

2007  
123  
ВЫСШЕЕ ОБРАЗОВАНИЕ

# ИНФОРМАТИКА ДЛЯ ЭКОНОМИСТОВ



В.П. Агальцов  
В.М. Титов



В. П. Агальцов, В. М. Титов

# **ИНФОРМАТИКА ДЛЯ ЭКОНОМИСТОВ**

*Допущено Учебно-методическим объединением по образованию  
в области прикладной информатики в качестве учебника  
для студентов высших учебных заведений, обучающихся  
по специальности «Прикладная информатика (по областям)»  
и другим экономическим специальностям*

4420

Москва  
ИД «ФОРУМ» — ИНФРА-М  
2011

УДК 002  
ББК 32.81  
А23

*Рецензенты:*

доктор экономических наук, профессор кафедры  
«Макроэкономическая, международная статистика и национальное  
счетоводство» Московского государственного университета  
экономики, статистики и информатики (МЭСИ) *К. Г. Чобану*,  
доктор экономических наук, профессор, зав. кафедрой  
предпринимательства, управления и маркетинга Российского  
государственного социального университета *Н. Н. Пилипенко*

**Агальцов В. П., Титов В. М.**

А23 Информатика для экономистов: учебник. — М.: ИД «ФОРУМ»: ИНФРА-М, 2011. — 448 с.: ил. — (Высшее образование).

ISBN 978-5-8199-0274-5 (ИД «ФОРУМ»)

ISBN 978-5-16-002665-7 (ИНФРА-М)

Представлены тексты лекций по дисциплине «Информатика», прочитанных в Российском государственном социальном университете. Содержание учебника обеспечивает изучение основных разделов дисциплины «Информатика» в соответствии с требованиями государственных образовательных стандартов высшего профессионального образования 2000 г. ряда специальностей, в том числе: 060200 — Экономика труда; 060400 — Финансы и кредит; 060500 — Бухгалтерский учет, анализ и аудит; 060600 — Мировая экономика; 060800 — Экономика и управление на предприятии.

Изложены основные сведения, необходимые для освоения современных информационных технологий и эффективного использования их в профессиональной деятельности. Особое внимание уделено рассмотрению принципов, положенных в основу функционирования наиболее распространенных и сложных офисных программ — MS Word, MS Excel и MS Access. Все положения иллюстрированы примерами.

Для студентов вузов, также может быть полезна специалистам различного профиля, осваивающим информационные технологии.

УДК 002  
ББК 32.81

ISBN 978-5-8199-0274-5 (ИД «ФОРУМ»)

ISBN 978-5-16-002665-7 (ИНФРА-М)

© В. П. Агальцов, 2011

© В. М. Титов, 2011

© ИД «ФОРУМ», 2011

---

Подписано в печать 26.02.2006. Формат 60 × 90/16.

Печать офсетная. Гарнитура «Таймс». Усл. печ. л. 28,0. Уч.-изд. л. 28,8.

Бумага офсетная. Доп. тираж 1000 экз. Заказ № 3189.

Отпечатано с готовых диапозитивов в ОАО ордена «Знак Почета»

«Смоленская областная типография им. В. И. Смирнова».

214000, г. Смоленск, проспект им. Ю. Гагарина, 2.

# Введение

---

---

В наши дни информационные технологии нашли применение почти во всех сферах профессиональной деятельности. Это привело к тому, что умение эффективно использовать компьютер для решения прикладных задач является необходимым атрибутом профессиональной деятельности любого специалиста и во многом определяет уровень его востребованности в обществе.

Изучение информатики включено практически во все Государственные образовательные стандарты высшего профессионального образования. Однако изучение информатики осложняется тем, что студенты приходят в вуз с различным уровнем компьютерной подготовки, а большинство учебников по информатике ориентировано на подготовку технических специалистов.

Поэтому при подготовке данного учебника авторы стремились обеспечить систематическое изложение теоретической и прикладной информатики в соответствии с требованиями государственных стандартов высшего образования России, ориентируясь на конкретные программы обучения студентов экономических специальностей.

В основу учебника положен длительный опыт чтения курса лекций по информатике студентам социально-экономического факультета Российского государственного социального университета.

# Глава 1

## ИНФОРМАЦИЯ И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ

---

---

### 1.1. Основные понятия информатики

#### 1.1.1. Информатика и информация

Термин «информатика» вошел в лексикон специалистов во второй половине XX в. для обозначения области человеческой деятельности, предметом которой являлась автоматизированная обработка информации с помощью электронных вычислительных машин (ЭВМ). Академик А. П. Ершов, один из выдающихся исследователей этой области знания, во многом благодаря которому информатика стала школьной дисциплиной, писал так: «Информатика — это находящаяся в процессе становления наука, изучающая законы и методы накопления, передачи и обработки информации с помощью ЭВМ, а также (в переносном смысле) область человеческой деятельности, связанная с применением ЭВМ» [10].

Таким образом, предметом этой дисциплины является **информация**, а важнейшими ее задачами — исследование **видов и типов информации**, а также **информационных процессов**.

Термин «информация» происходит от латинского слова *informatio*, означающего *сведения, разъяснения, изложение*. Смысловое содержание термина «информация» разъясняется в зависимости от сферы, в которой он употребляется. В курсе данной учебной дисциплины условимся понимать его как сведения о лицах, предметах, фактах, событиях, явлениях и процессах [10].

### 1.1.2. Содержание дисциплины «Информатика»

В информатике, как и во многих других науках, можно выделить два основных направления: теоретическое и прикладное. Теоретическая информатика — математическая дисциплина. Она использует методы математики для построения и изучения моделей обработки, передачи и использования информации, создает тот теоретический фундамент, на котором строится все «здание» информатики.

Сама теоретическая информатика включает ряд научных дисциплин. Среди них можно назвать следующие: философские основы информатики, математические основы информатики, теорию информации, методы измерения информации, информационное моделирование, теорию алгоритмов, представление знаний и интеллектуально-информационные системы.

Теоретическая информатика — это математическая дисциплина. Она использует математические методы для построения и изучения моделей обработки, передачи информации, является тем теоретическим фундаментом, на котором базируются все понятия информатики. Данный раздел информатики включает ряд самостоятельных дисциплин. Краткая характеристика основных дисциплин теоретической информатики представлена ниже.

**Теория информации** изучает информацию как абстрактный объект, лишенный конкретного содержания, выявляя общие свойства информации. К теории информации в качестве одного из разделов можно отнести теорию кодирования, изучающую формы представления и преобразования информации. В теории информации представлен также раздел, занимающийся математическими вопросами передачи информации по каналам связи.

**Математическая логика** изучает методы, позволяющие анализировать процессы переработки информации на ЭВМ. С помощью моделей логического типа изучаются процессы, протекающие в компьютере при решении задач.

**Теория алгоритмов** используется для анализа, формальных преобразований, построения и логического вывода алгоритмов решения задач на ЭВМ.

**Программирование** рассматривает вопросы реализации алгоритмов средствами конкретного языка программирования с учетом всего жизненного цикла программного продукта, в том числе процесса перевода программ, представленных на некотором

языке программирования, в форму, воспринимаемую устройствами компьютера.

**Теория принятия решения** изучает общие схемы, используемые при выборе научного решения из множества альтернативных возможностей. Выбор может осуществляться в условиях конфликта. Модели таких процессов изучаются в теории игр.

**Анализ данных** базируется на теории вероятности, таких ее важных законах, как распределение вероятностей, биномиальное распределение, нормальное распределение.

**Искусственный интеллект** связан с раскрытием механизмов творческой деятельности людей, их способности к овладению навыками, знаниями и умениями. С точки зрения информатики искусственный интеллект — наука не чисто теоретическая. Она занимается и прикладными вопросами, связанными с построением реально действующих интеллектуальных систем (например, роботов) или разработкой экспертных систем.

В практической информатике можно выделить два крупных раздела: средства информатизации и информационные технологии. В свою очередь различают технические и программные средства информатизации. К первой группе относятся:

- средства хранения и обработки данных, к числу которых относятся персональные компьютеры, рабочие станции, вычислительные системы, устройства ввода-вывода, накопители (магнитные, оптические, смешанные);
- средства передачи данных. Здесь можно назвать сети ЭВМ, вычислительные комплексы, технические средства связи, телекоммуникационные системы передачи аудио-, видео- и мультимедийной информации.

Программные средства информатизации разделяются на системное программное обеспечение (ПО), системы программирования и прикладное ПО (программные средства реализации информационных технологий).

Под технологией понимается совокупность методов, способов и приемов, применяемых для получения определенного вида продукции. Эта совокупность обеспечивает гарантированное получение заранее заданного результата. Это свойство является важнейшей характеристикой технологии, отличающей его от других процессов, например эксперимента, где результат не может быть предопределенно предсказан, и т. п.

Информационные технологии относятся к области информационной деятельности людей. Книгоиздание, теле- и радиове-

щение, издание газет, библиотечное дело и многое другое создали собственные технологии. Любое обучение предполагает использование специальных приемов, т. е. технологий.

Разработка и совершенствование компьютерной техники привели к созданию новых информационных технологий, основанных на компьютерных методах обработки информации.

В настоящее время широкое распространение получили такие виды новых информационных технологий, как:

- подготовка документов;
- поиск информации;
- технологии, основанные на использовании автоматизированных систем управления (АСУ);
- автоматизированное проектирование;
- геоинформационные;
- обучения.

### **1.1.3. Информационные ресурсы и информатизация общества**

Появление и развитие информатики объективно обусловлено всем процессом эволюции человечества. Любое человеческое сообщество обладает определенными ресурсами, наличие которых обеспечивает не только его выживание, но и развитие.

Ресурс — запасы, источники чего-нибудь [9]. Известно несколько основных видов ресурсов, ставших уже классическими экономическими категориями (материальные ресурсы — совокупность предметов труда, предназначенных для использования в процессе производства общественного продукта, например сырье, материалы, топливо, энергия, полуфабрикаты, детали и т. д.:

- природные ресурсы — объекты, процессы, условия природы, используемые обществом для удовлетворения материальных и духовных потребностей людей;
- трудовые ресурсы — люди, обладающие общеобразовательными и профессиональными знаниями для работы в интересах общества;
- финансовые ресурсы — денежные средства, находящиеся в распоряжении государственной или коммерческой структуры;



- энергетические ресурсы — носители энергии, например уголь, нефть, нефтепродукты, газ, гидроэнергия, электроэнергия и т. д.

Наряду с этими традиционными видами ресурсов во второй половине XX в. появилось понятие «информационные ресурсы», толкование и обсуждение которого велось с того момента, когда начали говорить о переходе к информационному обществу. В настоящее время принято следующее определение этого понятия [15]:

**Информационные ресурсы** — отдельные документы и отдельные массивы документов, документы и массивы документов в информационных системах (библиотеках, архивах, фондах, банках данных, других информационных системах). Следовательно, под информационными ресурсами можно понимать знания, подготовленные гражданами для социального использования и зафиксированные на материальном носителе.

Тип и количество потребляемых ресурсов, уровень глубины их переработки и потребления человеком определялся уровнем развития производительных сил общества. В настоящее время некоторыми авторами выделяется три основных этапа развития человеческого общества: аграрный, индустриальный и информационный (постиндустриальный). В качестве показателя, по которому оценивается уровень развития общества, используется доля трудоспособного населения, занятого в той или иной сфере материального производства. Иллюстрацией данного тезиса может служить пример Соединенных Штатов Америки. В 1906 г. — это аграрное государство, в котором более 50 % населения занято в сельском хозяйстве. К началу второй мировой войны более половины населения занято в промышленном производстве — США стали индустриальным государством. В 70-х годах, после изобретения микропроцессорной технологии и создания персонального компьютера, появился термин «постиндустриальное общество». Сейчас общество, в котором большинство работоспособного населения занято производством, хранением, переработкой и реализацией информации, принято называть **информационным**.

Очевидно, что переход от одного этапа к другому требует длительного времени. Так, переход от аграрного общества к индустриальному — индустриализация — потребовал создания революционных изобретений, обеспечивающих этот процесс. К их числу можно отнести изобретение колеса, водяного, а затем и

парового двигателей, электричества, электрических машин, двигателя внутреннего сгорания и т. п.

Важнейшей предпосылкой, обусловившей необходимость информатизации, явился информационный кризис (взрыв), проявившийся во второй половине XX в. Суть его заключается в противоречии между возможностями человека по восприятию и переработке информации и все увеличивающимися потоками и объемами хранимой информации. Это объясняется тем, что общая сумма знаний, накопленных человечеством, менялась вначале очень медленно, но с 1900 г. она удваивается за 50 лет, с 1950 г. удвоение происходит каждые 10 лет, с 1970 г. — уже каждые 5 лет, а с 1990 г. — ежегодно [8]. Вследствие этого сложилась ситуация, когда в мире накоплен громадный информационный потенциал, но люди не могут им воспользоваться в полном объеме. Внедрение ЭВМ и телекоммуникаций в различные сферы деятельности послужило началом процесса информатизации общества, основными чертами которого могут считаться следующие.

1. Важнейшими продуктами социальной деятельности становятся информационные технологии, услуги и знания. Основными аспектами их использования являются актуализация, защита и обеспечение целостности информации.

2. За счет появления новых информационных технологий (НИТ) удастся решать новые задачи, решение которых ранее было либо невозможно, либо экономически нецелесообразно.

3. Каждый гражданин и учреждение в любое время может получить любую информацию, необходимую для их жизни и деятельности. Существуют лишь три запрета на получение информации, оформленные законодательно и связанные с обеспечением национальной безопасности, сохранением коммерческой тайны, неприкосновенностью частной жизни граждан.

4. Существует вся необходимая инфраструктура для информационных технологий: вычислительная техника, средства телекоммуникаций и связи, программные продукты, базы данных и знаний коллективного использования и др.

Для перехода к информационному этапу необходима информатизация общества. Этот процесс следует понимать как создание и развитие *информационной среды*: комплекса условий и факторов, обеспечивающих наилучшие условия использования информационных ресурсов с учетом автоматизированных способов их переработки и использования в целях социального прогресса. Другими словами, информатизация сводится к формированию инфор-

мационных технологий и созданию условий для эффективного их использования для блага человеческого общества. В качестве основных вех информатизации можно назвать появление устной речи, а затем письменности, книгопечатания, а затем ЭВМ.

Сущность процесса информатизации можно проиллюстрировать следующим примером (рис. 1.1).

Общество обладает соответствующим информационным потенциалом, который составляют информационные ресурсы в совокупности со средствами, методами и условиями, позволяющими их использовать.

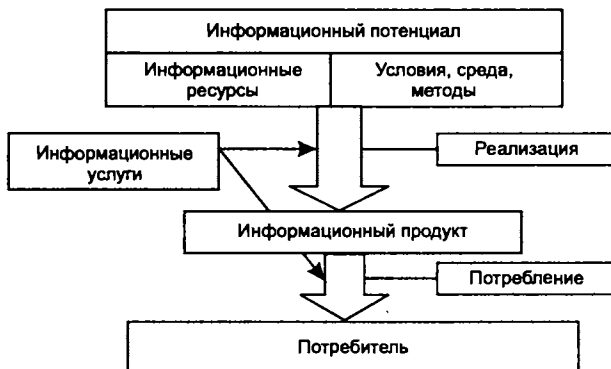


Рис. 1.1. Сущность процесса информатизации

Информационные ресурсы являются базой для создания информационных продуктов. При этом будем считать, что информационный продукт представляет собой совокупность данных, сформированных для распространения в вещественной или невещественной форме.

Такое понимание сущности процесса информатизации в полной мере согласуется с определением, приведенным в [15]: «Информатизация — организационный социально-экономический и научно-технический процесс создания оптимальных условий для удовлетворения информационных потребностей и реализации прав граждан, органов государственной власти, органов местного самоуправления, организаций, общественных объединений на основе формирования и использования информационных ресурсов».

Для распространения информационных продуктов (как и для распространения любого другого продукта) используется особая

форма деятельности — услуги. Если предположить, что услуга — это результат непродуцированной деятельности юридических или физических лиц, предназначенный для удовлетворения потребности общества в использовании различных продуктов, то можно считать, что информационная услуга — это получение и предоставление в распоряжение пользователя информационных продуктов. Эти продукты используются в самых различных сферах деятельности общества (рис. 1.2).

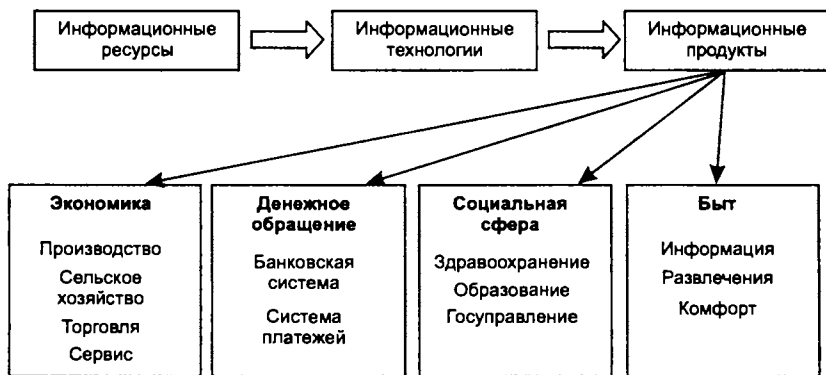


Рис. 1.2. Взаимосвязь понятий «ресурсы», «технологии», «продукты»

Следует обратить внимание на тот факт, что при определении понятия «информационный продукт» был использован новый термин «данные». Для того чтобы понять, каким образом данные связаны с информацией, как происходит преобразование информационных ресурсов в информационные продукты, следует рассмотреть существо информационных процессов.

## 1.2. Краткая характеристика информационных процессов

### 1.2.1. Классификация информации

Исходя из предпосылки о материальности окружающего нас мира, можно предположить, что любой материальный объект или процесс является носителем сведений о себе. Это обстоя-

тельство позволяет сделать вывод о том, что информация является неотъемлемым свойством всех объектов мироздания [5]. Многообразие информации вызывает потребность в ее классификации, для которой можно использовать несколько признаков:

А. По области возникновения:

1) элементарная информация — сведения, отражающие процессы и явления неживой природы;

2) биологическая информация — сведения, отражающие процессы и явления живой природы — животного и растительного мира;

3) социальная информация, сведения, отражающие процессы и явления человеческого общества.

Б. По способу восприятия:

1) с помощью органов чувств:

а) визуальная информация — сведения, воспринимаемые посредством видимых образов (символов) с помощью зрения;

б) аудиальная информация — сведения, воспринимаемые посредством звуков с помощью слуха;

в) тактильная информация — сведения, воспринимаемые посредством органов осязания;

г) органолептическая информация — сведения, воспринимаемые посредством запаха и вкуса с помощью органов обоняния и вкуса;

2) с помощью технических устройств.

В. По области человеческой деятельности:

1) научная;

2) техническая;

3) экономическая;

4) коммерческая;

5) управленческая;

6) общественно-политическая и т. д.

Г. По способу передачи:

1) с использованием «твердых носителей»;

2) с использованием средств электросвязи.

Существуют и другие признаки, в соответствии с которыми может осуществляться классификация информации. Например, по форме представления: текстовая, графическая или звуковая. Или же по общественному значению — личная, массовая или специальная, предназначенная для отдельных социальных групп.

Однако всегда следует помнить о том, что сведения об окружающем мире человек получает в результате наблюдений или измерений. Эти сведения отражаются в его сознании. Следовательно, можно предположить, что информация представляет собой отражение объектов или процессов окружающего мира в сознании человека. Это не что иное, как мысль. Очевидно, что если бы формирование знаний человека об окружающем мире происходило только на основе его собственного восприятия органами чувств, вряд ли прогресс человечества был бы столь значительным. Ряд достижений человечества, таких, как выход в космос и покорение морских глубин, овладение ядерной энергией и использование телекоммуникаций, создание государственных систем и систем искусственного интеллекта, находится за пределами чисто чувственного познания индивидуума. Это, несомненно, продукт многовековой коллективной интеллектуальной деятельности. Следовательно, прогресс человечества в целом и просто каждого человека в отдельности возможен только благодаря передаче знаний от одного человека другому. Возникает вопрос о том, как обеспечить возможность сохранения и передачи информации.

### 1.2.2. Информационные процессы

Очевидно, чтобы иметь возможность сохранить или передать сведения об объектах и явлениях внешнего мира, необходимо придать мысли материальную форму. Процессы изменения формы представления информации или ее содержания принято называть ее **обработкой**. Следовательно, данные — результат такого преобразования информации, при котором форма ее представления изменяется таким образом, что создается возможность хранить, передавать и обрабатывать информацию. Такая форма представления информации называется **данными** [10]. Процесс такого преобразования называется кодированием. При этом термин «код» служит для обозначения системы символов, применяемой для хранения, передачи и обработки информации.

Взаимосвязь объектов внешнего мира, информации и данных представлена на рис. 1.3. Из представленной схемы понятно, что когда человек считывает данные, в его сознании они воспринимаются как информация.



