение, вычитание	$+, -$	$X + Y - 5.3$

Пример. Записать оператор для вычисления y по формуле:

$$y = \frac{(3x+2) \times 4}{5x}$$

Оператор присваивания записывается следующим образом:

$$y = (3 * x + 2) * 4 / (5 * x)$$

2.2.8.2. Использование функций

Любой язык программирования имеет определенный набор стандартных встроенных функций. Среди таких функций обязательно есть группа числовых функций — функций, которые имеют числовые аргументы и возвращают числовые значения. Ниже в таблице представлены наиболее популярные числовые функции.

Обозначение функции	Назначение функции
ABS(X)	Абсолютная величина числа
ATN(X)	Арктангенс
COS(X)	Косинус, где X — угол в радианах
EXP(X)	Экспоненциальная функция, где $e=2.71828$
FIX(X)	Целая часть аргумента
INT(X)	Максимальное целое число, не превышающее X
LOG(X)	Натуральный логарифм, где X — положительное число
RND(X)	Формирование случайного числа
SGN(X)	Определяет знак аргумента, возвращает: +1 при $X > 0$, 0 при $X = 0$, -1 при $X < 0$
SIN(X)	Синус, где X — угол в радианах
SQR(X)	Корень квадратный из X , где X — положительное число
TAN(X)	Тангенс, где X — угол в радианах

Пример. Записать оператор присваивания для вычисления функции:

$$z = \sqrt{\frac{|x^2 + 6|}{|\sin(x)|}}$$

$$z = \text{SQR}(\text{ABS}((x^2 + 6) / \text{SIN}(x)))$$

2.2.9. РАЗВЕТВЛЯЮЩИЕСЯ АЛГОРИТМЫ

2.2.9.1. Программирование разветвляющихся процессов

Разветвляющиеся алгоритмы реализуются в программе с помощью оператора условия **If ... Then** и переключателя, реализуемого оператором **Select Case**.

Оператор условного перехода IF. Этот оператор изменяет порядок дальнейших действий в зависимости от результата проверки значения некоторого выражения или логического условия.

Оператор IF может использоваться в программе как в полном, так и в сокращенном виде (без ELSE). В общем виде формат этого оператора:

```
IF <условие> THEN <оператор 1> [ELSE <оператор 2>]
<оператор 3>
```

В полном виде оператор IF используется для реализации в программе алгоритма типа «Разветвление». Действие этого оператора заключается в следующем:

- проверяется условие (или значение логического выражения);
- если условие выполняется (т.е. значение логического выражения «истинно»), то далее будет выполняться *оператор 1*, а после его выполнения *оператор 3*;
- если условие не выполняется (т.е. значение логического выражения «ложно»), то далее будет выполняться *оператор 2*, а после его выполнения *оператор 3*.

Оператор IF в сокращенном виде реализует в программе алгоритм типа «Обход» и работает следующим образом:

- проверяется условие (или значение логического выражения);
- если условие выполняется (т.е. значение этого выражения «истинно»), то далее будет выполняться *оператор 1*, а после его выполнения *оператор 3*;
- если условие не выполняется (т.е. значение этого выражения «ложно»), то сразу же будет выполняться *оператор 3*.

2.2.9.2. Логические выражения

Логические выражения используют в операторе IF для записи условий, определяющих дальнейший ход выполнения программы.

Простые логические выражения используют для записи простых условий, включающих одну операцию отношения. Напомним, что операции отношения — это одна из шести операций:

Операция отношения	Знак отношения	Обозначение операции в программе
Равно	=	=
Не равно	≠	< >
Больше	>	>
Больше или равно	≥	>=
Меньше	<	<
Меньше или равно	≤	<=

Примеры:

x	y	Логическое выражение	Значение логического выражения
7		x>0	Истина (TRUE)
-7		x>0	Ложь (FALSE)
6	4	y=x	Ложь (FALSE)
4	4	y=x	Истина (TRUE)
49	60	x+y<100	Ложь (FALSE)
49	50	x+y<100	Истина (TRUE)

Сложные логические выражения используют для записи сложных условий, которые представляют собой простые условия (логические выражения), соединенные логическими операциями.

В сложных логических выражениях операции отношения имеют больший приоритет, чем логические операции. Следовательно, первыми выполняются операции отношения, а их результаты (TRUE — истина или FALSE — ложь) являются операндами для логических операций.

Для логических операций строят так называемые *таблицы истинности*, в которых рассматриваются все возможные сочетания значений операндов, соединенных логическими операциями, и результаты логических операций для соответствующих операндов.

Наиболее часто используются следующие логические операции (по убыванию приоритетов):

Обозначение операции	Название операции
NOT	НЕ — логическое отрицание
AND	И — конъюнкция (логическое умножение)
OR	ИЛИ — дизъюнкция (логическое сложение)
XOR	Исключающее ИЛИ

Таблица истинности основных логических операций

Обозначение операции в программе	Значение операндов		Результат операции
	X	Y	
NOT x	Ложь (FALSE) Истина (TRUE)		Истина (TRUE) Ложь (FALSE)
x AND y	Ложь (FALSE) Ложь (FALSE) Истина (TRUE) Истина (TRUE)	Ложь (FALSE) Истина (TRUE) Ложь (FALSE) Истина (TRUE)	Ложь (FALSE) Ложь (FALSE) Ложь (FALSE) Истина (TRUE)
x OR y	Ложь (FALSE) Ложь (FALSE) Истина (TRUE) Истина (TRUE)	Ложь (FALSE) Истина (TRUE) Ложь (FALSE) Истина (TRUE)	Ложь (FALSE) Истина (TRUE) Истина (TRUE) Истина (TRUE)
x XOR y	Ложь (FALSE) Ложь (FALSE) Истина (TRUE) Истина (TRUE)	Ложь (FALSE) Истина (TRUE) Ложь (FALSE) Истина (TRUE)	Ложь (FALSE) Истина (TRUE) Истина (TRUE) Ложь (FALSE)