Рис. 1.3. Схема связи объектов внешнего мира, информации и данных

Совокупность последовательных действий, производимых над информацией, принято называть информационными процессами. В соответствии с [15] к числу информационных процессов относятся сбор, обработка, накопление, хранение, поиск и распространение информации. Рассмотрим подробнее эти процессы.

### 1.2.2.1. Сбор информации

Условимся понимать, что сбор информации представляет собой процесс ее сосредоточения в устройствах внешней памяти компьютера. Очевидно, что первым этапом этого процесса является получение информации от объектов внешнего мира. На этом этапе реализуется способность живых организмов отражать различные свойства объектов, явлений и процессов окружающего мира. Все живые существа, даже самые простейшие, воспринимают информацию об окружающей их среде и реагируют на нее в целях обеспечения условий для своего существования и продолжения рода. У высших животных и человека для получения информации об окружающей среде имеются органы чувств, через которые воспринимается информация определенного вида.

Необходимость обмена информацией для обеспечения сохранения вида привела к появлению и развитию средств общения между живыми существами. У человека одним из таких

средств общения является речь. В процессе общения объекты, их свойства и связи между ними, отраженные в сознании человека, обозначаются соответствующим образом. Такие обозначения составляют основу языка и называются именами. Следовательно, в сознании человека полученная информация существует в форме мысли, выраженной в образах или в терминах устной речи. Это объясняется тем, что естественные человеческие языки представляют собой системы кодирования понятий для выражения мыслей посредством речи [5].

Второй этап процесса сбора информации заключается в том, что ее необходимо из сознания человека перенести в память компьютера. Для этой цели служат различные прикладные программы и устройства ввода. Очевидно, что при этом слова или образы, существующие в сознании человека, нужно преобразовать в форму, удобную для обработки компьютером, т. е. закодировать ее.

В способах получения информации человек пошел намного дальше других живых существ — он создал приборы, позволяющие извлекать ту информацию, которая недоступна ему в непосредственных ощущениях. К числу таких приборов относятся электронный микроскоп, телескоп, термометры, приборы инфракрасного видения, приборы фиксации ультразвука, радиометры и т. д. Более того, созданы системы, позволяющие организовать автоматический (без участия человека) сбор информации. Например, ЭВМ может периодически «опрашивать» один или несколько входных источников информации, «считывая» с них нужные сведения.

В вычислительной технике изначально используется система двоичного кодирования, основанная на представлении данных последовательностью всего двух знаков: 0 и 1. Эти знаки называются двоичными цифрами, по-английски — binary digit или сокращенно bit (бит).

Одним битом могут быть выражены два понятия: 0 или 1 (*да* или *нет*, *черное* или *белое*, *истина* или *ложь* и т. п.). Если количество битов увеличить до двух, то уже можно выразить четыре различных понятия:

00 01 10 11

Тремя битами можно закодировать восемь различных значений:

000 001 010 011 100 101 110 111



Эти примеры показывают, что при увеличении количества разрядов в системе двоичного кодирования на единицу количество значений, которое может быть выражено в данной системе, увеличивается 2 раза, т. е. общая формула имеет вид:

$$N = 2^m,$$

где  $N$  — количество независимых кодируемых значений;  $m$  — разрядность двоичного кодирования, принятая в данной системе.

**Кодирование чисел.** Целые числа кодируются двоичным кодом достаточно просто — достаточно взять целое число и делить его пополам до тех пор, пока частное не будет равно единице. Совокупность остатков от каждого деления, записанная справа налево вместе с последним частным, и образует двоичный аналог десятичного числа.

Для кодирования целых чисел от 0 до 255 достаточно иметь 8 разрядов двоичного кода (8 бит). Шестнадцать бит позволяют закодировать целые числа от 0 до 65 535, а 24 бита — уже более 16,5 миллионов разных значений.

Для кодирования действительных чисел используют 80-разрядное кодирование. При этом число предварительно преобразуется в *нормализованную форму*:

$$3,1415926 = 0,31415926 \cdot 10^1;$$

$$300\ 000 = 0,3 \cdot 10^6;$$

$$123\ 456\ 789 = 0,123456789 \cdot 10^{10}.$$

Первая часть числа называется *мантиссой*, а вторая — *характеристикой*. Большую часть из 80 бит отводят для хранения мантиссы (вместе со знаком) и некоторое фиксированное количество разрядов — для хранения характеристики (тоже со знаком). Любое число можно представить в форме мантиссы с порядком.

**Кодирование текстовых данных.** Для кодирования текстовой информации с помощью двоичного кода можно путем постановки в соответствие каждому символу алфавита определенного целого числа (например, порядкового номера). В этом случае с помощью восьми двоичных разрядов можно обеспечить кодирование 256 различных символов. Это количество позволяет выразить различными комбинациями восьми битов все символы английского и русского языков, как строчные, так и прописные, а также

знаки препинания, символы основных арифметических действий и некоторые общепринятые специальные символы.

Наиболее широко для кодирования текстовой информации используется система кодирования *ASCII* (*American Standard Code for Information Interchange — стандартный код информационного обмена США*). В этой системе вводятся две таблицы кодирования — *базовая* и *расширенная*. Базовая таблица закрепляет значения кодов от 0-го до 127-го, а расширенная относится к символам с номерами от 128-го до 255-го.

Первые 32 кода базовой таблицы, начиная с нулевого, отданы производителям аппаратных средств. Начиная с 32-го по 127-й код размещены коды символов английского алфавита, знаков препинания, цифр, арифметических действий и некоторых вспомогательных символов.

Расширенная часть системы кодирования, определяющая значения кодов со 128-го по 255-й, используется для кодирования символов национальных алфавитов. Примеры таблиц кодирования приведены во многих источниках [9].

**Кодирование графических данных.** Графическое изображение, напечатанное в газете или книге, состоит из мельчайших точек, образующих характерный узор, называемый *растром*. Поскольку линейные координаты и индивидуальные свойства каждой точки (яркость) можно выразить с помощью целых чисел, то можно сказать, что растровое кодирование позволяет использовать двоичный код для представления графических данных. Общепринятым на сегодняшний день считается представление черно-белых иллюстраций в виде комбинации точек с 256 градациями серого цвета, и, таким образом, для кодирования яркости любой точки обычно достаточно восьмиразрядного двоичного числа.

Для кодирования цветных графических изображений применяется *принцип декомпозиции* произвольного цвета на основные составляющие. В качестве таких составляющих используют три основных цвета: красный (*Red, R*), зеленый (*Green, G*) и синий (*Blue, B*). На практике считается (хотя теоретически это не совсем так), что любой цвет, видимый человеческим глазом, можно получить путем механического смешения этих трех основных цветов. Такая система кодирования называется системой *RGB* по первым буквам названий основных цветов.

Если для кодирования яркости каждой из основных составляющих использовать по 256 значений (восемь двоичных разрядов), как это принято для полутоновых черно-белых изображе-

ний, то на кодирование цвета одной точки надо затратить 24 разряда. При этом система кодирования обеспечивает однозначное определение 16,5 миллионов различных цветов, что на самом деле близко к чувствительности человеческого глаза. Режим представления цветной графики с использованием 24 двоичных разрядов называется *полноцветным (True Color)*.

**Кодирование звуковой информации.** В настоящее время для записи звуков с помощью двоичных кодов используется таблично-волновой метод (Wave-Table). Суть его сводится к тому, что создаются специальные таблицы, в которых хранятся образцы звуков для множества различных музыкальных инструментов. В технике такие образцы называют сэмплами. Числовые коды выражают тип инструмента, номер его модели, высоту тона, продолжительность и интенсивность звука, динамику его изменения, некоторые параметры среды, в которой происходит звучание, а также прочие параметры, характеризующие особенности звука. Поскольку в качестве образцов используются «реальные» звуки, то качество звука, полученного в результате синтеза, получается очень высоким и приближается к качеству звучания реальных музыкальных инструментов.

**Единицы измерения данных.** Поскольку разные типы данных имеют универсальное двоичное представление, постольку в информатике используются единицы данных, основанные на нем. Так, наименьшей единицей измерения является бит. Однако, как было показано ранее, даже для кодирования одного текстового символа нужно 8 бит. Такая единица измерения получила название 1 байт.

Поскольку одним байтом, как правило, кодируется один символ текстовой информации, то для текстовых документов размер в байтах соответствует лексическому объему в символах. Более крупная единица измерения — килобайт (Кбайт). Условно можно считать, что 1 Кбайт примерно равен 1000 байт. Условность связана с тем, что для вычислительной техники, работающей с двоичными числами, более удобно представление чисел в виде степени двойки, и потому на самом деле 1 Кбайт равен  $2^{10}$  байт (1024 байт). Однако всюду, где это не принципиально, с инженерной погрешностью (до 3 %) «забывают» о «лишних» байтах.

В килобайтах измеряют сравнительно небольшие объемы данных. Условно можно считать, что одна страница неформатированного машинописного текста составляет около 2 Кбайт.

Более крупные единицы измерения данных образуются добавлением префиксов *мега-*, *гига-*, *тера-*; в более крупных единицах пока нет практической надобности.

1 Мбайт = 1024 Кбайт =  $10^{20}$  байт;

1 Гбайт = 1024 Мбайт =  $10^{30}$  байт;

1 Тбайт = 1024 Гбайт =  $10^{40}$  байт.

Наглядное представление о соотношении количества информации, содержащейся в сообщениях, передаваемых с помощью различных носителей, дает следующий пример. На диске объемом 100 Мбайт может быть записано:

страниц текста — 50 000;

цветных фотографий — 150;

аудиозапись речи — 1,5 часа;

музыкальный фрагмент качества CD-стерео — 10 минут;

фильм высокого качества записи — 15 секунд;

протоколы операций с банковским счетом — за 1000 лет.

Таким образом, можно предположить, что сбор информации включает непосредственно получение информации, ее двоичное кодирование и запись во внешнюю память компьютера.

### 1.2.2.2. Хранение информации

Под хранением информации (от *хранить* — содержать в безопасности/целости [12]) следует понимать содержание информации во внешней памяти компьютера.

С хранением информации связаны такие понятия, как носитель информации (память), внутренняя память, внешняя память, хранилище информации. Носитель информации — это физическая среда, непосредственно хранящая информацию. Основным носителем информации для человека является его собственная биологическая память (мозг человека). Ее можно назвать внутренней памятью, поскольку ее носитель — мозг — находится внутри человека. Все прочие виды носителей информации можно назвать внешними (по отношению к человеку). Виды этих носителей менялись со временем от камня до бумаги. Развитие информационной техники привело к созданию магнитных, оптических и других современных видов носителей информации.

Хранилище информации — это определенным образом организованная совокупность данных на внешних носителях, предназначенная для длительного хранения и постоянного использо-

вания. Примерами хранилищ являются архивы документов, библиотеки, справочники, картотеки. Основной информационной единицей хранилища является определенный физический документ — анкета, книга, дело, досье, отчет и пр. Под организацией хранилища понимается наличие определенной структуры, т. е. упорядоченность, классификация хранимых документов. Такая организация необходима для удобства ведения хранилища: пополнения его новыми документами, удаления ненужных документов, поиска информации и пр.

Знания, хранящиеся в памяти человека, можно рассматривать как внутреннее хранилище информации, однако организацию этого хранилища нам понять трудно. Основное свойство человеческой памяти — большая скорость воспроизведения хранящейся в ней информации. Но по сравнению с внешними хранилищами человеческая память менее надежна. Поэтому для более надежного хранения человек использует внешние носители, организует хранилища информации.

Основные свойства хранилища информации — объем хранимой информации, надежность хранения, время доступа (т. е. время поиска нужных сведений), наличие защиты информации.

Информацию, хранимую на устройствах компьютерной памяти, принято называть данными. Организованные хранилища данных на устройствах внешней памяти компьютера принято называть базами данных.

В современных компьютерах основными носителями информации для внешней памяти служат магнитные и оптические диски. Рассмотрим, каким образом организуется хранение информации на магнитных дисках. Такой подход оправдан тем, что оптические диски для этой цели стали использоваться гораздо позже, а потому ради обеспечения совместимости с магнитными устройствами они во многом имитируют структуру последних.

**Размещение информации на носителях.** Обычный магнитный диск имеет две поверхности, пригодные для размещения информации, которые в технической литературе принято называть сторонами (side) диска. Учитывая, что в накопителях на жестких дисках на одной оси могут устанавливаться несколько дисковых пластин, общее количество сторон может быть и больше [8].

Каждую поверхность обслуживает собственная магнитная головка (head). Все головки собраны в единый механический блок и могут двигаться вдоль радиуса дисков, причем движение это является дискретным, т. е. головки занимают относительно дис-

ка только строго определенные положения. Наконец, каждая дорожка разделена на отдельные *секторы* (sector) (рис. 1.4). Сектор является неделимой порцией информации и может быть прочитан только целиком. Последней координатой информации на диске служит номер требуемого байта в секторе.

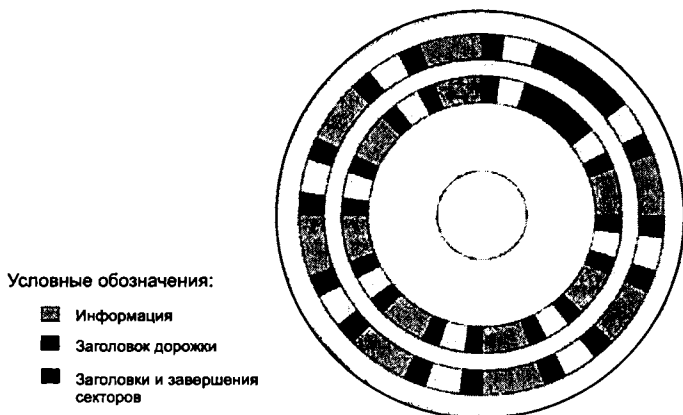


Рис. 1.4. Форматирование магнитного диска

Итак, положение байта информации на магнитном диске определяется четырьмя «координатами»: номером стороны, номером дорожки диска, номером сектора и номером байта в нем. Такая система хранения данных сложна и требует определенных усилий по их извлечению. Поэтому созданы специальные программы, которые позволяют пользователю извлекать необходимые данные, не требуя от него знания всех этих координат.

**Единицы хранения данных.** При хранении данных решаются две проблемы: как сохранить данные в наиболее компактном виде и как обеспечить к ним удобный и быстрый доступ (если доступ не обеспечен, то это не хранение). Для обеспечения доступа необходимо, чтобы данные имели упорядоченную структуру, а при этом возникает необходимость дополнительно записывать адресные данные. Без них нельзя получить доступ к нужным элементам данных, входящих в структуру [8].

Поскольку адресные данные тоже имеют размер и тоже подлежат хранению, хранить данные в виде мелких единиц, таких, как байты, неудобно. Их неудобно хранить и в более крупных единицах (килобайтах, мегабайтах и т. п.), поскольку неполное

заполнение одной единицы хранения приводит к неэффективности хранения.

В качестве единицы хранения данных принят объект переменной длины, называемый файлом. Файл — это последовательность произвольного числа байтов, обладающая уникальным собственным именем. Обычно в отдельном файле хранят данные, относящиеся к одному типу. В этом случае тип данных определяет тип файла.

В определении файла особое внимание уделяется имени. Оно фактически несет в себе адресные данные, без которых данные, хранящиеся в файле, не станут информацией из-за отсутствия метода доступа к ним. Кроме функций, связанных с адресацией, имя файла может хранить и сведения о типе данных, заключенных в нем. Для автоматических средств работы с данными это важно, поскольку по имени файла они могут автоматически определить адекватный метод извлечения информации из файла. Очевидно, что имя файла должно быть уникальным, так как это обеспечивает однозначность доступа к данным.

**Понятие о файловой структуре.** Хранение файлов организуется в иерархической структуре, которая в данном случае называется файловой структурой. В качестве вершины структуры служит имя носителя, на котором сохраняются файлы. Далее файлы группируются в каталоги (папки), внутри которых могут быть созданы вложенные каталоги (папки). Путь доступа к файлу начинается с имени устройства и включает все имена каталогов (папок), через которые проходит. В качестве разделителя используется символ «\» (обратная косая черта).

Уникальность имени файла обеспечивается тем, что полным именем файла считается собственное имя файла вместе с путем доступа к нему. Понятно, что в этом случае на одном носителе не может быть двух файлов с тождественными полными именами.

Пример записи полного имени файла:

```
<имя носителя>\<имя каталога1 >\...\<<имя каталогаM>\  
<собственное имя файла>
```

Следует иметь в виду, что сектора с информацией одного файла совсем не обязательно располагаются по порядку в одном месте диска. При записи система активно использует свободные места, которые образуются при удалении ненужных файлов. В результате отдельные части файла вполне могут попасть в раз-

ные области диска, что будет заметно замедлять доступ к информации. Для устранения этого явления в состав операционных систем обычно входят специальные служебные программы дефрагментации файлов.

Такой подход к организации хранения данных исторически был обусловлен тем, что в качестве материального носителя данных использовались накопители на магнитных дисках. При этом поверхность диска, на которую осуществлялась запись данных, форматировалась: разбивалась на дорожки и сектора. Программы форматирования обеспечивали создание сектора объемом 512 байт. Таким образом, для записи данных, принадлежащих одному файлу, как правило, требуется нескольких секторов. На рисунке отчетливо видно, что сектора на внешних дорожках диска имеют больший размер, чем ближайшие к центру. Это дает основания считать, что данные дорожки должны записываться более надежно. Поэтому нулевая дорожка, которая служит для хранения наиболее важной системной информации, всегда размещается на внешнем кольце поверхности диска.

### 1.2.2.3. Обработка информации

Под обработкой информации (от обработать — сделать готовым для использования) будем понимать ее преобразование с целью подготовки к практическому использованию [8]. Иногда обработка информации определяется как оперирование данными по определенным правилам.

В процессе обработки информации всегда решается некоторая информационная задача, заключающаяся в получении итоговой информации на основании исходных данных. Процесс перехода от исходных данных к результату и представляет собой обработку информации. Субъект, осуществляющий обработку, является исполнителем обработки. Исполнитель может быть человеком, а может быть специальным техническим устройством, в том числе компьютером.

Обычно обработка информации — это целенаправленный процесс. Для успешного выполнения обработки информации исполнителю должен быть известен способ обработки, т. е. последовательность действий, которую нужно выполнить, чтобы достичь нужного результата. Описание такой последовательности действий в информатике принято называть алгоритмом обработки.



Обычно различают два типа ситуаций, связанных с обработкой информации.

Первый тип — обработка, связанная с получением нового содержания знаний. К этому типу обработки относится решение математических задач. Способ обработки, т. е. алгоритм решения задачи, определяется математическими формулами, которые известны исполнителю. К этому типу обработки информации относится решение различных задач путем применения логических рассуждений. Например, руководитель, анализируя обстановку, принимает решение по организации работы для достижения нужных результатов.

Второй тип — обработка, связанная с изменением формы, но не изменяющая содержания. К этому типу обработки информации относится, например, перевод текста с одного языка на другой. Изменяется форма, но должно сохраниться содержание. Важным видом обработки для информатики является кодирование. Кодирование — это преобразование информации в символьную форму, удобную для ее хранения, передачи, обработки. Кодирование активно используется в технических средствах работы с информацией (телеграф, радио, компьютеры).

К обработке информации относится структурирование данных. Структурирование связано с внесением определенного порядка, определенной организации в хранилище информации. Примерами структурирования могут служить расположение данных в алфавитном порядке, группировка по некоторым признакам классификации, использование табличного представления.

Еще один важный вид обработки информации — поиск. Задача поиска состоит в отборе нужной информации, удовлетворяющей определенным условиям поиска, в имеющемся хранилище информации. Алгоритм поиска зависит от способа организации информации. Если информация структурирована, то поиск осуществляется быстрее, можно построить оптимальный алгоритм.

Таким образом, в зависимости от цели при обработке информации может изменяться форма ее представления либо ее содержание. Процессы изменения формы представления информации часто сводятся к процессам ее кодирования и декодирования и проходят одновременно с процессами сбора и передачи информации. Процесс изменения содержания информации включает в себя такие процедуры, как численные расчеты, редактирование, упорядочивание, обобщение, систематизация и т. д. Если прави-

ла преобразования информации строго формализованы и имеет-ся алгоритм их реализации, то можно построить устройство для автоматизированной обработки информации.

#### 1.2.2.4. Передача информации

Возможность передачи знаний, информации — основа прогресса всего общества в целом и каждого человека в отдельности.

Обычно общая схема передачи информации представляется таким образом — рис. 1.5.



Рис. 1.5. Общая схема процесса передачи информации

Для передачи информации могут использоваться различные способы (разные каналы связи): с помощью курьера, пересылка по почте, доставка транспортными средствами, дистанционная передача по каналам связи. Очевидно, что дистанционная передача по каналам связи сокращает время передачи данных, но для ее осуществления необходимы специальные технические средства.

Таким образом, процесс передачи информации состоит в ее распространении с помощью средств связи [8]. Схема этого процесса может быть представлена с помощью рис. 1.6.

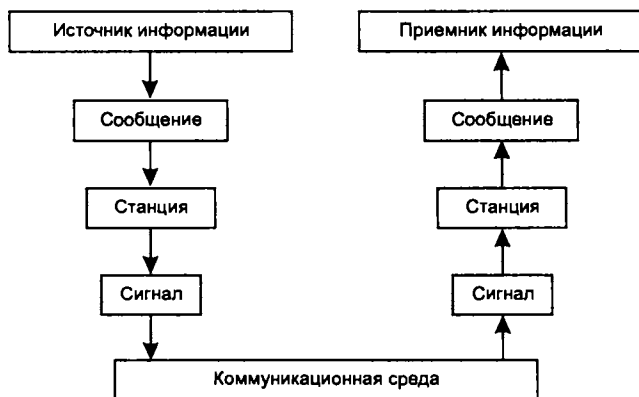


Рис. 1.6. Схема передачи информации

В качестве примера, наиболее наглядно поясняющего данную схему, рассмотрим процесс передачи информации по телеграфу. Сначала отправитель информации (источник) должен перенести ее на бланк телеграммы, т. е. подготовить сообщение. В информатике сообщением называются данные, предназначенные для распространения (передачи или приема) в коммуникационной среде. Этот бланк передается телеграфисту, который с помощью телеграфного аппарата преобразует сообщение в последовательность импульсов, которые распространяются по телеграфному проводу. С точки зрения теории связи эта последовательность импульсов представляет собой сигналы. Под термином «сигнал» условимся понимать физический процесс или явление, который способен обеспечить перенос информации в коммуникационной среде. В данном примере коммуникационную среду образуют аппаратура, соединяющая передающий и приемный телеграфные аппараты, а также линия связи между ними.

Таким образом, в процессе обмена информацией можно выделить ряд этапов. Сначала источник информации создает сообщение. Далее сообщение с помощью специальной аппаратуры передачи преобразуется в сигнал, который с помощью линии связи переносится в точку назначения. Там с помощью соответствующей аппаратуры сигнал принимается и восстанавливается до сообщения, которое и получает адресат.

При обозначении элементов системы передачи информации принято использовать следующие термины.

**Абоненты сети** — объекты, генерирующие или потребляющие информацию в сети. Абонентами могут быть отдельные ЭВМ, их комплексы, терминалы и т. п.

**Станция** — аппаратура, которая выполняет функции, связанные с передачей и приемом информации.

**Абонентская система** — совокупность абонента и станции.

**Коммуникационная среда** — аппаратура передачи данных и линии связи или пространство, в котором распространяются электрические сигналы.

Таким образом, обобщая сказанное, следует понимать следующее.

Информация — характеристика материи. Отражает свойства изменчивости окружающей среды, материи, вселенной. Информация выражается в форме любых сигналов (объяснение, представление). Философия определяет информацию как процесс отражения реальности. С инженерной точки зрения информа-

ция — любой набор данных, который является предметом хранения, передачи и преобразования.

Основные черты информации заключаются в том, что она:

- не материальна, но для своего использования требует наличия материального носителя;
- существует лишь для тех пользователей (приемников), которые способны ее воспринять;
- может быть представлена с помощью любых сигналов и знаков, которые могут содержать информацию сами по себе или в своем взаимном расположении.

### **Контрольные вопросы**

1. Дайте определение понятию «информация».
2. Предмет и задачи дисциплины Информатика.
3. Содержание дисциплины Информатика.
4. Сущность процесса информатизации.
5. Содержание базовых информационных процессов: сбора, хранения, обработки и передачи информации.

## **Глава 2**

# **ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ОБРАБОТКИ ДАННЫХ**

---

---

Итак, предметом дисциплины информатика является информация, а важнейшими ее задачами — исследование видов и типов информации, а также информационных процессов. Информация имеет две формы существования: нематериальную и материальную. Информация в нематериальной форме — это сведения об объектах, процессах и явлениях внешнего мира, отраженные в сознании человека. Для того чтобы иметь возможность сохранить ее, передать или обработать с помощью технических устройств, необходимо ее материализовать. Информация в материальной форме называется данными. Поэтому фактически предметом всех информационных процессов являются данные.

Рассмотрим некоторые аспекты процесса обработки данных. При этом в соответствии с [12] под термином «обработка» условимся понимать процесс преобразование информации (в форме данных) с целью ее подготовки к практическому использованию.

### **2.1. Данные и действия с ними**

#### **2.1.1. Классификация данных**

В процессе обработки проявляются различные свойства данных. Наиболее существенными характеристиками данных можно считать степень их изменения в процессе обработки, тип и способ организации.

### 2.1.1.1. Классы данных (константы и переменные)

С решением любой задачи связан определенный набор данных. Для подтверждения этого тезиса рассмотрим известный пример вычисления площади  $S$  треугольника с длинами сторон  $A$ ,  $B$ ,  $C$  по формуле Герона:

$$S = [p(p - a)(p - b)(p - c)]^{0,5},$$

где  $p = 0,5(a + b + c)$ .

Очевидно, что с решением этой задачи связаны следующие данные:  $a$ ,  $b$ ,  $c$ ,  $p$ ,  $S$  и  $0,5$ .

Данный пример показывает, что данные могут быть представлены двумя способами: с помощью конкретных значений ( $0,5$ ) и в обобщенном виде ( $a$ ,  $b$ ,  $c$ ,  $p$ ,  $S$ ).

Данные, представленные конкретными значениями, называются константами. Значения таких данных при решении задачи не меняются. Это означает, что в ячейке памяти, выделенной для хранения этого данного, все время хранится одно и то же значение, в нашем случае это число  $0,5$ . Примерами констант, часто используемых при решении задач, могут служить число  $\pi = 3,14\dots$ , величина ускорения свободного падения  $g = 9,81 \text{ м/с}^2$  и т. п.

Второй тип данных, представленных в нашем примере в обобщенном виде, получил название переменных. Переменная — это именованное данное, значение которого в памяти в процессе решения задачи или при повторных ее решениях может изменяться. Ячейка памяти для переменной снабжается именем (в нашем примере  $a$ ,  $b$ ,  $c$ ,  $p$ ,  $S$ ) и может в разные отрезки времени хранить разные значения (например,  $3,5$ ;  $7,8$  и т. п.).

Переменные различаются своей ролью в задаче. Одни из них отражают исходную, другие — промежуточную, третьи — результирующую информацию. С учетом этого различают три класса переменных:

- **исходные данные** (в нашем примере —  $a$ ,  $b$ ,  $c$ );
- **промежуточные данные** (в нашем примере —  $p$ );
- **результатирующие данные**, или результаты (в нашем примере —  $S$ ).

Значения исходных данных перед решением задачи вводятся в память компьютера пользователем посредством одного из устройств ввода (например, набираются на клавиатуре).

Значения промежуточных и результирующих данных формируются в памяти автоматически в процессе решения задачи. Зна-

чения результатов (а часто промежуточных и даже исходных данных) могут быть выведены пользователю посредством одного из устройств вывода (например, выведены на экран монитора или распечатаны на принтере).

### 2.1.1.2. Понятие о типах данных

Очевидно, что данными кодируются сведения о предметах, явлениях и процессах, различающихся своей природой. Поэтому обрабатываемые в ЭВМ данные различаются набором допустимых значений и видами операций, которые могут выполняться над ними. Эти две характеристики определяют тип данного. С этой точки зрения различают три типа данных: числовые, символьные и логические.

**Числовые данные** принимают числовые значения со знаком или без знака. Подразумевается, что число — это величина, с помощью которой осуществляется счет. Оно не может содержать алфавитных и специальных символов.

Допустимые над числовыми данными операции, реализуемые в ЭВМ, можно разделить на две группы (табл. 2.1).

Таблица 2.1. Операции, выполняемые над числовыми данными

Тип операций	Содержание операций	Символ операций	Результат выполнения
Арифметические	Сложение Вычитание Умножение Деление Возведение в степень	+ - * / ^	Число
Логические (сравнение двух величин)	Больше Больше или равно Меньше Меньше или равно Равно Не равно	> >= < <= = <>	Логическое значение TRUE (истина), если условие выполняется; FALSE (ложь), если условие не выполняется

**Символьные данные** принимают значения в виде последовательности любых символов. Иногда такие данные обозначают как данные текстового типа. Они используются для заголовков таблиц, заголовков строк и столбцов, имен данных, а также для комментариев. Обычно при записи алгоритмов и программ символьные данные выделяются с помощью кавычек.

Над символьными данными могут выполняться следующие операции:

- сравнения, «равно», в результате выполнения которых формируется значение TRUE или FALSE;
- сцепления двух символьных данных в одно (конкатенация).

**Логические данные** принимают одно из двух возможных значений: {TRUE, FALSE} или значения любого другого двухэлементного множества: {1, 0}, {ДА, НЕТ}, {ИСТИНА, ЛОЖЬ}, {T, F}. Логические данные являются результатом проверки истинности высказываний. Под высказыванием принято называть утверждение, о котором можно сказать, истинно оно или ложно. Например, истинность высказывания «*Экзаменационная сессия начнется 25-го июня*» может принять значение «да» (1) или «нет» (0). В зависимости от конкретной ситуации высказывание может быть истинным или ложным. Так, например, высказывание « $10 + 11 = 101$ » истинно, если числа записаны в двоичной системе счисления и ложно в остальных случаях.

### 2.1.1.3. Способы организации данных (объединения данных)

Данные описывают разные по сложности объекты, процессы и явления. Следовательно, для характеристики разных объектов может потребоваться различное число данных. По этой причине обработке могут подвергаться как одиночные переменные, так и различные их объединения. На практике чаще всего используются следующие объединения данных: массивы, записи, файлы.

**Массив** — это совокупность однородных (одного типа) переменных, объединенных общим именем.

Массивы бывают одномерными, двумерными и т. д. Аналогом одномерного массива в математике является вектор значений, аналогом двумерного массива — матрица значений. Доступ к элементам массива осуществляется посредством указания имени массива и соответствующих индексов — порядковых номеров элементов внутри массива. Примеры записи массивов:

Одномерный массив — *вектор*  $A = (a_1; a_2; a_3)$ .

Двухмерный массив — *матрица*  $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}$ .



**Запись** представляет собой совокупность переменных различного типа, объединенных общим свойством. Примерами записей могут служить строки в табл. 2.2, где зафиксированы данные о сотрудниках учреждения. Переменные, образующие запись, называют полями и могут обрабатываться независимо от других.

Таблица 2.2. Примеры записей данных

ФИО	Должность	Стаж работы, лет	Оклад, руб.
Попова О. А.	Менеджер	1	2600
Алексеева И. П.	Эксперт	4	2200
Кумарина Г. В.	Секретарь	2	2200

**Файл** представляет собой логически связанную совокупность данных, для размещения которой во внешней памяти компьютера выделяется поименованная область.

### 2.1.2. Обработка данных

**Выражения как средство обработки данных.** Знакомство с классами, типами и способами организации данных позволяет сделать вывод о том, что в результате обработки данных можно изменить не только форму их представления, но и содержание данных.

Процессы изменения формы представления информации часто сводятся к процессам кодирования, декодирования и структурирования данных и проходят одновременно с процессами сбора, передачи и хранения информации.

Процесс изменения содержания информации включает в себя такие процедуры, как численные расчеты, редактирование, логические рассуждения. При этом из констант и переменных соответствующего типа с помощью операций, допустимых для данного типа, создаются выражения. Таким образом, выражение представляет собой инструмент для обработки информации и используется для отображения этого процесса при разработке алгоритмов и программ.

Смысл выражения определяется типом входящих в него переменных. Например, выражение  $A + B$  — арифметическое, если  $A$  и  $B$  — числовые переменные типа, символьное (текстовое), если переменные  $A$  и  $B$  имеют тот же тип.

**Арифметические и логические выражения.** Наиболее простыми являются арифметические выражения. В этом случае значение выражения вычисляется по числовым исходным данным одного типа. Для обозначения числовых выражений, как правило, используется термин «формула». При вычислениях необходимо строго соблюдать следующие правила.

1. Вычисления выполняются в порядке приоритета операций:

- а) возведение в степень;
- б) умножение/деление;
- в) сложение/вычитание.

2. Операторы с одинаковым приоритетом выполняются слева направо.

3. Для изменения последовательности вычислений используются скобки. При этом выражения внутри скобок вычисляются в первую очередь.

Пример вычисления арифметических выражений приведен в табл. 2.3.

Таблица 2.3. Пример вычисления арифметических выражений

Формула	Результат
$=3*6+12/4-2$	19
$=3*6+12/(4-2)$	24
$=3*(6+12)/4-2$	11,5
$=(3*6+12)/4-2$	5,5
$=3*(6+12/(4-2))$	36

Логические выражения существенно отличаются от арифметических. Дело в том, что сравнивать между собой можно не только высказывания, но и значения двух выражений любого (но одного и того же) типа. При этом следует иметь в виду, что:

- символы операций сравнения для числовых переменных интерпретируются обычным образом;
- для символьных и строковых переменных знаки = и <> интерпретируются как буквальное совпадение или, соответственно, несовпадение значений этих переменных.

Из простых высказываний могут быть созданы составные. С этой целью используются логические операции (табл. 2.4).

Таблица 2.4. Логические операции

Логическая операция	Синонимы названия		Символ операции
	Дизъюнкция	Логическое сложение	
ИЛИ	Дизъюнкция	Логическое сложение	$\vee$
И	Конъюнкция	Логическое умножение	$\&$ или $\wedge$
НЕ	Инверсия	Отрицание	$\bar{\quad}$ (горизонтальная черта сверху) или $\neg$

При использовании логических операций необходимо руководствоваться следующими правилами:

1. Составное высказывание, полученное применением к любому простому высказыванию операции НЕ, истинно тогда и только тогда, когда ложно простое высказывание.

2. Составное высказывание, полученное применением к любым двум простым высказываниям операции ИЛИ, истинно тогда и только тогда, когда истинно хотя бы одно из простых высказываний.

Для вычисления значения сложных логических выражений служат таблицы истинности [13]. Они позволяют вычислить значение сложного логического выражения, если известны значения простых логических выражений, составляющих сложное. Эти таблицы приведены ниже (табл. 2.5, 2.6).

Таблица 2.5. Таблица истинности для операции И

$A$	$B$	$A \text{ И } B$
Истина	Истина	Истина
Истина	Ложь	Ложь
Ложь	Истина	Ложь
Ложь	Ложь	Ложь

Таблица 2.6. Таблица истинности для операций ИЛИ и НЕ

$A$	$B$	$A \text{ ИЛИ } B$	$A$	НЕ $A$
Истина	Истина	Истина	Истина	Ложь
Истина	Ложь	Истина	Ложь	Истина
Ложь	Истина	Истина		
Ложь	Ложь	Ложь		

Таблица 2.5 представляет результат вычисления значения сложного выражения  $A$  И  $B$ , когда известны значения простых логических выражений  $A$  и  $B$ , составляющих его.

Для логических операций так же, как и для арифметических, установлен строгий порядок выполнения. Самой старшей считается операция НЕ. Это означает, что в выражении, не содержащем скобок, она выполняется первой. Следующий приоритет присвоен операции И. Самый низкий приоритет принадлежит операции ИЛИ.

Для иллюстрации действия логических функций можно воспользоваться и логическими схемами (табл. 2.7). При этом 1 обозначает логическую переменную ИСТИНА, а 0 — ЛОЖЬ.

Таблица 2.7. Схемы логических операций

Конъюнкция	Дизъюнкция	Инверсия
		
		
		
		

### 2.1.3. Структуры и модели данных

**Структурирование данных.** Не вызывает сомнения, что при хранении большого количества данных неизбежно могут возникнуть проблемы с извлечением из этого множества только тех из них, которые необходимы для решения конкретной задачи. Само собой разумеется, что сделать это проще, если данные упорядочены. В подтверждение этого рассмотрим пример данных, содержащих некоторые сведения о студентах (рис. 2.1).

Личное дело № 16493, Сергеев Петр Михайлович, дата рождения 1 января 1976 г.; ЛУД № 16593, Петрова Анна Владимировна, дата рожд. 15 марта 1975 г.; № личн. дела 16693, д.р. 14.04.76, Анохин Андрей Борисович.

Рис. 2.1. Неструктурированные данные

Легко убедиться, что найти необходимые данные при такой организации их хранения непросто. Гораздо проще сделать это, если эти же данные упорядочить, например поместить каждый из элементов данных в отдельную ячейку таблицы так, как это показано на рис. 2.2.

№ личного дела	Фамилия	Имя	Отчество	Дата рождения
16493	Сергеев	Петр	Михайлович	01.01.76
16593	Петрова	Анна	Владимировна	15.03.75
16693	Анохин	Андрей	Борисович	14.04.76

Рис. 2.2. Структурированные данные

Введение соглашений (правил) о способах представления данных называется их структурированием. Очевидно, что вариантов таких соглашений может быть сколь угодно много. Способ представления данных определяет их структуру, т. е. их взаиморасположение и связь [10]. Наиболее распространенными являются три основных типа структур данных — иерархическая, сетевая и реляционная.

**Иерархическая структура данных.** Иерархический тип структуры предполагает расположение частей или элементов целого в порядке от высшего к низшему. Объекты, связанные иерархическими отношениями, образуют ориентированный граф (перевернутое дерево), вид которого представлен на рис. 2.3. К основным понятиям иерархической структуры относятся: уровень, элемент (узел), связь. Узел — это совокупность атрибутов данных, описывающих некоторый объект. На схеме иерархического дерева узлы представляются вершинами графа. Каждый узел на более низком уровне связан только с одним узлом, находящимся на более высоком уровне. Такой узел называется порожденным. Иерархическое дерево имеет только одну вершину (корень дерева), не подчиненную никакой другой вершине и находящуюся на самом верхнем (первом) уровне. Зависимые (порожденные) узлы находятся на втором, третьем и т. д. уровнях. Количество деревьев в базе данных определяется числом корневых записей. К каждой записи базы данных существует только один (иерархический) путь от корневой записи. Например, как видно из рис. 2.3, для записи С4 путь проходит через записи А и В3. В такой структуре связь имеет характер подчинения и направлена от исходного (родительского) узла к зависимому (порожденному).

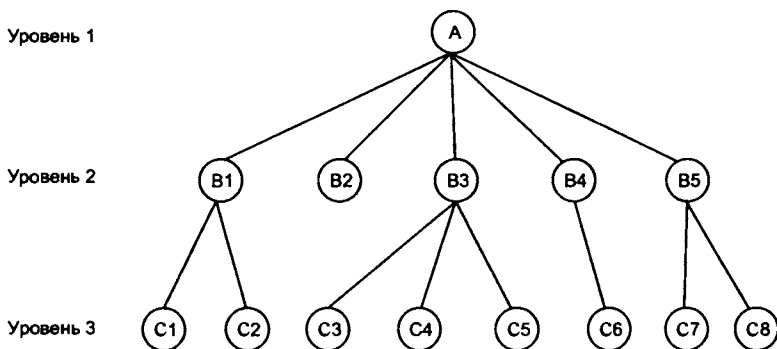


Рис. 2.3. Графическое изображение иерархической структуры БД

Таким образом, иерархическая структура данных, отличается тем, что адрес каждого элемента данных однозначно определяется путем доступа, ведущим от вершины структуры к данному элементу. Это обуславливает сложность записи адреса элемента данных.

Поиск данных в такой структуре осуществляется путем последовательного просмотра ветвей дерева. Такой процесс может быть достаточно длительным и трудоемким, что, несомненно, является недостатком такого структурирования. На практике задачу поиска можно упростить, если создать вспомогательную таблицу, связывающую элементы иерархической структуры с идентификаторами, облегчающими перемещение по ветвям дерева. Так, например, учебник представляет собой иерархическую структуру, уровнями которой являются разделы, главы и параграфы. Оглавление позволяет связать эти элементы структуры с номерами страниц, что позволяет читателю быстрее найти нужный фрагмент текста.

Несомненным достоинством иерархической структуры является простота ее обновления. Действительно, для этого достаточно ввести новую вершину на любом уровне. Свойства такой структуры обеспечили возможность ее применения для организации хранения данных во внешней памяти компьютера.

Пример иерархической структуры данных, описывающих гипотетический вуз, приведен на рис. 2.4. Предполагается, что данные об отдельных студентах располагаются на нижнем, шестом уровне. Применение иерархической структуры для рассматриваемого примера целесообразно потому, что каждый студент



Рис. 2.4. Пример иерархической структуры данных

учится в определенной (только одной) группе, которая относится к определенному (только одному) вузу. Для учета формы обучения (дневная, очно-заочная, заочная) можно ввести еще один уровень иерархии.