Примеры:

x	y	Запись логического выражения	Значение логического выражения
4	0	x>0 AND y>0	FALSE
4	9		TRUE
-1	-5		FALSE
3	3	x>=y OR x<=0	TRUE
-5	7		TRUE
4	10		FALSE
30		x>0 AND x<=25	FALSE
12			TRUE
0			FALSE

Если в операторе IF в случае выполнения (или невыполнения) условия необходимо выполнить несколько операторов (т.е. если оператор занимает более одной строки), то он должен заканчиваться END IF:

```
If <условие> Then
<Оператор 1>
<Оператор 2>
...
Else
<Оператор n>
End If
```

Оператор IF может использоваться для создания так называемых многоуровневых проверок: когда одна из ветвей разветвления, в свою очередь, также имеет разветвление. То есть один оператор IF содержит внутри себя еще операторы IF. Такая конструкция называется «вложенные операторы IF».

При использовании вложенных операторов If нужно помнить правило: альтернатива Else считается принадлежащей ближайшему оператору If, не имеющему ветви Else.

Пример. Вывести сообщение о том, достиг ли гражданин пенсионного возраста, проанализировав его пол и возраст.

```
IF pol = "м" THEN
IF wozrast >= 60 THEN
Text1.Text = "Пенсионер"
ELSE
Text1.Text = "Моложе 60 лет"
END IF
END IF
```

Если необходимо проверить дополнительное условие в случае невыполнения исходного условия, то используется такая синтаксическая конструкция:

```
IF <условие 1> THEN
<Оператор 1>
ELSEIF <условие 2> THEN
<Оператор 2>
ELSE
<Оператор 3>
END IF
```

В этом случае для оператора ELSEIF не ставится в соответствие END IF.

2.2.9.3. Переключатели

Оператор IF позволяет реализовать выбор между двумя вариантами дальнейших действий. Однако иногда в программе необходимо выбрать один из нескольких вариантов. Конструкция вложенных операторов IF при этом может быть очень сложной и запутанной, поэтому в современных языках используют другие конструкции множественного выбора.

В последнее время для реализации множественного выбора большинство популярных языков программирования имеют оператор выбора. Оператор выбора используется для реализации нескольких альтернативных вариантов действий в зависимости от значения выражения, называемого *переключатель*.

Формат оператора в общем виде:

```
SELECT CASE <переключатель>
CASE <условие 1>
<оператор 1>
CASE <условие 2>
<оператор 2>
...
[CASE ELSE
<блок операторов n>]
END SELECT
```

Работает этот оператор следующим образом:

- вычисляется значение выражения-переключателя;
- далее начинается проверка соответствия полученного значения условию 1;
- в случае соответствия этих значений выполняется *оператор 1* и далее управление передается оператору, следующему за оператором END SELECT;
- если значение переключателя не соответствует условию 1, то проверяется соответствие условию 2 и т.д.;
- если же не найдено соответствие ни одному из условий, то выполняются операторы, следующие за CASE ELSE, или выполняется оператор, следующий за оператором END SELECT, если CASE ELSE отсутствует.

При записи оператора выбора следует иметь в виду, что оператор выполняет проверку условий строго по порядку. Поэтому надо следить за тем, чтобы условия не «перекрывали» друг друга. Например, если первым поставить условие $x > 0$, а затем $x > 10$, то первое условие «перекроет» второе (и до второго условия оператор не дойдет).

Типы результата выражения-переключателя и условия должны соответствовать друг другу. Условие выбора может содержать список, в котором элементы отделяются запятыми, эти элементы списка условия можно записать тремя способами:

- как выражения или константы;
- задать границы диапазона:
 <нач. значение> TO <конеч. значение>
- задать условие через операцию отношения:
 IS <знак операции> <выражение>

Пример:

```
SELECT CASE x
CASE 1, 3, 5
y=x^2
CASE 6 TO 10, 20
y=x+2
CASE 25, IS >= 30
y=2*x
CASE ELSE
y=2*x+1
END SELECT
```

2.2.10. ПРОГРАММИРОВАНИЕ ЦИКЛИЧЕСКИХ ПРОЦЕССОВ

2.2.10.1. Оператор цикла FOR

Для реализации цикла со счетчиком используется оператор FOR.

Общий формат оператора:

```
FOR <счетчик цикла> =<нач. знач> TO <конеч. знач> [STEP <шаг>]
<оператор 1>
...
<оператор n>
NEXT [<счетчик цикла>],
```

где <счетчик цикла> — это числовая переменная. В начале выполнения цикла она принимает значение, задаваемое выражением <нач. знач>;

<оператор 1> ... <оператор n> составляют тело цикла, которое будет выполняться заданное число раз;

<шаг> определяет значение, на которое будет изменяться счетчик цикла после выполнения каждого шага цикла.

Шаг изменения счетчика цикла может быть любым положительным или отрицательным числом, а также выражением. Если шаг больше 0, то цикл завершится, когда <счетчик цикла> станет больше <конеч. знач>. Если шаг меньше 0, то цикл завершится, когда <счетчик цикла> станет меньше <конеч. знач>. Если шаг равен 1, то его в операторе можно не записывать.

Чтобы прервать цикл до его завершения и передать управление первому оператору, следующему за NEXT, используют оператор EXIT FOR.

2.2.10.2. Операторы циклов с предусловием

1. Тело цикла будет выполняться до тех пор, пока значение условия *истинно*.

```
DO WHILE <условие>
<оператор 1>
...
<оператор n>
LOOP
```

2. Тело цикла будет выполняться до тех пор, пока значение условия *ложно*.

```
DO UNTIL <условие>  
<оператор 1>  
...  
<оператор n>  
LOOP
```

Чтобы прервать цикл до его завершения и передать управление первому оператору, следующему за LOOP, используют оператор EXIT DO.

2.2.10.3. Операторы циклов с постусловием

1. Тело цикла будет выполняться до тех пор, пока значение условия *ложно*.

```
DO  
<оператор 1>  
...  
<оператор n>  
LOOP UNTIL <условие>
```

2. Тело цикла будет выполняться до тех пор, пока значение условия *истинно*.

```
DO  
<оператор 1>  
...  
<оператор n>  
LOOP WHILE <условие>
```

2.2.11. МАССИВЫ

В Visual Basic различают два вида переменных — простые переменные и переменные структурного типа. Переменные структурного типа предназначены для идентификации нескольких данных. Частным случаем переменной структурного типа является массив.

Массив — это упорядоченный набор однотипных значений — элементов массива.

Массив имеет имя, размер и базовый тип, которые задаются при описании массива.

Тип элементов массива определяет тип самого массива и называется *базовым типом*.

Имя массива записывается по тем же правилам, что и имена переменных. Имя массива задает имя каждому элементу массива: каждый элемент массива обозначается этим именем и индексом, который определяет номер (место) этого элемента в массиве.

Размер (или размерность) массива — это количество элементов в массиве.

Массивы могут быть одномерными и многомерными. Но мы ограничимся рассмотрением только одномерных и двумерных массивов.

Одномерные массивы — массивы, в которых элементы пронумерованы последовательно по порядку: первый элемент, второй, третий и т.д. Для обозначения элементов одномерного массива используется один индекс.

Двумерные массивы — массивы, в которых данные условно организованы в виде таблицы (матрицы), где положение каждого элемента определяется номером строки и номером столбца. Для обозначения элементов двумерного массива используются два индекса: первый индекс для обозначения номера строки, второй индекс для обозначения номера столбца.

Значения индексов можно задать непосредственно числом (прямая адресация) — a(1), a(4,2) или косвенно, указав в индексе идентификатор переменной, которая позволит вычислить индекс (косвенная адресация) — a(i), a(i, j+2).

При работе с массивами в программе они должны быть объявлены (описаны), т.е. указаны имя массива, тип элементов массива, его размерность.

При обращении к элементу массива, значение индекса которого выходит за допустимые границы, появляется сообщение об ошибке.

Синтаксис оператора описания массива:

```
[Public/Private] DIM <ИмяМассива>(<размерность>) AS <ТипДанных>
```

В качестве размерности указывается либо верхняя граница изменения индекса, либо конкретный диапазон изменения значений индексов: **In TO Ik**.

Пример:

```
DIM s1 (15), s2 (15) AS STRING
```

```
DIM c (9,11) AS INTEGER
```

Нижняя граница изменения индексов по умолчанию равна 0. То есть если описан массив:

```
DIM a(15) AS INTEGER
```

это значит, что можно обращаться к элементам: a(0), a(1), a(2), a(3), ..., a(15).

Для изменения начала отсчета индексов служит специальный оператор, который устанавливает начало отсчета либо с 0, либо с 1:

```
OPTION BASE 1
```

В этом случае:

```
DIM a(15) AS INTEGER
```

описывает массив a(1), a(2), ..., a(15).

Иногда в процессе выполнения программы размер массива требуется изменить. В этом случае первоначально массив объявляется как динамический (в объявлении не указывается его размерность).

Количество элементов в динамическом массиве и его размерность в процессе выполнения программы можно переопределить с помощью оператора ReDim. Например:

```
Dim s ( ) As String
```

```
...
```

```
n = 15
```

```
ReDim s (n)
```

```
...
```

2.2.12. ПРОЦЕДУРЫ И ФУНКЦИИ

Рассмотрим элементы, которые могут входить в состав проекта.

Процедура представляет собой поименованную последовательность совместно выполняемых инструкций (операторов).

Различают следующие типы процедур:

- процедуры обработки событий;
- процедуры макросов;
- процедуры пользователя.

Процедуры обработки событий связаны с каким-либо объектом, вызываются тогда, когда происходит соответствующее событие, и имеют следующий синтаксис:

```
Private Sub <ИмяОбъекта_Событие( )>
```

```
<Код обработки события>
```

```
End Sub
```

где *ИмяОбъекта* — имя объекта, с которым связывается процедура;

Событие — вид обрабатываемого события.

Например, процедура обработки события Click (щелчок) для объекта CommandButton1 (кнопка) имеет вид:

```
Private Sub CommandButton1_Click( )
```

```
<Код обработки события>
```

```
End Sub
```

Процедуры обработки макросов создаются при записи макросов. Макрос можно вызывать из программы оператором вызова процедуры, комбинацией клавиш либо щелчком на значке, связанным с макросом. Эти процедуры имеют синтаксис:

```
Sub <Имя макроса( )>
```

```
<Программный код макроса>
```

```
End Sub
```

Процедура пользователя создается им для выполнения каких-либо типовых действий в рамках разрабатываемой программы. Синтаксис процедуры пользователя:

```
[Private/Public] Sub <ИмяПроцедуры [(СписокПараметров)]>  
<Тело подпрограммы (строки кода)>  
End Sub
```

Необязательный элемент *<СписокПараметров>* программы служит для передачи процедуре исходных данных для вычислений и состоит из элементов списка, разделенных запятыми. Элемент списка параметров имеет синтаксис:

```
<ИмяЭлемента> As <ТипДанных>
```

При запуске процедур выполняются ее команды, а затем управление передается в вызвавшее данную процедуру приложение или процедуру. В процедурах типа *Sub* обычно содержится большинство исполняемых программ.

Функция отличается от предыдущего типа процедур тем, что обязательно возвращает одно результирующее значение, которое присваивается имени функции.

Visual Basic позволяет пользователю создавать собственные функции — функции пользователя.