**Сетевая структура данных.** Если в древовидной структуре порожденные узлы имеют связи более чем с одним исходным, то такая структура называется сетевой. Другими словами, в сетевой структуре при тех же основных понятиях (уровень, узел, связь) каждый элемент может быть связан с любым другим элементом (рис. 2.5).

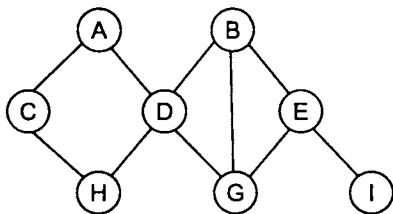


Рис. 2.5. Графическое изображение сетевой структуры

Примером сложной сетевой структуры может служить структура базы данных, содержащей сведения о дорожной сети какого-либо региона (рис. 2.6). Такая структура сложнее, чем иерархи-

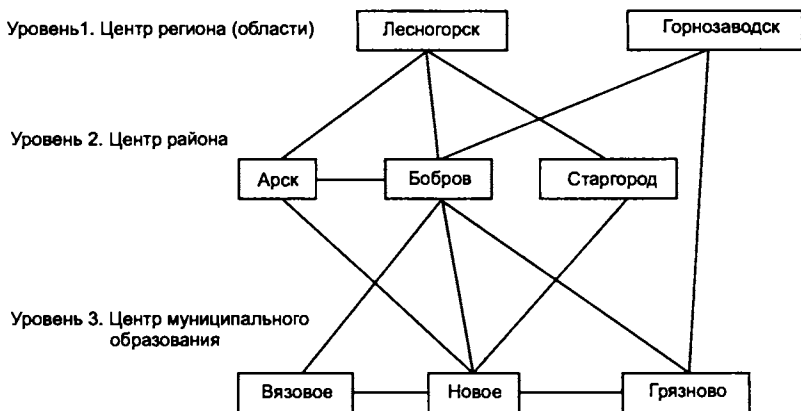


Рис. 2.6. Пример сетевой структуры базы данных

ческая. Следовательно, в ней труднее организовать поиск нужных данных. Реальным примером использования такой структуры на практике является структура глобальной информационной сети, которая получила название WWW (World Wide Web), или «всемирная паутина». Именно сложность поиска информации в этой структуре вызвали появление специальных средств поиска, таких, как «поисковые машины» и «каталоги».

**Реляционная структура данных.** Реляционные (от англ. *relation* — отношение) структуры данных отличаются простотой, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных. В этом случае данные организуются в виде двумерных таблиц. Каждая реляционная таблица представляет собой двумерный массив и обладает следующими свойствами:

- в ячейку таблицы помещается только один элемент данных;
- все столбцы в таблице однородные, т. е. все элементы в столбце имеют одинаковый тип (числовой, символьный и т. д.) и длину;
- каждый столбец имеет уникальное имя;
- одинаковые строки в таблице отсутствуют;
- порядок следования строк и столбцов может быть произвольным.

Примером реляционной таблицы может служить информация о студентах, обучающихся в вузе, представленная в табл. 2.8.



Таблица 2.8. Пример реляционной таблицы

№ личного дела	Фамилия	Имя	Отчество	Дата рождения	Группа
16493	Сергеев	Петр	Михайлович	01.01.76	111
16593	Петрова	Анна	Владимировна	15.03.75	112
16693	Анохин	Андрей	Борисович	14.04.76	111

Таблица отражает тип объекта реального мира (сущность), а каждая ее строка — конкретный объект. Строки таблицы представляют собой запись. Так, таблица «Студенты», представленная в нашем примере, содержит сведения обо всех студентах, а каждая ее строка — набор значений атрибутов конкретного студента. Значения конкретного атрибута выбираются из столбцов, в каждом из которых содержится множество всех возможных значений атрибута объекта. Имя столбца должно быть уникальным в таблице. Столбцы расположены в таблице в соответствии с порядком следования их имен при ее создании. Любая таблица содержит, по крайней мере, один столбец. В отличие от столбцов строки не имеют имен. Порядок следования строк в таблице не определен, а количество логически не ограничено. Так как строки в таблице не упорядочены, невозможно выбрать строку по ее позиции — среди них не существует «первой» и «последней».

Любая таблица имеет один или несколько столбцов, значения в которых однозначно идентифицируют каждую ее строку. Такой столбец (или комбинация столбцов) называется первичным ключом. В таблице «Студенты» первичным ключом служит столбец «№ личного дела». В таблице не должно быть строк, имеющих одно и то же значение первичного ключа. Если таблица удовлетворяет этому требованию, она называется отношением. Следовательно, отношение представляет собой сгруппированные в таблицу логически связанные данные, описывающие информационный объект.

Доступ к элементам данных в реляционной структуре осуществляется по адресу ячейки. Достоинством табличной структуры является простота адресации, а недостатком — сложность обновления. Последнее обстоятельство обусловлено тем, что при необходимости включения в такую структуру новых строк или столбцов изменяются адреса всех элементов данных, расположенных после заново включенных.

Особенности табличных структур данных определяют два класса задач, решаемых с их помощью. Первый из них — это вычислительные задачи. Применение табличных структур для обработки числовых данных восходит к раннему этапу развития математики. Так, еще за 2000 лет до н. э., в древнем Вавилоне, были широко распространены таблицы произведений натуральных чисел. В настоящее время таблицы широко применяются для решения широкого круга вычислительных задач, в том числе для численного решения уравнений, приближенного вычисления значений функций, интерполяции, численного интегрирования и др.

В то же время простота адресации данных, хранящихся в таблицах, делает удобным их использование для создания баз данных (БД). База данных — это поименованная совокупность структурированных данных, относящихся к определенной предметной области [10]. С помощью баз данных решаются задачи накопления и хранения данных с целью последующего выбора из всей БД только тех данных, которые отвечают заданным требованиям. Представляется возможным назвать такие задачи информационными.

## **2.2. Алгоритмы и программы обработки данных**

Как уже отмечалось, обработка информации представляет собой процесс перехода от исходных данных к результату, который осуществляется с целью решения определенной информационной задачи. Сущность этой задачи заключается в получении итоговой информации на основании исходных данных. Эту задачу решает исполнитель. Очевидно, для решения задачи необходимо, чтобы исполнителю была известна последовательность действий, приводящих к решению задачи.

### **2.2.1. Алгоритмы и формы их представления**

#### **2.2.1.1. Алгоритмы и их свойства**

Термин «алгоритм» (Algorithm) происходит от латинской транслитерации имени математика и астронома Абу Джафара ибн Мусы аль-Хорезми, жившего в IX в. Он разработал правила

выполнения арифметических действий. Эти правила при переводе трудов ученого на латинский язык в XII в. и были названы алгоритмами. Позже термин «алгоритм» стали использовать для обозначения любой последовательности действий, приводящей к решению какой-либо задачи. Таким образом, в широком смысле данный термин означает точное описание последовательности действий над заданными объектами, приводящее к достижению указанной цели за конечное число шагов.

При решении задачи на ЭВМ предполагается, что алгоритм — это описание метода решения задачи в виде последовательности доступных для ЭВМ операций, приводящих за конечное число шагов к получению требуемого результата. Результатом реализации алгоритма является некоторая информация или последовательность управляющих сигналов. Рассмотрим простой пример. Пусть нам необходимо из исходного состояния объекта **Документ**, содержанием которого является текст «информационная модель», получить конечное состояние, т. е. текст «модель информационная». Необходимую для реализации такого преобразования последовательность действий, т. е. алгоритм, запишем *на естественном языке*, который понятен пользователю компьютера:

- 1) выделить слово «информационная»;
- 2) вырезать этот фрагмент;
- 3) установить курсор на позицию после слова «модель»;
- 4) вставить фрагмент текста.

Предписание на выполнение одного шага (одного законченного действия) исполнителем называется командой алгоритма. Полный набор всех команд, которые могут быть выполнены конкретным исполнителем, составляет систему команд данного исполнителя.

Алгоритм решения задачи на ЭВМ должен обладать вполне определенными свойствами:

- дискретность алгоритма означает, что он выполняется по шагам: каждое действие, предусмотренное алгоритмом, выполняется только после того, как закончилось исполнение предыдущего;
- детерминированность (определенность) алгоритма. Данное свойство означает, что на каждом шаге однозначно определен способ действий;
- результативность алгоритма. Это свойство подразумевает, что каждый шаг (и алгоритм в целом) после своего завершения дает однозначно определенный результат;

- конечность алгоритма предполагает завершение его работы за конечное число шагов (при этом количество шагов может быть заранее неизвестным и различным для разных начальных ситуаций);
- понятность алгоритма заключается в том, что каждый шаг алгоритма обязательно представляет собой какое-либо допустимое действие исполнителя, т. е. алгоритм состоит только из предписаний, входящих в систему команд данного исполнителя;
- массовость алгоритма. Обычно это свойство означает возможность с помощью одного и того же алгоритма решать однотипные задачи.

Решения задачи осуществляется исполнителем. Схему этого процесса можно представить следующим образом (рис. 2.7).

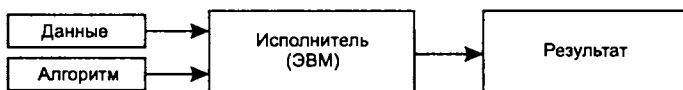


Рис. 2.7. Схема решения задачи на ЭВМ

Исполнителем алгоритма может быть любой субъект, способный распознать и выполнить предписание алгоритма. Таким субъектом может быть не только человек или группа людей, но и автомат, в том числе и компьютер. Наиболее существенным отличием компьютера от любого другого автомата является его универсальность: компьютеры работают по программе, причем, в зависимости от решаемой задачи, одна программа очень легко может быть заменена на другую.

Исполнители алгоритмов необычайно разнообразны, но есть общие черты, присущие всем исполнителям.

1. **Среда исполнителя.** Полный набор характеристик, описывающих состояние исполнителя, и обстановку, в которой он действует, принято называть средой данного исполнителя.

2. **Система команд исполнителя.** Предписание о выполнении отдельного законченного действия исполнителем называется командой алгоритма. Совокупность всех команд, которые могут быть выполнены некоторым исполнителем, образует систему команд данного исполнителя — СКИ.

3. **Режимы работы исполнителя.** Для большинства исполнителей предусмотрены два режима: непосредственного и программ-

ного управления. В первом случае исполнитель ожидает команд от человека и каждую поступившую команду немедленно выполняет. Во втором случае исполнителю сначала задается полная последовательность команд (программа), а затем он выполняет все эти команды в автоматическом режиме. Ряд исполнителей работает только в одном из названных режимов.

Разработка алгоритма — трудоемкая задача, требующая от человека глубоких знаний и больших затрат времени. Решение задачи по готовому алгоритму требует от исполнителя только строгого следования заданным предписаниям. Исполнитель действует формально, не вникая в смысл того, что он делает.

### 2.2.1.2. Формы записи алгоритмов

Для достижения необходимой точности и понятности алгоритм должен быть формализован, т. е. записан на одном из языков, понятных исполнителю. Наиболее часто для записи алгоритма применяется алгоритмический язык (АЯ). В качестве алфавита этого формального языка используются символы алфавита русского языка. Грамматическую основу АЯ составляют команды, обозначающие различные действия, а также набор служебных слов: *алг.*, *нач.*, *кон* и т. д. Предложения языка (алгоритмические конструкции) образуются на основе строгих правил синтаксиса, использующего логические и математические операции.

Очень часто алгоритм, записанный с помощью АЯ, трудно прочитать и понять. Чтобы сделать алгоритм более наглядным, часто используют *блок-схемы*. В этом случае различные элементы алгоритма изображаются с помощью различных геометрических фигур: для обозначения начала и конца алгоритма используются овалы, для последовательности команд — прямоугольники и т. д. (рис. 2.8).

Достоинство такой записи алгоритма состоит в том, что на блок-схеме хорошо видна структура алгоритма, и исполнителю (человеку) удобно отслеживать процесс его выполнения.

Любой алгоритм может быть построен из трех основных базовых структур:

- следование;
- ветвление;
- цикл.

Рассмотрим эти базовые структуры более подробно.


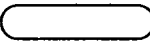
Наименование блока	Обозначение блока	Содержание операции
Процесс		Обработка данных
Принятие решения		Логический блок проверки условия
Передача данных		Ввод-вывод данных
Пуск, останов		Начало/конец программы
Модификация		Начало цикла

Рис. 2.8. Элементы блок-схем алгоритмов

Алгоритм следования соответствует линейному вычислительному процессу, когда операции следуют друг за другом. Блок-схема такой структуры представлена на рис. 2.9. Организация такого процесса требует, чтобы данные были представлены соответствующим образом. Например, для вычисления значения полинома

$$y = x^4 + ax^3 + bx^2 + cx + d \quad (2.1)$$

его целесообразно привести к виду

$$y = (((x + a)x + b)x + c)x + d. \quad (2.2)$$

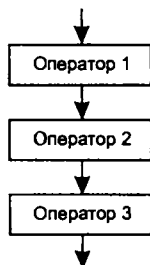


Рис. 2.9. Структура следования

Очевидно, что для вычисления значения полинома в данном случае команды (операторы) исполняются последовательно одна за другой. Блок-схема алгоритма вычисления данного полинома представлена на рис. 2.10.

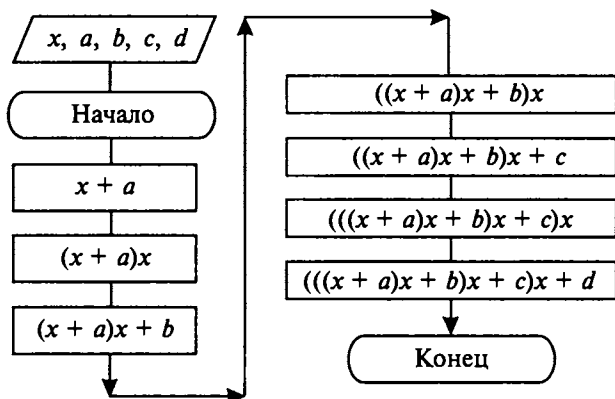


Рис. 2.10. Блок-схема вычисления полинома

Алгоритм ветвления (рис. 2.11) используется в тех случаях, когда исполнитель должен выбрать следующий шаг в зависимости от выполнения (или невыполнения) того или иного условия. Для формулирования условий используются высказывания — логические выражения, о которых можно однозначно утверждать истинно оно или ложно.

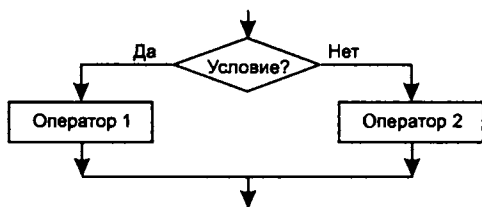


Рис. 2.11. Структура ветвления

Типичным примером разветвляющегося вычислительного процесса является вычисление корней квадратного уравнения

$$ax^2 + bx + c = 0. \quad (2.3)$$

Известно, что решение такого уравнения приводится к виду

$$x_{1,2} = -d \pm D^{(1/2)} \text{ при } D \geq 0 \quad (2.4)$$

или

$$x_{1,2} = -d \pm i|D|^{(1/2)} \text{ при } D < 0, \quad (2.5)$$

где  $d = b/(2a)$ ,  $D = (b/(2a))^2 - c/a$ .

В этом случае сначала вычисляются значения  $d$  и  $D$ , а затем проверяется выполнение условия  $D \geq 0$ . В случае истинности этого высказывания вычисляются действительные корни уравнения (формула (2.4)). Если высказывание ложно — комплексные (формула (2.5)). Блок-схема алгоритма вычисления корней квадратного уравнения приведена на рис. 2.12.

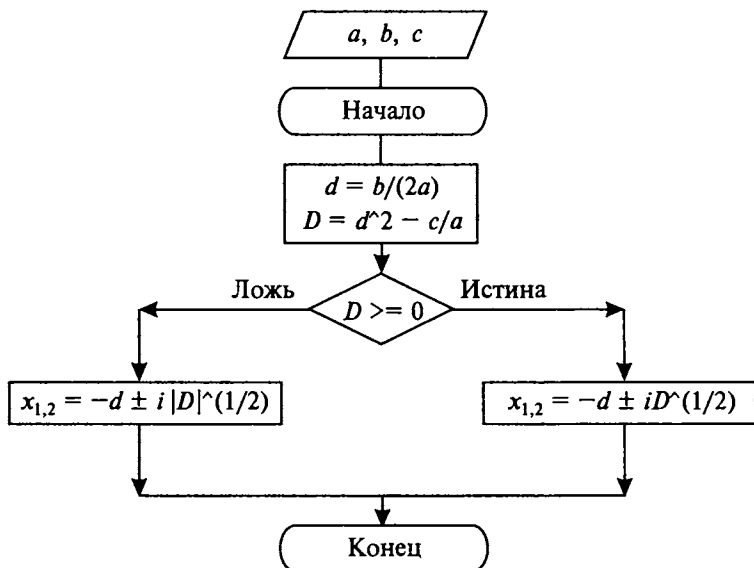


Рис. 2.12. Пример разветвляющегося вычислительного процесса

**Алгоритм цикла.** Циклические алгоритмы предусматривают многократное повторение одних и тех же операций над новыми исходными данными до тех пор, пока не будет выполнено заданное условие (рис. 2.13).

Циклический алгоритм позволяет вычислить ряд значений функции  $y_0, y_1, \dots, y_k$ , заданной уравнением (2.2) для последовательности значений аргумента  $x_1, x_2, \dots, x_k$ , заданных с шагом  $h$ . Вычисление каждого значения  $y_i$  производится так же, как и в случае линейного процесса, рассмотренного ранее. Затем на каждом цикле вычислений к аргументу  $x_i$  добавляется величина шага  $h$  и находится новое значение аргумента  $x_{i+1} = x_i + h$ , которое сравнивается с  $x_k$ . Если  $x_{i+1}$  окажется меньше  $x_k$ , то вновь вычисляется значение функции  $y_i$ . Процесс продолжается до тех пор,



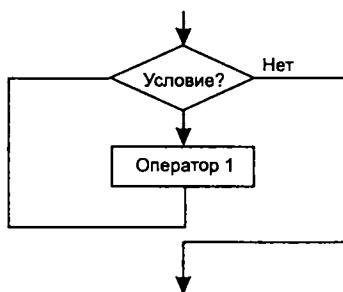


Рис. 2.13. Алгоритм цикла

пока очередное значение аргумента  $x_i$  не станет равным  $x_k$ . В результате будет вычислено  $k$  значений функции  $y$ . Блок-схема описанного алгоритма представлена на рис. 2.14.

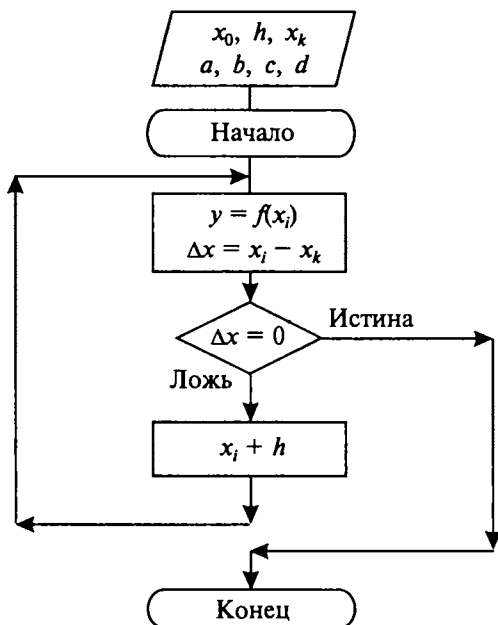


Рис. 2.14. Пример алгоритма циклического вычислительного процесса

Наглядное представление о достоинствах и недостатках рассмотренных форм записи алгоритмов дает пример решения задачи нахождения наибольшего общего делителя двух натуральных чисел (алгоритм Евклида).

А. Запись на алгоритмическом языке (справа представлен комментарий)

<b>алг</b> Евклида	Название алгоритма
<b>цел</b> $M, N$	Объявление переменных
<b>нач</b>	Объявление начала выполнения алгоритма
ввод $M$ и $N$	Ввод чисел $M$ и $N$
<b>пока</b> $M \neq N$ ,	Объявление цикла
<b>повторять</b>	
<b>Нц</b>	Объявления начала цикла
<b>Если</b> $M > N$ ,	Сравнение исходных значений $M$ и $N$
<b>То</b> $M := M - N$	Присвоение большему числу значения, равного разности исходных значений $M$ и $N$ ,
<b>Иначе</b> $N := N - M$	и повторение этого действия до тех пор, пока не сравняются оба значения
<b>кв</b>	
<b>Кц</b>	Объявление конца цикла
Вывод $M$	Вывод значения переменной после достижения равенства
<b>кон</b>	Конец алгоритма

Б. Блок-схема алгоритма Евклида (рис. 2.15).