Функция пользователя имеет следующий синтаксис:

```
[Private/Public] Function <ИмяФункции [(СписокПараметров)]>_As<Тип данных>  
<Тело функции (строки кода)>  
<ИмяФункции> = <ВозвращаемоеЗначение>  
End Function
```

где *<Тип данных>* — тип данных возвращаемого значения;

<ВозвращаемоеЗначение> — значение, возвращаемое функцией.

Обращение к функции может производиться из процедуры, из другой функции. Если в функции предусмотрено рекурсивное обращение, то ее можно вызвать из нее самой.

При обращении к функции в программном операторе указывается имя функции и передаваемые ей параметры.

Программа Visual Basic представляет собой совокупность процедур и функций, размещенных в зависимости от особенностей решаемой задачи в одном или нескольких модулях. Число модулей в составе проекта зависит от сложности проекта.

Каждый модуль имеет две области: общую область и область подпрограмм. В общей области помещаются операторы описания переменных, которые являются общими для всех процедур и функций этого модуля. В области подпрограмм помещается код программы.

2.2.13. ОТЛАДКА ПРОГРАММ

Для того чтобы задача была решена с помощью ЭВМ, нужно составить программу и выполнить ее на компьютере. При этом приходится пройти через определенные этапы: разработка алгоритма, проектирование общей структуры программы, кодирование или составление программы. После того как программа составлена, ее выполняют на компьютере. При выполнении программы могут возникать разного рода ошибки.

При разработке приложений на Visual Basic также возможны ошибки. Их можно разделить на три вида: синтаксические, ошибки при выполнении программы и логические ошибки.

Синтаксические ошибки связаны с опечатками при вводе текста программ или с недостаточным знанием правил языка программирования. Такие ошибки система выявляет либо при вводе текста программы, либо при попытке выполнения программы.

Ошибки при выполнении программы случаются, если обнаруженная синтаксическая ошибка или внешнее событие вызвали остановку работы программы. Например, использовано неправильное имя файла или произошло обращение к выключенному принтеру и т.д.

Логические ошибки являются следствием плохо разработанного алгоритма или неучета возможных значений входных данных и т.д. В этих случаях при выполнении программы получают неверные результаты.

Устранение ошибок программирования — это *отладка*. А доказательство того, что программа работает «правильно», — *верификация*. При отладке и верификации программы необходимо

проверить работоспособность программы при различных вариантах исходных данных, чтобы проверить работу всех возможных ветвей в программе.

В любой среде программирования имеются различные средства, которые можно использовать для отслеживания и исправления ошибок.

Так, например, возможность простого обнаружения необъявленных переменных позволяет устранить большое число ошибок.

Для обнаружения логических ошибок можно использовать некоторые приемы, используя специальные средства программной среды:

- в программе устанавливают *точки останова* (контрольные точки). Точки останова — это строки в программе, где прерывается выполнение программы. Имея прерываемую таким способом программу, легко проверять значения переменных, перемещаться строка за строкой по коду программы, делать небольшие изменения в коде или продолжить работу программы;
- выполняют программу в *пошаговом режиме* для проверки каждой строки кода программы в порядке ее выполнения;
- наблюдают за *значениями переменных*. Для этого устанавливают контрольную точку на той строке программы, которая содержит проверяемую переменную;
- создаются специальные процедуры — обработчики ошибок, которые вызываются в случае возникновения ошибок при выполнении программы.

Кроме того, после отладки программы необходимо провести *анализ полученных результатов*. На этом этапе сравниваются результаты работы программы, например, с результатами наблюдений или с результатами, рассчитанными вручную, или с результатами, полученными экспериментальным путем и т.д. Если результаты отличаются, то, возможно, придется изменить алгоритм или саму модель. Иногда такой этап называют «*опытной эксплуатацией*».

После того, как анализ полученных результатов показал, что программа удовлетворяет всем требованиям заказчика, программа передается заказчику в эксплуатацию.

Следующий этап — *сопровождение программы* — предполагает устранение некоторых недостатков программы, замеченных в процессе эксплуатации программы, консультации и при необходимости обучение заказчика работе с программой.

Контрольные вопросы

1. Какие языки относятся к языкам высокого уровня?
2. Чем отличается интерпретация от компиляции?
3. Дайте определение объекта, свойства, класса объектов.
4. Что такое метод обработки объекта?
5. Что такое событийное программирование?
6. Как представляется в памяти компьютера числовая, текстовая информация?
7. Что такое программа?
8. Какие два этапа включает в себя создание программы на VB?
9. Перечислите типы переменных, используемые в Visual Basic.
10. Что такое пользовательский тип?
11. Для чего объявляют переменные?
12. Как работает оператор присваивания?
13. Назовите правила записи арифметических выражений.
14. Как записываются простые и сложные логические выражения?
15. Сформулируйте правила записи оператора IF.
16. Что такое вложенный оператор IF?
17. Какие операторы используются для программирования циклических процессов?
18. В чем различие модулей-процедур и модулей-функций?
19. Что такое массив? Как описываются массивы?
20. Что значит отладить программу?

Литература

1. Браун С. Visual Basic 6: Учебный курс. — СПб.: Питер, 2001.
2. Гарнаев А.Ю. Использование MS Excel и VBA в экономике и финансах. — СПб.: БХВ, 1999. — С.170–206.
3. Гарнаев А.Ю. Самоучитель VBA. — СПб.: БХВ, 1999.
4. Гуревич Н., Гуревич О. Visual Basic 5: Освой самостоятельно. — М.: БИНОМ, 1998.
5. Информатика: Учебник / Под ред. проф. Н.В.Макаровой. — М.: Финансы и статистика, 1998. — С.662–665.
6. Мальшев С.А. Самоучитель VBA. Как это делается в Word, Excel, Access. — СПб.: Наука и техника, 2001.
7. Мельников П.П. и др. Практикум по экономической информатике: Учебное пособие: Ч.III. — М.: Финансы и статистика; Перспективы, 2002.
8. Основы алгоритмизации и языки программирования: Сборник лабораторных работ в среде Visual Basic. Ч.1, 2, 3. — Новосибирск: НГАЭиУ, 2000.
9. Степанова Т.И. Основы алгоритмизации и программирования: Учебное пособие. — Новосибирск: НГАЭиУ, 2002.