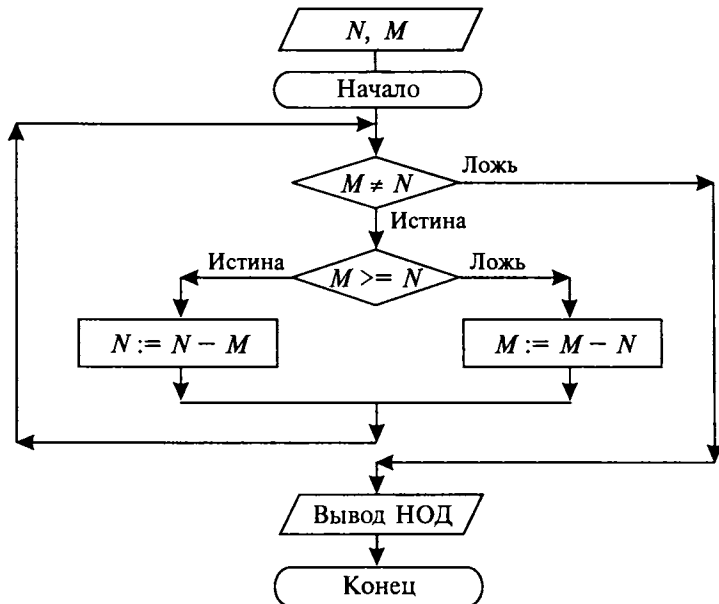


Рис. 2.15. Блок-схема вычисления наибольшего общего делителя (НОД)

Таким образом, существует несколько способов записи алгоритмов, отличающихся друг от друга наглядностью, компактностью, степенью формализации. Наибольшее распространение получили такие общеизвестные способы, как графический, словесный и в виде программ для ЭВМ. Графический способ записи алгоритмов предполагает использование определенных графических символов — блоков, каждый из которых предписывает выполнение определенных действий. Совокупность блоков образует так называемую схему алгоритма, или блок-схему. Данный способ записи алгоритма обладает высокой степенью наглядности и может оказаться полезным на начальных стадиях обучения программированию.

Словесная запись алгоритма ориентирована, прежде всего, на исполнителя — человека и допускает запись предписаний в свободной форме. При этом запись должна быть такова, чтобы исполнитель мог понять суть предписаний и механически (формально) их выполнить. Можно использовать так называемые алгоритмические языки, представляющие собой систему обозначений (кодировки) предписаний и правил их записи. Основу словаря любого алгоритмического языка составляют *ключевые слова*, смысл и способ употребления которых строго определен. Алгоритмические языки включают в себя и математическую символику: обозначения величин и функций, знаки операций, скобки и т. п. Словесный способ записи алгоритмов характеризуется высокой степенью понятности для исполнителя, но абсолютно недоступен пониманию компьютера.

## 2.2.2. Языки программирования

### 2.2.2.1. Сущность и содержание программирования

Являясь исполнителем программ, компьютер обладает своей системой команд. Команды управления работой компьютера называют *машинными командами*. Машинная команда содержит в себе следующую информацию:

- какую операцию выполнить;
- где находятся операнды;
- куда поместить результаты операции;
- где находится следующая операция.

Машинные операции достаточно универсальны: арифметические и логические операции, перепись информации, переходы, сдвиги кодов — вот практически весь их набор. Комбинируя машинные операции между собой, можно получить программу практически для любой предметной области.

Поэтому следующим шагом решения задачи обработки информации с помощью компьютера является программирование, которое представляет собой процесс преобразования алгоритма в программу. Следовательно, программа — это форма представления алгоритма для исполнения его машиной или набор команд, которые следует выполнить компьютеру для реализации того или иного алгоритма. Известно, что процессор компьютера способен воспринимать и исполнять команды только в машинных кодах. Поэтому первым программистам приходилось писать команды в «машинных кодах», т. е. именно так, как их воспринимает машина. Программирование в машинных кодах невероятно сложно и трудоемко. Например, команда сложения двух чисел могла выглядеть примерно так:

03 1138 4855 6773.

При этом следовало понимать значение символов, составляющих данную команду следующим образом:

03 — это код операции сложения, который процессор воспринимает как команду сложить два неких числа и их сумму записать в память;

1138, 4855 и 6773 — это номера ячеек памяти, причем первые два — номера ячеек, из которых надо взять слагаемые, а третий — в которую надо отправить сумму.

Очевидна большая трудоемкость процесса составления таких программ и высокая степень возможности ошибок при их написании. Особенно, если помнить, что эту команду необходимо перевести в двоичную форму.

Поиски путей устранения этих проблем привели к пониманию того, что процесс формирования машинного кода можно автоматизировать, если создать систему условных обозначений (язык программирования) для записи команд и составить внутреннюю программу, которая сама «переводила» бы такие команды в машинный код. Например, вместо указанной команды с кодом 03 можно написать:

$$a = m + n.$$

В этой записи с помощью символов латинского алфавита представлена вся информация, содержащаяся в рассмотренном примере.

При этом:

+ — оператор сложения и имеет смысл — произвести арифметическое сложение двух чисел  $m$  и  $n$ ;

= — оператор присвоения числу  $a$  значения, равного сумме чисел  $m$  и  $n$ ;

При такой записи следует иметь в виду, что символы латинского алфавита обозначают числовые переменные, значения которых хранятся в ячейках памяти. Перед исполнением команды машина должна сопоставить переменные с реальными номерами ячеек, в которые эти переменные должны быть помещены.

Подобным же образом можно записывать более сложные операции, которые должен исполнить компьютер. Например, команду проверки условия  $M \geq N$  в алгоритме Евклида, рассмотренном ранее, можно записать на языке, близком к одному из естественных:

Английский:

if  $M \geq N$  then  $M = M - N$ ;

Русский:

если  $M \geq N$ , то  $M = M - N$ .

Для реализации такого подхода необходимо:

1) создать строгую систему условных обозначений для записи команд в понятной для человека форме, т. е. язык программирования;

2) создать программу, предназначенную для преобразования команды, записанной на этом языке, на язык, понятный машине. Такая программа называется транслятором.

Языки программирования — искусственные языки. От естественных они отличаются ограниченным числом «слов», значение которых понятно транслятору, и очень строгими правилами записи команд (*операторов*). Совокупность подобных требований образует *синтаксис* языка программирования, а *смысл* каждой команды и других конструкций языка — его *семантику*. Нарушение формы записи программы приводит к тому, что транслятор не может понять назначение оператора и выдает сообщение о синтаксической ошибке, а правильно написанное, но не отвечающее алгоритму использование команд языка приводит к семантиче-

ским ошибкам (называемые еще логическими ошибками или ошибками времени выполнения).

Процесс поиска ошибок в программе называется *тестированием*, процесс устранения ошибок — *отладкой*.

Само собой разумеется, что с помощью языка программирования создается не готовая программа, а только ее текст, описывающий ранее разработанный алгоритм. Такой текст принято называть исходным модулем программы. Чтобы получить работающую программу, надо этот текст автоматически перевести в машинный код с помощью транслятора.

Применяются два метода использования трансляторов. Первый заключается в том, что исходный модуль автоматически переводится в машинный код, а затем функционирует отдельно от исходного текста. Трансляторы, работающие таким образом, получили название компиляторов. При реализации второго метода команды языка, указанные в тексте программы, выполняются поочередно сразу после перевода. По такой схеме работают *программы-интерпретаторы*. Различие в методах трансляции представлено на схеме (рис. 2.16).

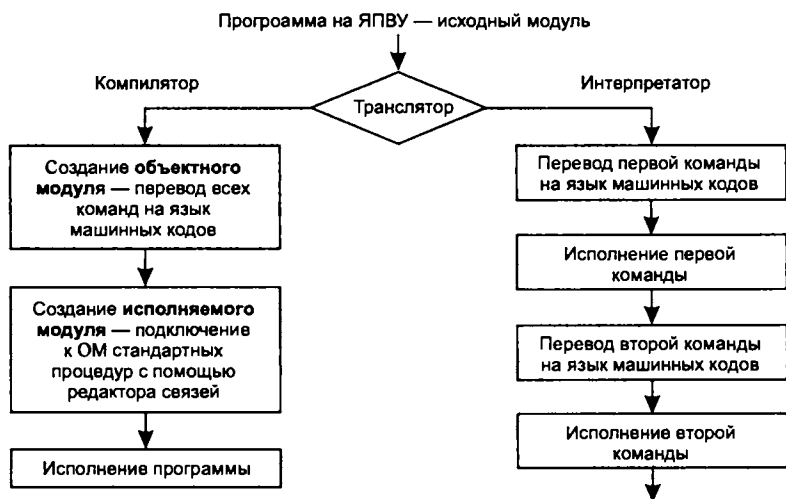


Рис. 2.16. Возможные способы трансляции программ

**Компиляторы** полностью обрабатывают весь исходный модуль программы. Они просматривают его в поисках синтаксических ошибок (иногда несколько раз), выполняют определенный

смысловой анализ и затем автоматически переводят (*транслируют*) на машинный язык — генерируют машинный код. Нередко при этом выполняется *оптимизация* с помощью набора методов, позволяющих повысить быстродействие программы (например, с помощью инструкций, ориентированных на конкретный процессор, путем исключения ненужных команд, промежуточных вычислений и т. д.). Другими словами при компиляции осуществляется два основных процесса:

- собственно перевод — превращение исходного модуля в объектный;
- подключение к нему с помощью специальной программы (редактора связей) стандартных процедур из библиотек, являющихся приложением к языку программирования.

В результате законченная программа (исполняемый модуль) получается компактной и эффективной, работает много быстрее программы, выполняемой с помощью интерпретатора, и может быть перенесена на другие компьютеры с процессором, поддерживающим соответствующий машинный код. Файлы таких программ имеют расширение `.EXE` и `.COM`.

Основной недостаток компиляторов — трудоемкость трансляции языков программирования, ориентированных на обработку данных сложной структуры, часто заранее неизвестной или динамически меняющейся во время работы программы. Тогда в машинный код приходится вставлять множество дополнительных проверок, анализировать наличие ресурсов операционной системы, динамически их захватывать и освобождать, формировать и обрабатывать в памяти компьютера сложные объекты, что на уровне жестко заданных машинных инструкций осуществить довольно трудно, а для ряда задач практически невозможно. К числу таких языков программирования относятся Паскаль, C++, Delphi, Fortran и др.

**Интерпретатор** берет очередной оператор языка из текста программы, анализирует его структуру и затем сразу исполняет (обычно после анализа оператор транслируется в некоторое промежуточное представление или даже машинный код для более эффективного дальнейшего исполнения). Только после того как текущий оператор успешно выполнен, интерпретатор перейдет к следующему. При этом, если один и тот же оператор должен выполняться в программе многократно, интерпретатор всякий раз будет выполнять его так, как будто встретил впервые. Вследствие этого, программы, в которых требуется осуществить большой

объем повторяющихся вычислений, могут работать медленно. Кроме того, для выполнения такой программы на другом компьютере также должен быть установлен интерпретатор — ведь без него текст программы является просто набором символов.

Достоинством интерпретатора является то, что с его помощью можно в любой момент остановить работу программы, исследовать содержимое памяти, организовать диалог с пользователем, выполнить сколь угодно сложные преобразования данных и при этом постоянно контролировать состояние окружающей программно-аппаратной среды, благодаря чему достигается высокая надежность работы. Интерпретатор при выполнении каждого оператора проверяет множество характеристик операционной системы и при необходимости максимально подробно информирует разработчика о возникающих проблемах. Кроме того, интерпретатор очень удобен для использования в качестве инструмента изучения программирования, так как позволяет понять принципы работы любого отдельного оператора языка.

В реальных системах программирования одновременно используются оба вида трансляции: и компиляции, и интерпретации. Так, в процессе отладки программа может выполняться по шагам, а результирующий код не обязательно будет машинным — он даже может быть исходным кодом, написанным на другом языке программирования. Использование такого приема существенно упрощает процесс трансляции, но требует компилятора для конечного языка.

По степени близости языка программирования к процессору или к программисту различаются его уровни (рис. 2.17).

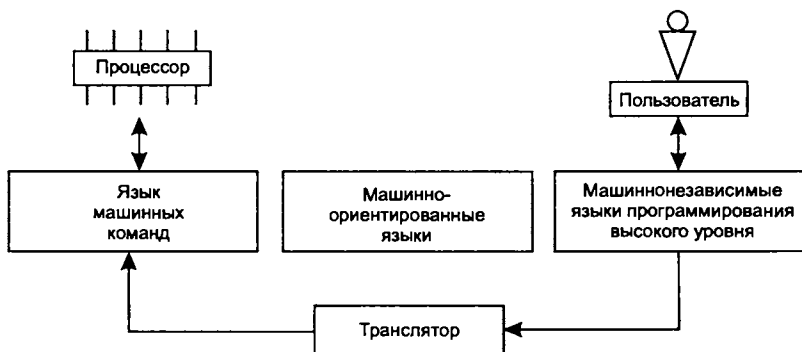


Рис. 2.17. Уровни языков программирования



### 2.2.2.2. Уровни языков программирования

Если язык программирования ориентирован на конкретный тип процессора и учитывает его особенности, то он называется *языком программирования низкого уровня*. В данном случае «низкий уровень» не значит «плохой». Имеется в виду, что операторы языка близки к машинному коду и ориентированы на конкретные команды процессора.

Языком самого низкого уровня является *язык ассемблера*, который просто представляет каждую команду машинного кода, но не в виде чисел, а с помощью символьных условных обозначений, называемых *мнемониками*. Однозначное преобразование одной машинной инструкции в одну команду ассемблера называется *транслитерацией*. Так как наборы инструкций для каждой модели процессора различаются, конкретной компьютерной архитектуре соответствует свой язык ассемблера, и написанная на нем программа может быть использована только в этой среде.

С помощью языков низкого уровня создаются очень эффективные и компактные программы, так как разработчик получает доступ ко всем возможностям процессора. С другой стороны, при этом требуется очень хорошо понимать устройство компьютера, затрудняется отладка больших приложений, а результирующая программа не может быть перенесена на компьютер с другим типом процессора. Подобные языки обычно применяют для написания небольших системных приложений, драйверов устройств, модулей стыковки с нестандартным оборудованием, когда важнейшими требованиями становятся компактность, быстрое действие и возможность прямого доступа к аппаратным ресурсам. В некоторых областях, например в машинной графике, на языке ассемблера пишутся библиотеки, эффективно реализующие требующие интенсивных вычислений алгоритмы обработки изображений.

*Языки программирования высокого уровня* значительно ближе и понятнее человеку, нежели компьютеру. Особенности конкретных компьютерных архитектур в них не учитываются, поэтому создаваемые программы на уровне исходных текстов легко переносимы на другие платформы, для которых создан транслятор этого языка. Разрабатывать программы на языках высокого уровня с помощью понятных и мощных команд значительно проще, а ошибок при создании программ допускает-ся гораздо меньше.

Языки программирования принято делить на пять *поколений*. В первое поколение входят языки, созданные в начале 50-х годов, когда первые компьютеры только появились на свет. Это был первый язык ассемблера, созданный по принципу «одна инструкция — одна строка».

Расцвет второго поколения языков программирования пришелся на конец 50-х — начало 60-х годов. Тогда был разработан символический ассемблер, в котором появилось понятие переменной. Он стал первым полноценным языком программирования. Благодаря его возникновению заметно возросли скорость разработки и надежность программ.

Появление третьего поколения языков программирования принято относить к 60-м годам. В это время родились универсальные языки высокого уровня, с их помощью удастся решать задачи из любых областей. Такие качества новых языков, как относительная простота, независимость от конкретного компьютера и возможность использования мощных синтаксических конструкций, позволили резко повысить производительность труда программистов. Понятная большинству пользователей структура этих языков привлекла к написанию небольших программ (как правило, инженерного или экономического характера) значительное число специалистов из некомпьютерных областей. Подавляющее большинство языков этого поколения успешно применяется и сегодня.

С начала 70-х годов по настоящее время продолжается период языков четвертого поколения. Эти языки предназначены для реализации крупных проектов, повышения их надежности и скорости создания. Они обычно ориентированы на специализированные области применения, где хороших результатов можно добиться, используя не универсальные, а проблемно-ориентированные языки, оперирующие конкретными понятиями узкой предметной области. Как правило, в эти языки встраиваются мощные операторы, позволяющие одной строкой описать такую функциональность, для реализации которой на языках младших поколений потребовались бы тысячи строк исходного кода.

Рождение языков пятого поколения произошло в середине 90-х годов. К ним относятся также системы автоматического создания прикладных программ с помощью визуальных средств разработки, без знания программирования. Главная идея, которая закладывается в эти языки, — возможность автоматического формирования результирующего текста на универсальных язы-

ках программирования (который потом требуется откомпилировать). Инструкции же вводятся в компьютер в максимально наглядном виде с помощью методов, наиболее удобных для человека, не знакомого с программированием.

### 2.2.2.3. Краткая характеристика языков программирования высокого уровня

Использование языков программирования высокого уровня для создания программ началось в 60-х годах. С тех пор и по настоящее время создано и используется множество различных языков программирования как универсальных, так и ориентированных на определенный круг задач.

Каждый язык программирования имеет свое название. Названия большинства языков программирования были зафиксированы при их первоначальном появлении. С тех пор правила, используемые в языках, могли существенно меняться, но название языка, как правило, осталось без изменения. Рассмотрим их краткие характеристики [9].

**Fortran (Фортран)** (formula translator). Это первый компилируемый язык, созданный Джимом Бэкусом в 50-е годы. Программисты, разрабатывавшие программы исключительно на ассемблере, выражали серьезное сомнение в возможности появления высокопроизводительного языка высокого уровня, поэтому основным критерием при разработке компиляторов Фортрана являлась эффективность исполняемого кода. Хотя в Фортране впервые был реализован ряд важнейших понятий программирования, удобство создания программ было принесено в жертву возможности получения эффективного машинного кода. Однако для этого языка было создано огромное количество библиотек начиная со статистических комплексов и кончая пакетами управления спутниками, поэтому Фортран продолжает активно использоваться во многих организациях.

**Cobol (Кобол)** (Common Business-Oriented Language). Это компилируемый язык для применения в экономической области и решения бизнес-задач, разработанный в начале 60-х годов. Он отличается большой «многословностью» — его операторы иногда выглядят как обычные английские фразы. В Коболе были реализованы очень мощные средства работы с большими объемами данных, хранящимися на различных внешних носителях. На этом языке создано очень много приложений, которые активно

эксплуатируются и сегодня. Достаточно сказать, что наибольшую зарплату в США получают программисты на Коболе.

**Algol (Алгол).** Компилируемый язык, созданный в 1960 г. Он был призван заменить Фортран, но из-за более сложной структуры не получил широкого распространения. В 1968 г. была создана версия Алгол 68, по своим возможностям и сегодня опережающая многие языки программирования, однако из-за отсутствия достаточно эффективных компьютеров для нее не удалось своевременно создать хорошие компиляторы.

**Pascal (Паскаль).** Язык Паскаль, созданный в конце 70-х годов основоположником множества идей современного программирования Никлаусом Виртом, во многом напоминает Алгол, но в нем ужесточен ряд требований к структуре программы и имеются возможности, позволяющие успешно применять его при создании крупных проектов.

**Basic (Бейсик)** (beginner's all purpose symbolic instruction code). Для этого языка имеются и компиляторы, и интерпретаторы, а по популярности он занимает первое место в мире. Он создавался в 60-х годах в качестве учебного языка и очень прост в изучении.

**C (Си).** Данный язык был создан в лаборатории Bell и первоначально не рассматривался как массовый. Он планировался для замены ассемблера, чтобы иметь возможность создавать столь же эффективные и компактные программы, и в то же время не зависеть от конкретного типа процессора.

Си во многом похож на Паскаль и имеет дополнительные средства для прямой работы с памятью. На этом языке в 70-е годы написано множество прикладных и системных программ и ряд известных операционных систем (Unix).

**C++ (Си++)**. Си++ — это объектно-ориентированное расширение языка Си, созданное Бьярном Страуструпом в 1980 г. Этот язык обладает множеством новых мощных возможностей, позволивших резко повысить производительность труда программистов. Поскольку он является языком невысокого уровня, постольку создание сложных и надежных программ на нем требует от разработчиков высокого уровня профессиональной подготовки.

**Java (Джава, Ява).** Этот язык был создан компанией Sun в начале 90-х годов на основе Си++ для того, чтобы упростить разработку приложений на основе Си++ путем исключения из него всех низкоуровневых возможностей.

Особое внимание в развитии этого языка уделяется двум направлениям: поддержке всевозможных мобильных устройств и

микрокомпьютеров, встраиваемых в бытовую технику (технология Jini) и созданию платформно-независимых программных модулей, способных работать на серверах в глобальных и локальных сетях с различными операционными системами (технология Java Beans). Пока основной недостаток этого языка — невысокое быстродействие, так как язык Ява интерпретируемый.