ТЕМА 2.3. VISUAL BASIC FOR APPLICATION

2.3.1. VISUAL BASIC В ПРИЛОЖЕНИЯХ WINDOWS

Система программирования Visual Basic for Application (VBA) занимает важное место в стратегии программных продуктов фирмы Microsoft.

Visual Basic for Application — это подмножество Visual Basic, которое включает почти все его средства создания приложений, структуры данных и управляющие структуры, возможности создания пользовательских типов данных. В нем есть возможность создания форм со стандартным набором элементов управления и написания процедур, обрабатывающих события, которые возникают при тех или иных действиях системы и пользователя.

Visual Basic for Application — развитая система визуального программирования для создания прикладных программ в среде MS Office.

VBA включает в себя основные конструкции языка для Windows и встроен в приложения MS Office.

Применительно к среде MS Office употребляется термин «офисное программирование». В отличие от обычного программирования (создания программ) «офисное программирование» — это создание документов. Документ при этом становится первичным, и программа — лишь его часть. Изменилось само понятие «документ». При работе на VBA целью является создание документа в широком смысле (документа Word, рабочей книги Excel, презентации PowerPoint и т.д.).

Все создаваемые программные компоненты документа объединяются в одно целое, называемое *проектом*. Проект на VBA является частью документа, независимо от какого-либо документа его создать нельзя.

VBA позволяет работать с огромным набором объектов — по существу, в нем определены все объекты приложений MS Office. Для работы с VBA программист открывает одно из приложений MS Office, и в этот момент в языке VBA автоматически становится доступным объект Application, определяющий это приложение, и все встроенные в него объекты. Каждое приложение (Word, Excel, Access, PowerPoint и др.) дополняет средства VBA новыми функциональными возможностями с помощью объектов, свойств и методов, присущих этому приложению.

VBA отличается от Visual Basic и других языков программирования тем, что предоставляет возможность непосредственной работы с объектами MS Office. Это позволяет эффективно его использовать для автоматизации деятельности, связанной с обработкой различных типов документов.

Каждое из приложений, входящих в состав MS Office (Word, Excel, Access, PowerPoint), представляет собой совокупность объектов. Причем центральным (корневым) объектом является само приложение (например, Excel.Application). Все остальные объекты данного приложения

являются его элементами и встроены в него. Если какой-либо объект X встроен в приложение, то говорят, что у приложения есть свойство X, и обратиться к объекту X можно с помощью записи *Application.X*. В объект X могут быть также встроены другие объекты, т.е. образуется цепочка вложенности объектов. Например, в Excel корневым объектом является само приложение, в которое вложен объект *рабочая книга*, в рабочую книгу вложен объект *рабочий лист* и т.д. Иногда, чтобы добраться до нужного объекта, требуется указать несколько уровней вложенности.

Язык VBA позволяет создавать собственные классы объектов и работать с объектами остальных приложений MS Office.

С помощью VBA можно легко и быстро создавать различные приложения, даже не являясь профессиональным программистом. VBA имеет графическую инструментальную среду, позволяющую создавать экранные формы и управляющие элементы. С его помощью можно создавать, например, свои собственные функции для Excel, которые будут вызываться с помощью мастера функций, создавать собственные меню, разрабатывать макросы и т.д.

С помощью VBA можно создавать объекты управления графического интерфейса пользователя, задавать и изменять свойства объектов, подключать к ним соответствующий программный код.

Методика программирования с использованием средств VBA сводится к следующему:

- создание объектов управления и контроля (диалоговые окна, пиктограммы, меню);
- разработка процедур, используемых при вызове объектов.

Прикладные программы на языке VBA оперируют со следующими понятиями:

- *объект управления и контроля* — экранные формы, графические элементы внутри форм, в том числе текстовые окна, линейки прокрутки, пиктограммы, окна-списки, командные кнопки и др.;
- *свойство* (параметр) — характеристика или атрибут объекта управления;
- *значение свойства*;
- *событие* — действие, которое распознается объектом управления;
- *метод доступа* — оператор, который всегда воздействует на объект;
- *процедуры*, которые делятся на событийные (запускаются при наступлении событий) и общие процедуры.

2.3.2. СОЗДАНИЕ МАКРОСОВ

Все приложения MS Office имеют макроязык и средство записи макросов.

Приложения MS Office предоставляют возможность сохранить определенную последовательность действий пользователя в виде макрокоманд и использовать их в дальнейшем как программу на макроязыке VBA.

Макрокоманда (macros, macro, instruction) — предложение языка, идентифицирующее набор простейших команд (инструкций).

Применительно к пакету Microsoft Office макрокоманда — это сохраняемая под определенным именем последовательность действий на уровне команд главного меню, программных инструкций на языке программирования Visual Basic.

Макросом называется файл, в котором хранится программа последовательности действий, заданная пользователем. Каждый макрос должен иметь уникальное *имя*. После вызова макроса записанная последовательность действий или команд будет в точности воспроизведена. С помощью макросов можно автоматизировать многие типовые технологические этапы при работе с документами, например, макрос, выполняющий последовательность команд по созданию стиля абзаца документа, открытие файла, вводу информации в таблицы и т.д.

При записи макроса создается код, который можно просмотреть и отредактировать. Макрос можно создать двумя способами:

- запись макроса при помощи *макрорекодера*, с последующим его редактированием;
- создание макроса в редакторе VBA — языковые макрокоманды.