**PL/I (ПЛ/1).** В середине 60-х годов компания IBM решила взять все лучшее из языков Фортран, Кобол и Алгол. В результате в 1964 г. на свет появился новый компилируемый язык программирования, который получил название Programming Language One. По своим возможностям ПЛ/1 значительно мощнее многих других языков (Си, Паскаль).

**Smalltalk (Смолток).** Работа над этим языком началась в 1970 г. в исследовательской лаборатории корпорации XEROX, а закончились спустя 10 лет, воплотившись в окончательном варианте интерпретатора SMALLTALK-80. Данный язык оригинален тем, что его синтаксис очень компактен и базируется исключительно на понятии объекта. В этом языке отсутствуют операторы или данные. Все, что входит в Смолток, является объектами, а сами объекты общаются друг с другом исключительно с помощью сообщений

**LISP (Лисп).** Интерпретируемый язык программирования, созданный в 1960 г. Джоном Маккарти. Ориентирован на структуру данных в форме списка и позволяет организовывать эффективную обработку больших объемов текстовой информации.

**Prolog (Пролог).** Создан в начале 70-х годов. Программа на этом языке, в основу которого положена математическая модель теории исчисления предикатов, строится из последовательности фактов и правил, а затем формулируется утверждение, которое Пролог будет пытаться доказать с помощью введенных правил. Человек только описывает структуру задачи, а внутренний «мотор» Пролога сам ищет решение с помощью методов поиска и сопоставления.

Особое место среди языков программирования занимают языки программирования баз данных. Эта совокупность языков отличается от процедурных языков прежде всего тем, что они предназначены для работы с базами данных. База данных — это файл (или группа файлов), представляющий собой упорядоченный набор *записей*, имеющих единообразную структуру и организованных по единому шаблону (как правило, в табличном виде). База данных может состоять из нескольких таблиц. Удоб-

но хранить в базах данных различные сведения из справочников, картотек, журналов бухгалтерского учета и т. д.

При работе с базами данных чаще всего требуется выполнять следующие операции:

- создание/модификация свойств/удаление таблиц в базе данных;
- поиск, отбор, сортировка информации по запросам пользователей;
- добавление новых записей;
- модификация существующих записей;
- удаление существующих записей.

Первые базы данных появились очень давно, как только возникла потребность в обработке больших массивов информации и выборке групп записей по определенным признакам. Для этого был создан *структурированный язык запросов SQL* (Structured Query Language). Он основан на мощной математической теории и позволяет выполнять эффективную обработку баз данных, манипулируя не отдельными записями, а *группами* записей. Для управления большими базами данных и их эффективной обработки разработаны СУБД (Системы Управления Базами Данных). Практически в каждой СУБД помимо поддержки языка SQL имеется также свой уникальный язык, ориентированный на особенности этой СУБД и не переносимый на другие системы. Сегодня в мире насчитывается пять ведущих производителей СУБД: Microsoft (SQL Server), IBM (DB2), Oracle, Software AG (Adabas), Informix и Sybase. Их продукты нацелены на поддержку одновременной работы тысяч пользователей в сети, а базы данных могут храниться в распределенном виде на нескольких серверах.

С появлением персональных компьютеров были созданы так называемые настольные СУБД. Родоначальником современных языков программирования баз данных для ПК принято считать СУБД dBase II — язык, который был интерпретируемым. Затем для него были созданы компиляторы, появились СУБД FoxPro и Clipper, поддерживающие диалекты этого языка. Сегодня похожие, но несовместимые версии языков семейства dBase реализованы в продуктах Visual FoxPro фирмы Microsoft и Visual dBase фирмы Inprise. СУБД MS ACCESS включена в офисный пакет Microsoft Office.

С активным развитием глобальной сети было создано немало реализаций популярных языков программирования, адаптированных специально для Интернет. Все они отличаются харак-

терными особенностями: языки являются интерпретируемыми, интерпретаторы для них распространяются бесплатно, а сами программы — в исходных текстах. Такие языки называют *скрипт-языками*.

**HTML.** Общеизвестный язык для оформления документов. Он очень прост и содержит элементарные команды форматирования текста, добавления рисунков, задания шрифтов и цветов, организации ссылок и таблиц. Все Web-страницы написаны на языке HTML или используют его расширения.

**Perl.** Этот язык был разработан в 80-х годах. Он задумывался как средство эффективной обработки больших текстовых файлов, генерации текстовых отчетов и управления задачами. По мощности Perl значительно превосходит языки типа Си. В него введено много часто используемых функций работы со строками, массивами, всевозможные средства преобразования данных, управления процессами, работы с системной информацией и др.

**VRML.** В 1994 г. был создан язык VRML для организации виртуальных трехмерных интерфейсов в Интернете. Он позволяет описывать в текстовом виде различные трехмерные сцены, освещение и тени, текстуры (покрытия объектов), создавать свои миры, путешествовать по ним, «облетать» со всех сторон, вращать в любых направлениях, масштабировать, регулировать освещенность и т. д.

До появления ПК считалось, что на таких языках, как BASIC, FORTRAN, Pascal, можно писать программы, не учитывая, на какой машине они будут исполняться (т. е. декларировалась переносимость, независимость языка от аппаратной платформы).

После появления ПК стало невозможным говорить о языке вне его связи с конкретной реализацией на компьютере, вне связи со *средой* программирования. Среда программирования (или система программирования, СП) — это комплекс программных средств (их также называют инструментальными средствами), с помощью которых создается программа на одном из языков программирования.

Из универсальных языков программирования сегодня наиболее популярны следующие:

- Бейсик (Basic) — для освоения требует начальной подготовки (общеобразовательная школа);
- Паскаль (Pascal) — требует специальной подготовки (школы с углубленным изучением предмета и общетехнические вузы);

- Си++ (C++), Ява (Java) — требуют профессиональной подготовки (специализированные средние и высшие учебные заведения).

Для каждого из этих языков программирования сегодня имеется немало систем программирования, выпускаемых различными фирмами и ориентированных на различные модели ПК и операционные системы. Наиболее популярны следующие визуальные среды быстрого проектирования программ для Windows:

- Basic: Microsoft Visual Basic;
- Pascal: Borland Delphi;
- C++: Borland C++Builder.

Следует отметить, что с точки зрения пользователя все языки программирования высокого уровня обладают одним существенным недостатком — для их практического применения необходимы специальные знания. Поэтому особый интерес для нас представляют современные системы автоматического создания прикладных программ с помощью визуальных средств разработки, без профессионального знания программирования.

Как уже отмечалось, главная особенность таких систем состоит в том, что они предоставляют пользователю возможность вводить инструкции в компьютер в максимально наглядном виде с помощью методов, наиболее удобных для человека, не знакомого с программированием. К числу таких средств относятся языки программирования VBA и HTML.

## Контрольные вопросы

1. Классы данных: константы и переменные.
2. Типы данных: числовые, символьные и логические данные.
3. Объединения данных: массивы, записи, файлы.
4. Арифметические и логические выражения.
5. Арифметические операции и порядок их исполнения.
6. Приведите примеры простых и составных высказываний.
7. Сущность структурирования данных. Иерархические, сетевые и табличные структуры данных.
8. Дайте определение понятия «алгоритм».
9. Свойства алгоритмов.
10. Формы записи алгоритмов.
11. Уровни языков программирования.
12. Способы трансляции программ.



## Глава 3

# ОСНОВЫ УСТРОЙСТВА ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА

---

---

### 3.1. Принципы устройства компьютера

Среди пользователей сложилось два подхода к проблеме изучения устройства компьютера. Существует большое количество людей, которым принципиально все равно, как работает компьютер. В основу своего взаимодействия с ним они кладут советы грамотных коллег и хорошую память. В этом случае запоминают только алгоритм действий, позволяющих решить ту или иную задачу с помощью компьютера. Такой подход оправдывает себя, когда для решения профессиональных задач пользователь использует компьютер от случая к случаю, выполняя ограниченный набор действий. Невершенство этого подхода проявляется в нештатных ситуациях, когда необходимо предпринимать нестандартные действия.

Особенности второго подхода сформулировал автор ряда программ и работ в области прикладной информатики Питер Нортона: «Знать, как он работает, не менее важно, чем уметь работать с ПК. Вы можете вполне успешно пользоваться услугами компьютера, не понимая того, что в нем происходит. Однако чем глубже вы представляете процессы, происходящие в ПК, тем лучше будете использовать его возможности... Если что-нибудь случится в процессе работы с компьютером, вероятность того, что вы примите правильное решение, а не наделаете глупостей и не испортите все окончательно, будет выше» [7]. Представляется, что такая точка зрения в большей степени соответствует требованию квалификационной характеристики специалиста с высшим образованием.

### 3.1.1. Классификация ЭВМ

Электронно-вычислительные машины могут быть классифицированы по многим признакам. Для нашего курса наибольший интерес представляет классификация по времени создания и по назначению [8].

#### 3.1.1.1. Классификация ЭВМ по времени создания (поколения ЭВМ)

С момента появления электронно-вычислительных машин прошло около шестидесяти лет. Однако за этот сравнительно небольшой отрезок времени сменилось уже четыре поколения ЭВМ. В основе существующего деления на поколения лежит элементная база, на которой строятся ЭВМ (табл. 3.1). Более подробно о смене поколений ЭВМ можно прочитать в любом учебнике информатики, например, в книге [13] под поколением будем понимать серию ЭВМ, качественно отличающихся от предыдущих. Смена поколений ЭВМ связана с переходом на новую элементную базу, что приводит к существенному улучшению основных характеристик ЭВМ. В свою очередь, вследствие этого появляется возможность использования более совершенного программного обеспечения, и возникают новые области применения ЭВМ и новые информационные технологии.

Таблица 3.1. Поколения ЭВМ

Поколение	Элементная база	Способ общения с пользователем	Время появления, год
1	Электронные лампы	Двоичные коды (ДК)	1945
2	Транзисторы	ДК + языки программирования	1955
3	Микросхемы	ДК + язык управления заданиями	1965
4	Большие интегральные схемы (БИС)	ДК + пользовательский интерфейс	1975

Вместе с развитием вычислительной техники совершенствуются способы общения человека с машиной и создаются новые устройства ввода и вывода данных, обеспечивающие более комфортные условия этого общения. Происходит постепенная замена языка машины (двоичные коды) на все более близкие человеку формы обмена информацией. ЭВМ становится более дружелюбной, а требования к уровню подготовки пользователя снижаются.

Чтобы показать огромную скорость развития в производстве вычислительной техники, обычно приводится масса разнообразных сравнений и фактов. Например, в 1993 г. мощность первых процессоров Pentium превышала вычислительные возможности NASA на момент высадки космонавтов на Луну [7].

### 3.1.1.2. Классификация ЭВМ по назначению

Назначение ЭВМ стало одним из первых классификационных признаков [8]. В соответствии с ним различают большие ЭВМ, мини-ЭВМ, микроЭВМ и персональные компьютеры.

**Большие ЭВМ** — это мощные компьютеры, которые применяются для обслуживания очень крупных организаций и даже целых отраслей народного хозяйства. За рубежом компьютеры этого класса называют *мэйнфреймами (mainframe)*. В России за ними закрепился термин *большие ЭВМ*. Штат обслуживания большой ЭВМ составляет до нескольких десятков человек. На базе таких суперкомпьютеров создают *вычислительные центры*, включающие в себя несколько отделов или групп.

**Мини-ЭВМ** отличаются от больших ЭВМ уменьшенными размерами и, соответственно, меньшей производительностью и стоимостью. Такие компьютеры используются крупными предприятиями, научными учреждениями и некоторыми высшими учебными заведениями, сочетающими учебную деятельность с научной. Мини-ЭВМ может сочетать управление производством с другими задачами. Например, он может помогать экономистам в осуществлении контроля себестоимости продукции, нормировщикам — в оптимизации времени технологических операций, конструкторам — в автоматизации проектирования станочных приспособлений, бухгалтерии — в осуществлении учета первичных документов и подготовки регулярных отчетов для налоговых органов. Для организации работы с мини-ЭВМ тоже требуется специальный вычислительный центр, хотя и не такой многочисленный, как для больших ЭВМ.

**МикроЭВМ** доступны многим предприятиям. Организации, использующие микроЭВМ, обычно не создают вычислительные центры. Для обслуживания такого компьютера достаточно небольшой вычислительной лаборатории в составе нескольких человек. Несмотря на относительно невысокую производительность по сравнению с большими ЭВМ, микроЭВМ находят применение и в крупных вычислительных центрах. Там им поручают

вспомогательные операции, для которых нет смысла использовать дорогие суперкомпьютеры. К таким задачам, например, относится предварительная подготовка данных.

**Персональные компьютеры (ПК)** получили широкое распространение в течение последних двадцати лет. Из названия видно, что такой компьютер предназначен для обслуживания одного пользователя. Несмотря на свои небольшие размеры и относительно невысокую стоимость, современные персональные компьютеры обладают немалой производительностью. Многие современные персональные модели превосходят большие ЭВМ 70-х годов, мини-ЭВМ 80-х годов и микро-ЭВМ первой половины 90-х годов. Персональный компьютер (*Personal Computer, PC*) вполне способен удовлетворить большинство потребностей малых предприятий и отдельных лиц.

Особенно широкую популярность персональные компьютеры получили после 1995 г. в связи с бурным развитием Интернет. Персонального компьютера вполне достаточно для использования всемирной сети в качестве источника научной, справочной, учебной, культурной и развлекательной информации. Персональные компьютеры являются также удобным средством автоматизации учебного процесса по любым дисциплинам, средством организации дистанционного (заочного) обучения и средством организации досуга. Они вносят большой вклад не только в производственные, но и в социальные отношения. Их нередко используют для организации домашней трудовой деятельности, что особенно важно в условиях безработицы.

В свою очередь, по размерам и функциональным возможностям большие ЭВМ, мини-ЭВМ, микроЭВМ и персональные компьютеры подразделяют на массовые, деловые, портативные, развлекательные и рабочие станции [8]. Начиная с 1999 г., в области персональных компьютеров действует международный сертификационный стандарт — *спецификация PC99*, — устанавливающий следующие категории персональных компьютеров:

- Consumer PC (массовый ПК);
- Office PC (деловой ПК);
- Mobile PC (портативный ПК);
- Workstation PC (рабочая станция);
- Entertainment PC (развлекательный ПК).

В соответствии с данной спецификацией значительная часть ПК, выпускающихся серийно, относится к числу массовых и позволяет решать широкий круг задач. Деловые компьютеры пред-

назначены для использования в учреждении. Поэтому для них минимизированы требования к средствам воспроизведения графики и не регламентированы требования по работе со звуковыми данными. Цель создания портативного ПК заключалась в обеспечении пользователя информационными технологиями не только на стационарно оборудованном рабочем месте, но и вне его. Поэтому ПК, относящиеся к данной категории, обязательно должны иметь в своем составе средства компьютерной связи. Рабочие станции создаются для работы в сети, построенной в соответствии с архитектурой «клиент — сервер». Поэтому они должны отвечать повышенным требованиям к средствам хранения данных. Развлекательные ПК оборудуются мощными средствами воспроизведения графики и звука.

### 3.1.1.3. Другие принципы классификации ЭВМ

На практике используются и другие принципы классификации ПК. Например, по уровню специализации. При использовании этого показателя компьютеры делят на универсальные и специализированные. Универсальные ПК предназначаются для многоцелевого использования, а специализированные — для решения конкретных задач.

Иногда можно столкнуться с классификацией персональных компьютеров по типоразмерам. Так, различают *настольные* (*desktop*), *портативные* (*notebook*) и *карманные* (*palmtop*) модели.

Особого упоминания заслуживает принцип классификации по совместимости. Это объясняется тем, что компьютеры выпускаются разными производителями, собираются из разных деталей, работают с разными программами. При этом очень важным вопросом становится *совместимость* различных компьютеров между собой. От совместимости зависит взаимозаменяемость узлов и приборов, предназначенных для разных компьютеров, возможность переноса программ с одного компьютера на другой и возможность совместной работы разных типов компьютеров с одними и теми же данными. Различают несколько видов совместимости, из которых для обычного пользователя наиболее существенное значение имеет аппаратная совместимость. По этому признаку различают так называемые аппаратные платформы. Наиболее широкое распространение в наши дни получили две аппаратные платформы — *IBM PC* и *Apple Macintosh*.

### 3.1.2. Принципы «фон Неймана»

Несмотря на стремительные темпы развития вычислительной техники, фундаментальные принципы построения ЭВМ практически не изменились. Базовые идеи построения вычислительных устройств были сформулированы известным математиком Джоном фон Нейманом с группой соавторов в 1946 г. и получили название «принципов фон Неймана». Сущность этих принципов:

- **использование двоичной системы** для представления чисел. В работе фон Неймана были убедительно продемонстрированы преимущества двоичной системы для технической реализации, удобство и простота выполнения в ней арифметических и логических операций. В дальнейшем ЭВМ стали обрабатывать и нечисловые виды информации: текстовую, графическую, звуковую и др. Но по-прежнему двоичное кодирование данных составляет информационную основу любого современного компьютера;
- **принцип «хранимой программы»**. Согласно этому принципу программа, записанная с помощью двоичных кодов, должна храниться в той же самой памяти, что и обрабатываемые ею данные;
- **принцип адресности**. Команды и данные помещаются в ячейки памяти, доступ к которым осуществляется по адресу. Адресом ячейки фактически является ее номер; таким образом, местонахождение информации в ОЗУ также кодируется в виде двоичных чисел.

В ЭВМ, построенной по принципам фон Неймана, происходит последовательное считывание команд из памяти и их выполнение. Номер (адрес) очередной ячейки памяти, из которой будет извлечена следующая команда программы, формируется и хранится в специальном устройстве — счетчике команд.

Фон Нейман с соавторами не только выдвинули основополагающие принципы логического устройства ЭВМ, но и предложили ее структуру, которая полностью воспроизводилась в течение первых двух поколений ЭВМ. Схема устройства такой ЭВМ представлена на рис. 3.1.

В соответствии с принципами фон Неймана компьютер должен иметь в своем составе следующие устройства:

- **арифметико-логическое устройство (АЛУ)** — предназначено для обработки закодированной информации и может выполнять арифметические и логические операции;

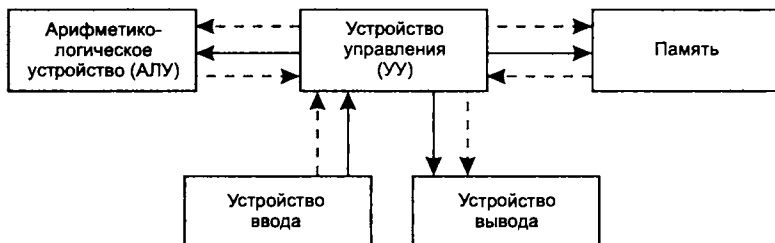


Рис. 3.1. Схема ЭВМ

- *устройство управления (УУ)* — назначение данного устройства состоит в организации процесса выполнения программ;
- *память* или *запоминающее устройство (ЗУ)* — служит для хранения программ и данных. Память компьютера должна состоять из некоторого количества пронумерованных ячеек. В каждой из них могут находиться или обрабатываемые данные, или инструкции программ. Все ячейки памяти должны быть одинаково легко доступны для других устройств компьютера;
- *внешние устройства* — *устройство ввода* и *устройство вывода* — выполняют задачи по вводу и выводу данных, т. е. обеспечивают прямую и обратную связь пользователя с компьютером.

Обработка информации в таком компьютере предписывается алгоритмом и сводится к последовательному выполнению команд в порядке, однозначно определяемом программой. На рис. 3.1 сплошными линиями показано прохождение команд, а пунктирными — данных.

Следует отметить, что благодаря разработке и внедрению больших интегральных схем стало возможным объединение устройства управления и АЛУ в одно, называемое центральным процессором, который определяет действия, подлежащие выполнению, путем считывания команд из оперативной памяти. Но несмотря на это, в основе работы современных ЭВМ лежат принципы, сформулированные более полувека тому назад. Однако прогресс в области разработки и создании элементной базы привел к совершенствованию архитектуры компьютеров. В настоящее время термин «**архитектура ЭВМ**» используется для описания принципа действия, конфигурации и взаимного соединения основных логических узлов ЭВМ.

### 3.1.3. Принцип открытой архитектуры

Значительные успехи в миниатюризации электронных схем создали предпосылки для заметного роста быстродействия процессора. Возникло существенное противоречие между высокой скоростью обработки информации внутри машины и медленной работой устройств ввода-вывода, в большинстве своем содержащих механические движущиеся части. Если бы процессор обеспечивал взаимодействие устройств компьютера, как описано выше, то значительную часть времени он был бы вынужден простаивать в ожидании информации от внешних устройств, что существенно снижало бы эффективность работы всей ЭВМ в целом. Решением этой проблемы было освобождение центрального процессора от функций обмена информацией и к передаче их специальным электронным схемам управления работой внешних устройств. Такие схемы получили название *контроллер внешнего устройства*. Наличие интеллектуальных контроллеров внешних устройств стало важной отличительной чертой машин третьего и в дальнейшем четвертого поколений.