Чтобы записать макрос при помощи макрорекодера, необходимо, загрузив одно из приложений MS Office, выполнить команду *Сервис/Макрос/Начать запись*. В появившемся окне *Запись макроса* ввести имя макроса, при желании для быстрого вызова макроса поставить

создаваемому макросу в соответствие свою кнопку на панели инструментов или «горячую клавишу» и начать запись макроса. Далее все действия пользователя будут записываться в виде команд Visual Basic. Причем когда идет запись макрокоманды, действия, производимые с помощью мыши, не записываются, поэтому все эти действия нужно производить при помощи клавиатуры (в этом случае говорят о клавишных макрокомандах — последовательности нажатий клавиш, сохраненных под указанным именем). Необходимо отметить, что при использовании макросов, записанных с помощью макрорекодера, выполнение операций происходит не всегда безупречно. Поэтому возникает потребность откорректировать текст макроса. Для корректировки макроса достаточно выбрать его имя в списке макросов окна команды *Макрос* и нажать кнопку *Изменить*. Если нужно проследить выполнение макроса шаг за шагом, используют пошаговый режим макроса (режим отладки).

Чтобы создать макрос в редакторе VBA, нужно вызвать команду *Сервис/Макрос/Создать*. В появившемся окне редактора Visual Basic вводится код программы, т.е. программируется последовательность команд, подлежащих выполнению. С помощью этого метода можно создавать макрокоманды, недоступные для записи макрорекодером.

Макрос может храниться в самом файле документа (т.е. текущей рабочей книге) или в отдельной рабочей книге.

Контрольные вопросы

1. Чем отличается VBA от Visual Basic?
2. Что такое макрокоманда?
3. Как осуществляется запись макроса при помощи макрорекодера?
4. В чем отличие создания макроса в редакторе VBA от создания при помощи макрорекодера?

Литература

1. *Гарнаев А.Ю.* Использование MS Excel и VBA в экономике и финансах. — СПб.: БХВ, 1999.
2. *Гарнаев А.Ю.* Самоучитель VBA. — СПб.: БХВ, 1999.
3. *Малышев С.А.* Самоучитель VBA. Как это делается в Word, Excel, Access. — СПб.: Наука и техника, 2001.
4. *Мельников П.П. и др.* Практикум по экономической информатике: Учебное пособие: Ч.III. — М.: Финансы и статистика; Перспективы, 2002.

ГЛОССАРИЙ

Абзац — часть текста, заканчивающаяся символом конец абзаца «->», который ставится после нажатия клавиши Enter.

Абоненты сети — объекты, генерирующие или потребляющие информацию в сети.

Адаптеры — модули, которые осуществляют управление дополнительными периферийными устройствами ЭВМ.

Алгоритм — точно определенная последовательность действий, которые нужно выполнить над исходной информацией, чтобы получить решение задачи.

Алгоритмический язык — определенный набор символов и специальных слов, которые в соответствии со строгими правилами записи команд (синтаксиса языка) описывают алгоритм решения задачи.

АЛУ — арифметико-логическое устройство, в котором выполняются арифметические и логические операции над данными, хранящимися в регистрах арифметического устройства.

Антивирусное средство — программный продукт или устройство, выполняющие одну либо несколько следующих функций:

- защиту данных файловой структуры от разрушения;
- обнаружение вирусов;
- нейтрализация вирусов.

Архивация — слияние нескольких файлов и даже каталогов в единый файл — архив.

Архиватор — программа, осуществляющая сжатие и упаковку данных.

Атрибут — см. реквизит.

Байт — совокупность из 8 битов, воспринимаемая компьютером как единое целое.

Бит — наименьшая единица информации в ЭВМ — двоичный разряд, принимающий значение 0 или 1.

Блок-схема — такое графическое представление алгоритма, когда отдельные действия (или команды) представляются в виде геометрических фигур — блоков.

Бод — число изменений состояния среды передачи в секунду.

Буфер обмена — некоторое динамическое пространство оперативной памяти для временного размещения данных обмена.

Верификация — доказательство того, что программа работает «правильно».

Видеопамять — память, в которую МП записывает изображение, а затем уже видеоадаптер выводит содержимое видеопамяти.

Гипертекст — текст, содержащий в себе связи с другими текстами, графической, видео- или звуковой информацией.

Графический образ — совокупность точек, линий и фигур, с помощью которых изображаются данные.

Данные — признаки или результаты наблюдений над объектами или явлениями, которые по каким-то причинам не используются, а только хранятся.

Драйвер — специальная программа, при помощи которой осуществляется управление внешними устройствами ПК.

Интерфейс пользователя — комплекс программ, реализующих диалог пользователя с компьютером на стадии как ввода, так и получения информации.

Информатизация общества — организованный процесс создания оптимальных условий для удовлетворения информационных потребностей граждан, организаций, органов власти на основе формирования и использования информационных ресурсов.

Информатика — комплексная научная и инженерная дисциплина, изучающая все аспекты разработки, проектирования, создания, оценки, функционирования основанных на ЭВМ систем переработки информации, их применения и воздействия на различные области социальной практики.

Информация — новые сведения, воспринятые, понятые и оцененные как полезные для решения тех или иных задач.

Информационная модель — совокупность информации об объекте или процессе.

Информационная система — взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.

Информационная система экономического объекта — совокупность средств и методов, обеспечивающих реализацию всего комплекса операций по обеспечению процесса управления необходимой информацией.

Информационная технология — процесс, использующий совокупность средств и методов обработки и передачи первичной информации для получения информации нового качества о состоянии объекта, процесса или явления.

Информационные коммуникации — пути и процессы, обеспечивающие передачу сообщений от источника информации к ее потребителю.

Информационные ресурсы — различные формализованные знания (теории, идеи, изобретения), данные (отдельные документы и массивы документов), технологии и средства их сбора, обработки, анализа, интерпретации и применения, а также обмена между источниками и потребителями информации.

Информационный процесс — процесс, в результате которого осуществляется прием, передача (обмен), преобразование и использование информации.

Клавиатура — устройство для ввода в компьютер текстовой и цифровой, а также некоторой управляющей информации.

Классификация — процесс разбиения всего множества объектов на подмножества на основе выбранных признаков и в соответствии с определенными правилами.

Кластер — минимальная единица пространства на магнитном диске, которое может быть отведено файлу.

Кодирование — процесс присвоения объектам кодовых обозначений.

Колонтитулы — дополнительная информация, которая помещается в верхнее или нижнее поле на каждой странице.

Компьютерная (вычислительная) сеть — совокупность компьютеров и терминалов, соединенных с помощью каналов связи в единую систему, удовлетворяющую требованиям распределенной обработки данных.

Компьютерный вирус — специально написанная программа, способная самопроизвольно присоединяться к другим программам, создавать свои копии и внедрять их в файлы, системные области компьютера и в вычислительные сети с целью нарушения работы программ, порчи файлов и каталогов, создания всевозможных помех в работе на ПК.

Консолидация — особый способ вычисления итогов.

Константа (англ. constant) — величина, значение которой постоянно и не изменяется при выполнении программы.

Курсор — мигающая черточка или прямоугольное пятнышко, стрелка и т.д., которое отмечает на экране место, куда попадет очередной введенный символ, указывает на программное окно, которое нужно активизировать и пр.

Кэш-память (так называемая сверхоперативная память) — предназначена для согласования скорости работы медленных устройств с более быстрыми.

Макрокоманда (macros, macro, instruction) — предложение языка, идентифицирующее набор простейших команд (инструкций).

Манипуляторы (координатно-указательные устройства) — устройства управления курсором и подачи некоторых команд.

Массив — это упорядоченный набор однотипных значений — элементов массива.

Математический сопроцессор — специальный блок для операций с плавающей запятой. Применяется для особо точных и сложных расчетов, а также для работы с рядом графических программ.

Материнская плата — модуль, в котором помещен микропроцессор (МП).

Метод — программа действий над объектом или его свойствами.

Микропроцессор — программно-управляемое (т.е. функционирует путем выполнения некоторой программы) электронное цифровое устройство, предназначенное для обработки информации, представленной в цифровом виде, и построенное на одной или нескольких БИС (большая интегральная схема), в которых сосредоточена сложнейшая логическая схема.

Модем — устройство, выполняющее модуляцию и демодуляцию информационных сигналов при передаче их из ЭВМ в канал связи и при приеме в ЭВМ из канала связи.

Модуль — самостоятельная часть программы, имеющая определенное назначение и обеспечивающая заданные функции обработки данных.

Модульность структуры ЭВМ — все электронное оборудование расчленено на модули (так называемые электронные платы), связанные между собой системной шиной.

Мультимедиа. Средства мультимедиа (multimedia — многосредовость) — комплекс аппаратных и программных средств, позволяющих человеку общаться с компьютером, используя самые разные, естественные для себя среды: звук, видео, графику, тексты, анимацию и др.

Мышь — устройство для управления перемещением курсора на экране дисплея, а также для ручного ввода графических данных.

Накопители — запоминающие устройства, предназначенные для длительного хранения информации, т.е. они являются энергонезависимыми.

Область видимости переменной — область программы, в которой может быть использована переменная.

Объект — совокупность свойств (параметров) определенных сущностей и методов их обработки.

Объект управления — за О.у. закрепляется выполнение планов, выработанных управленческим аппаратом, т.е. реализация той деятельности, для которой создавалась система управления.

Объектно-ориентированный интерфейс — управление ресурсами вычислительной системы посредством осуществления операций над объектами, представляющими файлы, каталоги (папки), дисководы, программы, документы и т.д.

ОЗУ (RAM — Random Access Memory — память с произвольным доступом) — оперативное запоминающее устройство, которое позволяет с большой скоростью записывать и считывать информацию, подготовленную для МП.

OLE-технология — дает возможность обмена данными, разработанными в различных программах и имеющими различный формат, и объединения их в составной документ.

Оператор — запись в программе на алгоритмическом языке действия, которое должно выполняться в программе.

Операционная система (ОС) — совокупность программных средств, осуществляющих управление ресурсами ЭВМ, запуск прикладных программ и их взаимодействие с внешними устройствами и другими программами, а также обеспечивающих диалог пользователя с компьютером.

Организационная диаграмма (или оргдиаграмма) — диаграмма, которая изображает иерархию некоторой организации, используя совокупность блоков и соединительных линий.

Открытая архитектура — архитектура, которая, во-первых, использует принцип взаимозаменяемости, а во-вторых, предоставляет возможность доукомплектования ПК, наращивания его мощности уже в ходе эксплуатации ПК.

Отладка программы — устранение ошибок программирования.

Пакет прикладных программ (ППП) — совокупность сложно организованных машинных программ, дополненная соответствующей технической документацией, для решения задач определенного класса конкретной предметной области.

Переменная (англ. variable) — некоторая величина, которая может изменяться, принимая в процессе выполнения программы различные значения.

Показатель — информационная совокупность наименьшего состава, достаточная для образования самостоятельного сообщения или формирования документа.

Пользовательский интерфейс — программные и аппаратные средства взаимодействия пользователя с программой или ЭВМ.

Порты — специальные разъемы, расположенные на тыльной стороне системного блока, через которые осуществляется связь компьютера с различными внешними устройствами.

Постоянное запоминающее устройство (ПЗУ) — используется для хранения и чтения неизменной информации, некоторых часто встречающихся величин, стандартных программ и т.п.

Программа — упорядоченная последовательность команд для ЭВМ, реализующая алгоритм решения какой-либо задачи.