Контроллер можно рассматривать как специализированный процессор, управляющий работой соответствующего внешнего устройства по специальным встроенным программам. Применение контроллеров позволило использовать для связи между отдельными функциональными узлами ЭВМ принципиально новое устройство — общую шину (часто ее называют магистралью). Шина состоит из трех частей: шина данных, по которой передается информация; шина адреса, определяющая, куда именно передаются данные; шина управления, регулирующая процесс обмена информацией.

Наличие магистрали позволяет изменить организацию обмена информацией между ОЗУ и внешним устройством. С этой целью центральный процессор выдает контроллеру задание на осуществление этого обмена, а последний создает канал для этого. Дальнейшая передача информации протекает под управлением контроллера без использования аппаратно-программных средств центрального процессора. Это создает возможность центральному процессору продолжать выполнение программы. Схема устройства компьютера в этом случае приобретает вид, представленный на рис. 3.2.

Компьютер, созданный по такой схеме, легко пополнять новыми устройствами. Данное свойство называют *открытостью*



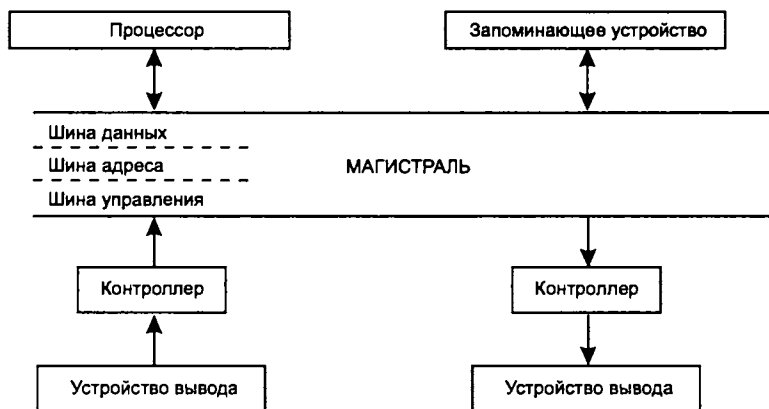


Рис. 3.2. Схема ЭВМ с использованием магистральной

*архитектуры.* Для пользователя открытая архитектура означает возможность свободно выбирать состав внешних устройств для своего компьютера, т. е. конфигурировать его в зависимости от круга решаемых задач. Принцип построения ЭВМ, в соответствии с которым обмен информацией между устройствами организуется с помощью магистральной, получил название магистрально-модульной или принципа открытой архитектуры. Этот термин означает, что модульный принцип позволяет потребителю самому комплектовать нужную ему конфигурацию компьютера и производить при необходимости ее модернизацию. При этом модульная организация компьютера опирается на магистральный (шинный) принцип обмена информацией между модулями.

## 3.2. Аппаратная реализация персонального компьютера

### 3.2.1. Основные устройства персонального компьютера

Основными особенностями современного компьютера являются объединение АЛУ и УУ в одном устройстве — процессоре, и разделение памяти на внутреннюю и внешнюю. Рассмотрим эти особенности.

### 3.2.1.1. Процессор

**Процессор** — устройство, предназначенное для обработки информации и управления этим процессом. Представление о составе и организации процессора дает рис. 3.3. Его основу составляют регистры — быстродействующие устройства, предназначенные для хранения и обработки информации (команд и данных).

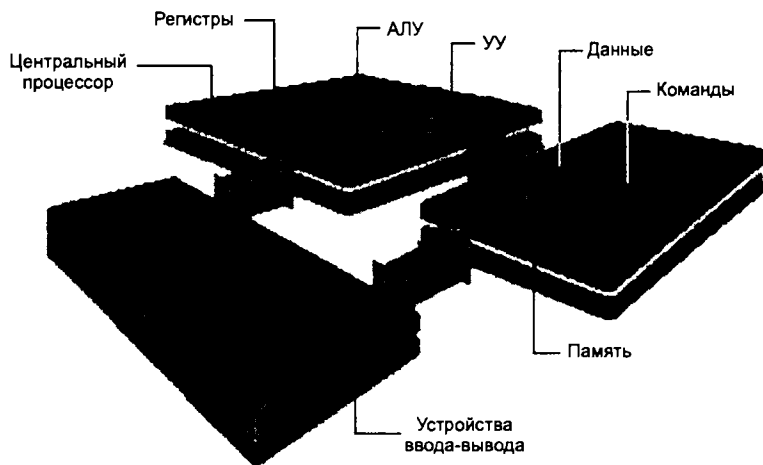


Рис. 3.3. Схема устройства процессора

Процессор состоит из арифметико-логического устройства (АЛУ), устройства управления (УУ) и нескольких ячеек внутренней памяти — регистров. В регистрах хранятся команды, данные и адреса. АЛУ выполняет числовые и логические операции с данными в соответствии с кодом команды, хранящимся в регистре команд (сложение, сравнение, простые логические операции — И, ИЛИ, НЕ, операции сдвига, арифметического, логического или циклического сложения и т. п.). УУ с помощью набора управляющих сигналов организует согласованную работу всех блоков процессора и управляет как передачей адресов, команд и данных в процессоре по внутренней шине, так и взаимодействием процессора с внешними устройствами.

Главными характеристиками процессора являются его быстродействие — число выполняемых операций в единицу времени, и разрядность — объем информации, которую процессор обра-

батывает за одну операцию. Быстродействие современных процессоров превышает 1 ГГц ( $10^9$  Герц). Под разрядностью процессора принято понимать количество двоичных разрядов в его регистрах. Разрядность наиболее распространенных современных моделей составляет 32.

### 3.2.1.2. Память

Необходимость разделения памяти на внутреннюю и внешнюю вызвано тем, что ЭВМ не может непрерывно работать в течение неопределенно длительного периода времени. Действительно, в случае необходимости устранения неисправности или профилактического обслуживания питание компьютера необходимо отключить. Следовательно, возникает проблема сохранения программ и данных, находящихся в памяти компьютера.

В современных компьютерах различают два вида памяти:

*Внутренняя память* — электронная (полупроводниковая) память, которая размещается на системной плате или на платах расширения.

*Внешняя память* — память, реализованная в виде устройств с различными принципами хранения информации и обычно с подвижными носителями. В настоящее время сюда входят устройства магнитной (дисковой и ленточной) памяти, оптической и магнитооптической памяти. Устройства внешней памяти могут размещаться как в системном блоке компьютера, так и в отдельных корпусах.

Основным назначением внутренней памяти является совместное хранение данных и программ их обработки в процессе преобразования этих данных. Внешняя же память предназначена для длительного хранения информации, когда те или иные данные или программы не используются или же компьютер выключен.

**Внутренняя память.** Внутренняя память имеет в своем составе несколько устройств. Основным является оперативное запоминающее устройство (ОЗУ). Соответствующий по смыслу английский термин, который часто встречается в технической литературе, — RAM (Random Access Memory), т. е. память с произвольным (случайным) доступом. Такой доступ подразумевает возможность получать данные из памяти по любым адресам в любом порядке.

ОЗУ имеет непосредственную связь с процессором. В нем хранятся программные команды и данные, участвующие в дан-

ное время в вычислениях. В него записываются результаты вычислений перед пересылкой их во внешнюю память или на устройства вывода. Основными особенностями ОЗУ являются:

- возможность считывать и записывать информацию из произвольного места памяти;
- высокая скорость работы, близкая к быстрдействию микропроцессора;
- необходимость специальных мер по сохранению информации из ОЗУ после завершения работы (энергозависимость).

Другим важным устройством внутренней памяти компьютера является *постоянное запоминающее устройство* (ПЗУ). Техническое (английское) название этого устройства памяти — ROM (Read Only Memory), т. е. память только для чтения. Информация в это устройство записывается производителем, сохраняется неизменной и постоянно доступна компьютеру, в том числе сразу в момент включения. ПЗУ играет очень важную роль, потому что в нем записана программа начальной загрузки компьютера. Кроме того, в этой же самой микросхеме обычно хранятся минимальные программы работы с клавиатурой и другими устройствами, поэтому ее часто называют **BIOS** — Basic Input/Output System.

В современных компьютерах быстрдействие процессора и ОЗУ может существенно отличаться. Поэтому, для повышения производительности системы в качестве буфера между АЛУ и ОЗУ используется сверхоперативное ЗУ (СЗУ) или кэш-память. Название «кэш» происходит от английского слова «cache», которое обозначает тайник. СЗУ невидимо для пользователя и данные, хранящиеся в нем, недоступны для прикладного программного обеспечения.

Основная идея работы кэш-памяти заключается в том, что извлеченные из ОЗУ данные или команды программы копируются в СЗУ; одновременно в специальный каталог адресов, который находится в той же самой памяти, запоминается, откуда информация была извлечена. Если данные потребуются повторно, то уже не надо будет терять время на обращение к ОЗУ — их можно получить из кэш-памяти значительно быстрее.

В настоящее время кэш-память обычно реализуется по двухуровневой схеме. При этом первичный кэш (level 1 — уровень 1) встроен непосредственно внутрь процессора, а вторичный (level 2) устанавливается на системной плате. Как и для

ОЗУ, увеличение объема кэша повышает эффективность работы компьютера.

**Внешняя память.** Основное назначение внешней памяти компьютера заключается в длительном хранении информации (как программ, так и данных). Наличие внешней памяти обеспечивает возможность неоднократного использования информации в течение длительного времени. Информация во внешней памяти хранится в двоичном представлении, что позволяет обрабатывать ее без каких бы то ни было дополнительных преобразований. Поскольку скорость записи и считывания информации в устройствах внешней памяти намного ниже быстродействия центральных устройств, постольку для их подключения к магистрали необходим контроллер. Как правило, в составе компьютера имеются несколько устройств внешней памяти. Для персонального компьютера это главным образом гибкие и жесткие магнитные диски, а также оптические диски.

**Накопители на гибких магнитных дисках (НГМД).** В повседневной жизни гибкие магнитные диски называют дискетами. Их устройство приведено на рис. 3.4.

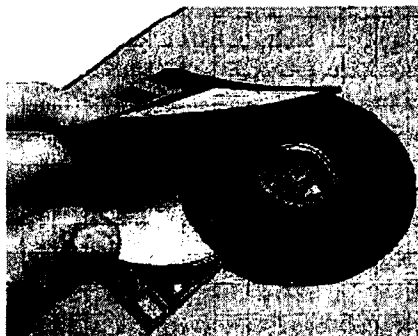


Рис. 3.4. Устройство дискеты

Специальное устройство, называемое **дискководом**, позволяет записывать на дискету и считывать с нее информацию. Дискета вставляется в специальное устройство, называемое накопителем на гибких дисках (НГМД, FDD — floppy disk drive). Диск вращается в пластиковом футляре, содержащем специальную прокладку для уменьшения трения. На тонкую пластиковую основу диска нанесен ферромагнитный порошок. Две магнитные головки (одна сверху, другая снизу), могут быть подведены к одной из

80) концентрических окружностей, на которые условно разбита поверхность диска. С их помощью можно либо считать информацию, либо записать ее. Причем в качестве носителей информации выступают микроскопические частички порошка, которые могут быть либо намагничены (соответствует сигналу 1), либо не намагничены (соответствует сигналу 0).

На стандартную дискету обычно можно поместить до 1,44 Мбайт информации, хотя и появились устройства, позволяющие записать на нее несколько десятков мегабайт информации.

**Накопители на жестких магнитных дисках (НЖМД).** Принцип действия накопителя на жестких магнитных дисках не отличается от принципа действия накопителя на гибких магнитных дисках. Жесткий магнитный диск со снятым кожухом и накопитель на жестких магнитных дисках (НЖМД, HDD — hard disk drive) изображены на рис. 3.5. На жестких магнитных дисках ферромагнитный порошок нанесен на алюминиевую или на стеклянную основу 1. На одном вращающемся шпинделе 2 крепится целый пакет дисков (до трех), к каждой стороне которых подходит магнитная головка 3. Таким образом, при фиксированном положении позиционера 4 все головки описывают несколько концентрических окружностей, составляющих один цилиндр. Позиционер представляет собой шаговый электродвигатель, имеющий несколько сот (или даже тысяч) фиксированных положений. Ко-



Рис. 3.5. Устройство НМЖД

личество этих положений и определяет количество цилиндров на НЖМД.

**Накопители на оптических дисках (CD-ROM).** Дисководы оптических дисков считывают информацию в 10—15 раз быстрее, чем гибкие, но все же медленнее, чем жесткие. Устройство накопителя на компакт-дисках (CD-ROM — Compact Disk — Read Only Memory) изображено на рис. 3.6. В направляющем лотке компакт-диск 2 подается внутрь устройства и крепится на вращающемся шпинделе 1 (закрыт прижимной крышкой). При вращении диск освещается лазерной головкой 3, луч которой либо отражается от поверхности (соответствует 1), либо рассеивается (соответствует 0). Лазерная головка перемещается вдоль поверхности диска с помощью позиционера. Отраженный луч вырабатывает сигнал, передающийся в конечном итоге через оперативную память в центральный процессор для обработки.

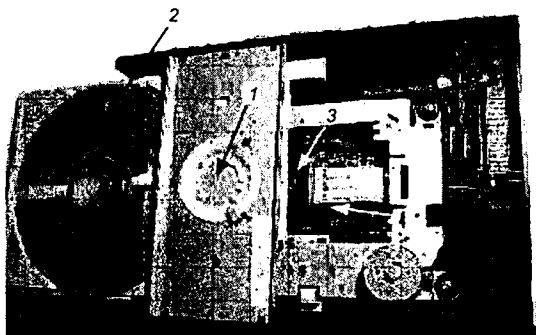


Рис. 3.6. Устройство CD-ROM

Поверхность компакт-диска представляет собой одну спиральную дорожку, на которой располагаются микроскопические впадины, рассеивающие попадающий на них лазерный луч. Набор нулей и единиц на диске располагается по правилам, которые называются форматом. Музыкальные компакт-диски — это один из форматов. Устройства характеризуются скоростью, с которой считывается информация с диска. Одинарная скорость соответствует скорости вращения музыкального диска и составляет 150 Кбит в секунду.

**Иерархия памяти.** Все виды компьютерной памяти связаны между собой, образуя иерархическую структуру, представленную на схеме (рис. 3.7).

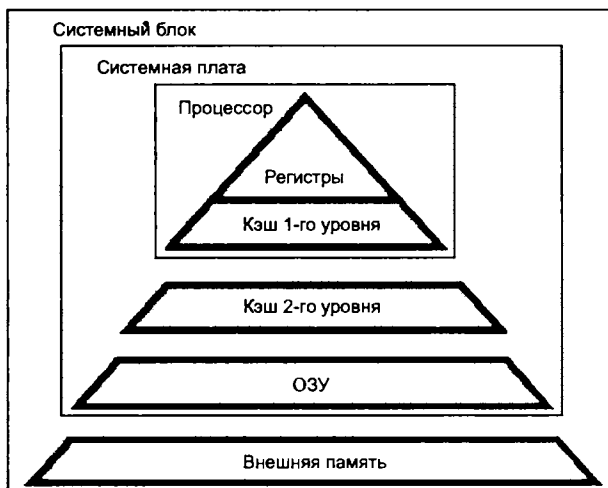


Рис. 3.7. Схема организации памяти компьютера

Схема показывает, что чем ближе то или иное устройство памяти к процессору, тем меньше ее объем, но зато больше скорость ее работы. Иерархия памяти специально спроектирована так, что информация, необходимая для решения задачи, распределяется по уровням памяти в соответствии с потребностью в ней, например, файлы, в которых хранятся требуемые для решения задачи данные, считываются в ОЗУ, а наиболее часто используемые данные из них попадают в кэш-память.

Особого рассмотрения заслуживает взаимодействие ОЗУ и внешней памяти. Дело в том, что объем ОЗУ ограничен, а размер исполняемых файлов современных прикладных программ достигает нескольких десятков мегабайт. Следует учесть, что одновременно могут выполняться несколько приложений, которые должны быть загружены в ОЗУ. К этому добавляется объем обрабатываемых данных, который также постоянно растет. Кроме того, для обеспечения работы компьютера в оперативной памяти должны находиться и файлы операционной системы.

Решением данного противоречия является использование *виртуальной памяти*. Современные операционные системы работают в предположении, что компьютер обладает настолько большим объемом внутренней памяти, что реально установленное ОЗУ составляет лишь часть ее. Вся оставшаяся часть внутренней памяти, необходимой для решения задач, располагается в специ-



альном системном файле на жестком диске. Эта область жесткого диска называется файлом подкачки. Теперь, если объем ОЗУ по какой-то причине оказывается недостаточным, система копирует менее востребованную в данный момент область оперативной памяти в этот файл, освобождая тем самым необходимый объем ОЗУ. Когда, наоборот, потребуются данные с диска, то они будут возвращены в оперативную память на место, подготовленное там тем же самым способом.

Очевидно, что скорость обмена данными между ОЗУ и файлом подкачки невелика. Поэтому, если объем ОЗУ в компьютере недостаточен, то программы будут выполняться медленно из-за постоянного обмена данными между ОЗУ и жестким диском. В подобной ситуации для повышения скорости работы компьютера необходимо увеличить объем установленного на плате ОЗУ.

### 3.2.1.3. Системная (материнская) плата

Основным конструктивным элементом персонального компьютера является системная (материнская) плата. Обычно на ней монтируются основной процессор, устройства внутренней памяти (ОЗУ, ПЗУ и СЗУ) и несколько групп устройств:

- разъемы для подключения отдельных устройств;
- шина — информационная магистраль, связывающая их воедино;
- базовый набор микросхем, чипсет, с помощью которого материнская плата осуществляет контроль над всем происходящим внутри системного блока;
- встроенные (или интегрированные) дополнительные устройства.

Самые важные характеристики материнской платы — скорость передачи данных, число поддерживаемых моделей процессоров, базовый тип оперативной памяти, параметры работы с ней и некоторые другие напрямую зависят от типа чипсета. Главные составляющие любого чипсета называются «мостами». Каждый из двух имеющихся в любом чипсете «мостов» — это специальная микросхема, выполняющая определенные функции. Так, «северный» мост соединяет между собой процессор, оперативную память и видеошину AGP, т. е. фактически он обеспечивает работу центральных устройств компьютера. В свою очередь второй, «южный» мост обеспечивает работу всех периферийных устройств, подключенных к компьютеру.

Всегда находится на материнской плате контроллер клавиатуры и накопителей на магнитных дисках. Такие контроллеры называются *встроенными* или *интегрированными* (в материнскую плату). Поскольку разным пользователям в компьютере нужен разный набор контроллеров, постольку в большинстве компьютеров материнская плата всегда содержит несколько разъемов (*слотов*), в которые могут вставляться электронные платы, содержащие контроллеры для подключения дополнительных устройств (*платы контроллеров*). При вставке в разъем материнской платы контроллер подключается к *шине* — магистрали передачи данных между оперативной памятью и устройствами. Именно таким способом реализуется принцип открытой архитектуры.

### 3.2.2. Периферийные устройства персонального компьютера

#### 3.2.2.1. Устройства ввода

Обязательным устройством ввода информации является клавиатура. С ее помощью можно не только вводить числовую и текстовую информацию, но и управлять работой компьютера.

Ввод информации осуществляется с помощью клавиш. Нажатие на клавишу вызывает генерирование последовательности электрических импульсов (8-битные коды клавиш). Клавиатура подключается к магистрали через контроллер, который расположен на системной плате, а его разъем выведен на заднюю панель системного блока. Технически клавиатура реализуется различными способами, внутренние клавишные переключатели могут быть механическими (с различными типами контактов) или мембранными («замыкание» — «размыкание» контакта осуществляется за счет изменения электростатической емкости конденсатора).

Совершенствование способов диалога человека и ЭВМ привело к созданию других устройств, обеспечивающих ввод информации. Так, переход от текстового способа общения (набор команды с помощью клавиатуры) к графическому (выбор пиктограммы или пункта меню) вызвал появление координатных устройств ввода. К их числу относятся манипуляторы мышь, трекбол, сенсорная панель.

В манипуляторах типа мышь и трекбол используется оптико-механический принцип действия. Основным рабочим органом манипуляторов является массивный шар (металлический, покрытый резиной). При использовании мыши этот шар вращается при перемещении корпуса манипулятора по горизонтальной поверхности. У трекбола этот шар вращается рукой пользователя, а корпус манипулятора остается неподвижным.

Вращение шара передается роликам, которые фиксируют перемещение по координатной плоскости и с помощью фоточувствительных элементов передают информацию о величине этого перемещения в компьютер. Таким образом, вращение шара-манипулятора превращается в движение курсора или какого-либо другого объекта на экране монитора. Манипуляторы обычно подключаются к одному из последовательных портов компьютера.