Программный интерфейс — это совокупность средств, обеспечивающих взаимодействие устройств и программ в рамках вычислительной системы.

Путь к файлу — цепочка соподчиненных каталогов, которую необходимо пройти по иерархической структуре от корневого или текущего каталога до каталога, где зарегистрирован искомый файл, разделенных символом «\».

Рабочая книга — основное рабочее пространство ЭТ.

Рабочая станция — персональный компьютер, подключенный к сети. через который пользователь получает доступ к ее ресурсам.

Распределенная обработка данных — обработка данных, выполняемая на независимых, но связанных между собой компьютерах, представляющих распределенную систему.

Регистры — это электронное цифровое устройство для временного запоминания информации в форме двоичного числа или кода.

Резервирование файлов — создание копий файлов на машинных носителях информации и систематическое их обновление в случае изменения резервируемых файлов.

Реквизит (синонимы: слово, элемент данных, атрибут) — информационная совокупность, неделимая далее на более мелкие единицы.

Реквизиты-признаки — характеризуют качественные свойства отражаемых объектов.

Реквизиты-основания — количественные величины, характеризующие данный объект.

Ресурс — любой логический или физической компонент ЭВМ и предоставляемые им возможности.

Свойство — характеристика объекта, его параметр.

Сектор — участок дорожки МД, хранящий минимальную порцию информации, которая может быть считана с диска или записана на него.

Сервер — компьютер, подключенный к сети и обеспечивающий ее пользователей определенными услугами.

Сетевые операционные системы — комплекс программ, обеспечивающих обработку, передачу и хранение данных в сети. Сетевая ОС предоставляет пользователям различные виды сетевых служб (управление файлами, электронная почта, процессы управления сетью и др.).

Сжатие (упаковка) информации — процесс преобразования информации, хранящейся в файле, к виду, при котором уменьшается избыточность в ее представлении и соответственно требуется меньший объем памяти для хранения.

Система — любой объект, который одновременно рассматривается и как единое целое, и как совокупность разнородных объектов, объединенных для достижения определенного результата.

Система программного обеспечения — совокупность программ и правил вместе со всей связанной с этими компонентами документацией, позволяющие выполнять определенный круг задач как по организации работ самого компьютера, так и по решению конкретных задач.

Системная плата (модуль) — см. материнская плата.

Системная шина — основная интерфейсная система компьютера, обеспечивающая сопряжение и связь всех его устройств между собой.

Сканер — устройство для считывания в компьютер графической и текстовой информации.

Событие — изменение состояния объекта.

Спецификация файла — имя дисковод:путь\полное имя файла.

Стиль — набор форматирующих команд, сохраняемый под своим именем для многократного использования.

Станция — аппаратура, которая выполняет функции, связанные с передачей и приемом информации.

Субъект управления — управленческий аппарат, который формулирует цель, разрабатывает планы, требования, контролирует их выполнение.

Табличный процессор (синоним — электронная таблица) — пакет прикладных программ, обеспечивающий автоматизированную обработку информации, представленной в табличной форме.

Тактовая частота — показатель скорости работы процессора, измеряется в мегагерцах (1 МГц = 1 млн тактов в секунду).

Текстовые редакторы (процессоры) — специальные сервисные программы, входящие в состав прикладного программного обеспечения, предназначенные для подготовки, редактирования, оформления и печати документов и текстов различных видов с помощью ЭВМ.

Телеконференции — дискуссионные группы, входящие в состав Usenet.

Техническое обеспечение — комплекс технических средств, предназначенных для работы информационной системы, и соответствующая документация на эти средства.

Топология сети — усредненная геометрическая схема соединений узлов сети.

Трансляторы — специальные программы, которые переводят программы, написанные на языке высокого уровня, на язык машинных кодов.

Упаковка — см. сжатие.

Управленческая информация — информация, которая обслуживает процессы производства, распределения, обмена и потребления материальных благ и обеспечивает решение задач организационно-экономического управления народным хозяйством и его звеньями. Она представляет собой разнообразные сведения экономического, технологического, социального, юридического, демографического и другого содержания.

Файл — логически связанная совокупность данных, для размещения которой во внешней памяти выделяется именованная область.

Файловая система — часть операционной системы, управляющая размещением и доступом к файлам и каталогам на магнитных дисках.

Фильтрация списка — вывод на экран подмножества элементов списка, удовлетворяющих запросу пользователя.

Форматирование документа — формирование внешнего вида документа с помощью выбора различных шрифтов, установление размеров букв, интервалов, отступов и полей.

Форматирование диска — создание структуры записи информации на его поверхности: разметка дорожек, секторов, записи маркеров и другой служебной информации.

Форм-фактор — определяет тип разъема на материнской плате.

Цикл — последовательность действий, которая может выполняться многократно.

Шаблон документа — пустой документ с заданными свойствами, который «накладывается» на создаваемый документ или на основе которого строится новый документ.

Шаблон имени файла — специальная форма, в которой в полях имени и типа файла используются символы «?» и «*».

Шина — электрическое соединение или группа параллельных соединений, которые обеспечивают обмен информацией между компонентами компьютера.

Экономическая информатика — наука, изучающая методы автоматизированной обработки экономической информации с помощью средств вычислительной техники.

Экономическая информация — отражает социально-экономические процессы в производственной и непроизводственной сферах, во всех отраслях народного хозяйства, во всех органах и на всех уровнях управления.

Электронная вычислительная машина (ЭВМ) — комплекс технических средств, предназначенных для автоматической обработки информации в процессе решения вычислительных и информационных задач.

Электронная таблица — область экрана дисплея с сеткой, которая делит ее на столбцы и строки, или см. табличный процессор.

Ярлыки — специальные значки, которые обеспечивают быстрый доступ к наиболее часто используемым файлам, папкам, устройствам.