Еще одним координатным устройством ввода является сенсорная панель (TouchPad). Она представляет собой панель прямоугольной формы, чувствительную к нажатию пальцев. Это устройство обеспечивает выполнение тех же функций, что и манипуляторы, но является более компактным и не требует перемещения в пространстве. Эти качества обеспечивают успешное использование сенсорной панели в портативных компьютерах. Прикосновение пальца к поверхности панели и его перемещение управляет курсором так же, как и соответствующие воздействия манипулятора.

Физически сенсорная панель представляет собой сетку из металлических проводников, разделенных тонкой изолирующей прокладкой. Такая конструкция эквивалентна набору большого количества миниатюрных конденсаторов. Так как человеческое тело является хорошим проводником, то приближение руки пользователя к поверхности панели вызывает изменение емкости этих конденсаторов. По изменению емкости того или иного конденсатора в сетке можно точно определить координаты пальца на поверхности панели.

В качестве устройств ввода иногда используются игровые манипуляторы — джойстики. Они применяются в различного рода тренажерах и для управления ходом компьютерных игр. Конструктивно обычно они выполняются в виде рукоятки или рулевого колеса, подобного автомобильному, кнопками на подставке. Джойстики подключаются к специальному разъему (игровой порт) на звуковой плате.

Для обеспечения оптического ввода изображений, представленных в виде фотографий, рисунков, слайдов, а также текстовых документов, и их преобразования в цифровую форму используются сканеры. В процессе сканирования изображение, которое необходимо ввести в ЭВМ, освещается. В черно-белых сканерах для этого используется белый цвет, а в цветных — три цвета (красный, зеленый и синий). Отраженный свет проецируется на линейную матрицу фотоэлементов, которая движется. При этом происходит последовательное считывание изображения и его преобразование в двоичный код. В результате исходное изображение преобразуется в графический файл. Сканеры подключаются к магистрали с помощью специальных контроллеров или через параллельный порт компьютера. Для преобразования отсканированного текста из графического формата в текстовый служат системы распознавания текстовой информации.

Для получения видеоизображений и фотоснимков непосредственно в цифровом (компьютерном) формате служат цифровые камеры (видеокамеры и фотоаппараты). Они могут быть постоянно подключены к компьютеру и тем самым обеспечивают запись изображений на жесткий диск или его передачу по компьютерным сетям. Цифровые камеры способны хранить в своей памяти десятки изображений, которые после подключения к компьютеру могут быть переписаны на жесткий диск.

Звуковая информация вводится в компьютер с помощью микрофона, который подключается к специальному контроллеру — звуковой карте. Это устройство обеспечивает 16-битное двоичное кодирование звука. Обычно звуковая карта имеет также дополнительную возможность синтезировать звук (в памяти звуковой карты хранятся звуки 128 различных музыкальных инструментов) и воспроизводить одновременно 32 и более инструмента. Звуковая карта устанавливается в один из слотов расширения на системной плате.

### 3.2.2.2. Устройства вывода

Основным устройством вывода информации, обеспечивающим постоянное взаимодействие пользователя с компьютером, служит монитор. Принцип вывода информации на экран монитора состоит в следующем: в видеопамяти компьютера содержится битовая карта изображения и периодически происходит считывание содержимого видеопамяти и отображение его на эк-

ран. Частота считывания (регенерации изображения) определяет стабильность изображения на экране, которое в современных мониторах обычно составляет более 75 Гц.

Монитор подключается к специальному контроллеру — видеокарте (видеоадаптеру). Этот контроллер может быть как встроенным, так и установленным в слот расширения системной платы. Именно на видеокарте располагается видеопамять, с помощью которой осуществляется управление возможными режимами вывода изображения на экран монитора (количество точек по горизонтали и вертикали, количество цветов).

В настольных компьютерах могут использоваться мониторы на электронно-лучевой трубке или дисплейные панели на жидких кристаллах.

Для вывода числовой, текстовой или графической информации на бумагу — создания «твердой копии» — служат принтеры. По принципу действия принтеры делятся на матричные, струйные и лазерные. Печатающая головка матричного принтера состоит из вертикального столбца маленьких стержней (обычно 9 или 24), которые под воздействием магнитного поля «выталкиваются» из головки, ударяют по бумаге (через красящую ленту) и оставляют строку символов. Скорость печати этих принтеров невелика. Поэтому сейчас они имеют ограниченное применение.

Широкое распространение получили черно-белые и цветные струйные принтеры. В них используются термическая или пьезоэлектрическая чернильная печатающая головка, которая под давлением выбрасывает чернила из ряда сопел на бумагу. Струйные принтеры обеспечивают достаточно высокое качество и скорость печати.

Лазерные принтеры обеспечивают наиболее высокое качество печати. В основе действия лазерного принтера лежит электрографический принцип печати, принятый в ксерографии. Фоточувствительный барабан облучается модулированным лазерным лучом (на барабане возникают положительно заряженные точки), затем на барабан наносится электрографический проявитель (тонер), который осаждается на положительно заряженных участках. Предварительно отрицательно заряженная бумага прижимается к барабану, и тонер прилипает к ней. Последний этап состоит в термической фиксации порошка на бумаге. Этот процесс проходит при температуре около 200 °С. Принтер подключается к магистрали компьютера через параллельный порт.

Для вывода сложных и широкоформатных графических объектов (плакатов, чертежей, электрических и электронных схем и т. д.) используются специальные устройства вывода — графопостроители или плоттеры. В основу работы плоттера положен тот же принцип действия, как и у струйных принтеров.

Для вывода звука в ЭВМ используются акустические колонки или наушники, которые подключаются к выходу звуковой платы.

### 3.2.2.3. Устройства хранения данных

Необходимость во внешних устройствах хранения данных возникает в двух случаях:

- когда на вычислительной системе обрабатывается больше данных, чем можно разместить на ее основном жестком диске;
- если данные имеют повышенную ценность и необходимо выполнять регулярное резервное копирование на внешнее устройство (копирование данных на жестком диске не является резервным и только создает иллюзию безопасности).

В настоящее время для внешнего хранения данных используют несколько типов устройств, использующих магнитные или магнитооптические носители.

**Стримеры** — это накопители на магнитной ленте. Их отличает сравнительно низкая цена. К недостаткам стримеров относят малую производительность, что связано с тем, что магнитная лента — это устройство последовательного доступа, и недостаточную надежность (кроме электромагнитных наводок, ленты стримеров испытывают повышенные механические нагрузки и могут физически выходить из строя).

Емкость магнитных кассет (картриджей) для стримеров составляет до нескольких сот мегабайт. Дальнейшее повышение емкости за счет повышения плотности записи снижает надежность хранения, а повышение емкости за счет увеличения длины ленты сдерживается низким временем доступа к данным.

**ZIP-накопители** выпускаются компанией Imega, специализирующейся на создании внешних устройств для хранения данных. Устройство работает с дисковыми носителями, по размеру незначительно превышающими стандартные гибкие диски и имеющими емкость 100/250 Мбайт. Они подключаются к стан-

дартному параллельному порту. Поэтому скорость обмена данными существенно снижается.

**Накопители JAZ** также выпускаются компанией Iomega. По своим характеристикам JAZ-носитель приближается к жестким дискам, но в отличие от них является сменным. В зависимости от модели накопителя на одном диске можно разместить 1 или 2 Гбайт данных.

### 3.2.3. Базовая конфигурация компьютера

На основе изложенного ранее можно сделать вывод, что современный персональный компьютер представляет собой универсальную техническую систему. Благодаря реализации магистрально-модульного принципа построения ПК его конфигурацию можно изменять в зависимости от тех задач, для решения которых предназначен конкретный ПК, т. е. в зависимости от функциональных обязанностей пользователя, для оснащения рабочего места которого и будет использована данная машина. В соответствии с [8] термин «конфигурация» означает состав оборудования ПК.

На практике существует понятие базовой конфигурации, включающей минимально комплект устройств, необходимых для нормальной работы. В настоящее время принято считать [3], что в базовой конфигурации компьютер включает системный блок, монитор, клавиатуру и мышь (рис. 3.8). Системный блок представляет собой основной узел, внутри которого установлены наиболее важные компоненты. Устройства, подключаемые к системному блоку, называют *внешними*. Внешние дополнительные устройства, предназначенные для ввода, вывода и длительного хранения данных, также называют *периферийными*.

Корпуса персональных компьютеров выпускают в горизонтальном (*desktop*) и вертикальном (*tower*) исполнении. Корпуса, имеющие вертикальное исполнение, различают по габаритам: *полноразмерный (big tower)*, *среднеразмерный (midi tower)* и *малоразмерный (mini tower)*. Корпуса персональных компьютеров поставляются вместе с блоком питания и, таким образом, мощность блока питания также является одним из параметров корпуса. Для массовых моделей достаточной является мощность блока питания 200—250 Вт.



Рис. 3.8. Компоновка компьютера

Внутри системного блока размещаются системная плата, жесткий диск и дисковод для гибких магнитных дисков. В последнее время в базовую комплектацию включают и устройство для чтения компакт-дисков.

В свою очередь, системная плата является основой, на которой монтируются все внутренние устройства ПК.

### Контрольные вопросы

1. Поколения ЭВМ.
2. Принципы устройства ЭВМ.
3. Назначение и устройство процессора.
4. Назначение и состав внутренней памяти компьютера.
5. Назначение внешней памяти компьютера и устройства, входящие в ее состав.
6. Назначение и состав периферийных устройств компьютера.



## Глава 4

# ПРОГРАММНЫЕ СРЕДСТВА ИНФОРМАТИЗАЦИИ

Очевидно, что по своей сути работа ЭВМ представляет собой процесс последовательного выполнения набора команд — программ. Это означает, что возможности компьютера как технической основы системы обработки данных связаны с используемым программным обеспечением (программами). На этом основано утверждение о том, что современный компьютер представляет собой единство аппаратной и программной частей. Об этом свидетельствует и соотношение стоимости этих составляющих вычислительного комплекса. Известны данные [8] о динамике соотношения стоимости программных и аппаратных средств (табл. 4.1, рис. 4.1).

Таблица 4.1. Соотношение стоимости программной и аппаратной частей ЭВМ

Части ЭВМ	Годы XX в.			
	50-е	60-е	70-е	80-е
Аппаратные средства, %	70	35	20	15
Программные средства, %	30	65	80	85

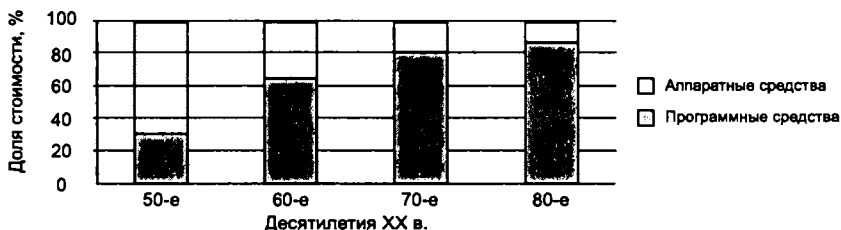


Рис. 4.1. Соотношение стоимости программной и аппаратной частей ЭВМ

При этом следует помнить, что именно программное обеспечение позволяет использовать одни и те же аппаратные средства для решения задач в разных отраслях человеческой деятельности.

Все это свидетельствует о том, что для эффективного использования компьютера пользователь должен глубоко понимать принципы функционирования программного обеспечения.

## **4.1. Классификация программного обеспечения**

В процессе решения задачи компьютер обрабатывает данные, выполняя определенную программу. Программа позволяет использовать один и тот же компьютер для решения разных задач: проектирования технических устройств, ведения регистров бухгалтерского учета, выполнения сложных вычислений и, наконец, игр. Программа представляет собой упорядоченную последовательность команд, которые исполняются компьютером с целью решения той или иной задачи. Однако программы могут использоваться не только их разработчиками. Поэтому для нормального их использования необходимы документы, содержащие описание и инструкции пользователям. Совокупность программ обработки данных и документов, необходимых для их эксплуатации, представляет собой программное обеспечение (ПО). Легальное распространение ПО может осуществляться путем продаж на рынке или с помощью телекоммуникаций. Программы, предназначенные для распространения с помощью рынка, должны отвечать ряду требований, и получили название программный продукт. Программный продукт — это комплекс взаимосвязанных программ для решения определенной задачи массового спроса, подготовленный к реализации как любой образец промышленной продукции.

### **4.1.1. Классификация программного обеспечения по способу распространения**

Программный продукт, как правило, является результатом работы специалистов высокой квалификации, и, как продукт интеллектуального труда, охраняется законодательством. Большинство современных программных продуктов требует сопровождения. Сопровождение программного продукта предполагает поддержку его работоспособности, переход на его новые, усовершенствованные версии, внесение изменений, устранение

ошибок, обнаруженных при эксплуатации программ. Этими обстоятельствами обусловлено существование различных вариантов легального распространения программных продуктов.

**Freeware** — бесплатные программы. Они свободно распространяются, а их поддержка осуществляется самим пользователем.

**Shareware** — условно-бесплатные программы. Обычно такие программы с ограниченным набором функций или ограниченным временем использования распространяются бесплатно. Если пользователь пожелает продлить срок использования или получить полнофункциональную версию, то он должен заплатить определенную сумму собственнику программы.

**Коммерческие программы.** Такие программы распространяются путем продаж через официальную торговую сеть. Этот способ приобретения программного обеспечения в цивилизованном мире является общепринятым. Программное обеспечение, приобретенное официально, часто носит название лицензионного. Владелец такого ПО получает значительные права. К их числу относятся:

- гарантии качества и стабильной работы (отсутствие сбоев носителя, завершенность версии, полный комплект необходимых компонентов, работоспособность во всех режимах и т. д.);
- техническая поддержка по любым вопросам, связанным с установкой и эксплуатацией продукта;
- подробная сопроводительная документация, зачастую состоящая из нескольких книг, в число которых входит руководство пользователю;
- информация о новых продуктах и обновление текущих версий с большой скидкой;
- соответствие закону (для организаций это обстоятельство может быть проверено в любой момент).

Обычно различают две версии лицензионных программ: коробочная и OEM (Original Equipment Manufacturer). Коробочная версия продается как отдельный товар и представляет собой дистрибутив программы на каком-либо носителе (как правило, это компакт-диск) и комплект документации. Для того чтобы получить право на сопровождение этого ПО, необходимо после его установки на компьютер зарегистрироваться в установленном порядке.

ОЕМ-программы поставляются вместе с компьютером, на который они устанавливаются дилером при предпродажной подготовке.

#### **4.1.2. Классификация программного обеспечения по назначению**

По функциональному предназначению (по области применения) различают три класса программных продуктов [8]:

- системное программное обеспечение;
- прикладное программное обеспечение;
- системы программирования.

##### **4.1.2.1. Системное программное обеспечение**

Программы, относящиеся к данному классу, предназначены для управления работой аппаратной части компьютеров и компьютерных сетей. Они, как правило, не решают конкретных задач пользователя, но создают условия для их решения. С их помощью решаются следующие задачи:

- обеспечение устойчивой работы компьютера и вычислительной сети;
- создание условий для нормальной работы прикладных программ;
- диагностика и профилактика аппаратной части компьютера и вычислительной сети;
- выполнение вспомогательных технологических операций (резервное копирование, архивирование, восстановление файлов).

Все множество системных программ можно разделить на две большие группы: базовое и сервисное системное ПО. Базовое программное обеспечение — это минимальный набор программных средств, обеспечивающих работу компьютера [8]. Эти программы, как правило, поставляются вместе с компьютером. В этот подкласс ПО обычно включают операционные системы (ОС), операционные оболочки, сетевые операционные системы.

**Операционная система** — это комплекс программ, предназначенных для управления выполнением пользовательских программ, планирования и управления вычислительными ресурсами ПК.

**Операционные оболочки** — специальные программы, предназначенные для облегчения общения пользователя с командами операционной системы. Операционные оболочки имеют текстовый и графический варианты интерфейса конечного пользователя.

**Сетевые операционные системы** — комплекс программ, обеспечивающих обработку, передачу и хранение данных в сети.

К подклассу сервисного программного обеспечения относятся программы, предназначенные для обслуживания компьютера. Эти программы расширяют возможности базового ПО. Иногда некоторые сервисные программы включают в состав операционной системы, но, как правило, они устанавливаются дополнительно. По функциональному признаку среди сервисного ПО можно выделить:

- программы диагностики работоспособности компьютера;
- программы обслуживания дисков, обеспечивающие проверку качества поверхности магнитного диска, контроль сохранности файловой системы, сжатие дисков, создание страховых копий дисков, резервирование данных на внешних носителях и др.;
- антивирусные программы, обеспечивающие защиту компьютера, обнаружение и восстановление зараженных файлов;
- программы архивирования данных, которые обеспечивают процесс сжатия информации в файлах с целью уменьшения объема памяти для ее хранения;
- программы обслуживания сети.

Программы, служащие для выполнения вспомогательных операций обработки данных или обслуживания компьютеров (диагностики, тестирования аппаратных и программных средств, оптимизации использования дискового пространства, восстановления разрушенной на магнитном диске информации и т. п.), называют утилитами.

#### 4.1.2.2. Прикладное программное обеспечение

Прикладное ПО предназначено для решения прикладных задач, т. е. непосредственно обеспечивает выполнение необходимых пользователю задач на компьютере. Условно среди множества прикладных программ по их назначению можно выделить следующие подклассы:

- программы обработки текстов;
- графические редакторы;

- программы обработки фото- и видеоизображений;
- системы деловой и презентационной графики, позволяющие наглядно представить на экране различные числовые данные и зависимости между ними;
- системы научной и инженерной графики, позволяющие автоматизировать весь комплекс работ по созданию чертежей, схем, графиков, создавать объекты трехмерной графики и т. п.;
- электронные таблицы;
- системы управления базами данных;
- информационно-поисковые системы;
- сетевое программное обеспечение. Оно позволяет не только объединить компьютеры в сети различного ранга, но и облегчает навигацию по таким сетям. К нему же относятся и программы работы с электронной почтой, доступа к видеоконференциям, браузеры Интернет и т. п.;
- игровые программы.

Очень часто программы различных типов объединяются в интегрированные пакеты, что чрезвычайно упрощает обмен информацией между ними. Так, пакет Microsoft Office состоит из текстового редактора, простого графического редактора, электронной таблицы, системы разработки и управления базами данных, программы электронной почты, программы разработки Интернет-страниц, программы просмотра Интернет-страниц, программы создания презентационной графики.

Кроме того, даже отдельные программы часто включают в себя элементы программ другого типа. Так, почти все игровые программы снабжены сетевыми компонентами, позволяющими подключить к игре несколько компьютеров.

#### 4.1.2.3. Системы программирования

Данный класс ПО включает программы, позволяющие разрабатывать и отлаживать новое программное обеспечение. В состав современных систем обычно входят следующие основные части:

- средства создания пользовательского интерфейса;
- средства создания и редактирования текстов программ;
- библиотеки стандартных подпрограмм и функций для работы с графикой, в Интернете и пр.;
- программы тестирования и отладки.

Главной частью таких систем являются программы-трансляторы, принцип которых рассматривался выше. Как правило, эти системы работают в многооконном режиме работы, обеспечивающем возможность просматривать исходный текст, работать в конструкторе пользовательского интерфейса, пользоваться справочной системой и т. п.

## 4.2. Общие сведения об операционных системах

### 4.2.1. Назначение операционных систем

Вычислительная система состоит из трех элементов: компьютера (аппаратная часть), программного обеспечения и пользователя. Операционная система представляет собой комплекс программ, который объединяет эти элементы в единую систему, обеспечивая их взаимодействие. Для выполнения своего предназначения операционная система решает следующие основные задачи:

- управление устройствами компьютера;
- управление вычислительным процессом, предполагающее распределение памяти, обмен информацией с внешними устройствами, реакцию на ошибочные ситуации;
- предоставление пользователю возможности управления компьютером;
- управление размещением и доступом к папкам и файлам на диске.

Рассмотрим подробнее сущность этих задач. Средства, обеспечивающие взаимодействие между элементами компьютерной системы, называют интерфейсом. Таким образом, в вычислительной системе реализуется несколько видов интерфейсов:

- аппаратный интерфейс — взаимодействие между различными узлами компьютера;
- программный интерфейс — взаимодействие между программами, выполняемыми компьютером;
- аппаратно-программный интерфейс — взаимодействие между узлами компьютера и аппаратурой и выполняемыми программами;
- интерфейс пользователя — взаимодействие пользователя с аппаратно-программным комплексом.

Аппаратный интерфейс в компьютере обеспечивают изготовители оборудования. Они следят за тем, чтобы все узлы имели совместимые разъемы и работали с одинаковыми напряжениями. Но поскольку компьютеры могут иметь различную конфигурацию, на них может быть установлено различное программное обеспечение, постольку возникает необходимость в обеспечении аппаратно-программного, программного и пользовательского интерфейсов. Согласование между программным и аппаратным обеспечением, а также между компьютером и пользователем организует и осуществляет операционная система.

Не менее важной задачей является организация хранения информации в памяти компьютера. Эту задачу также решает операционная система.

#### **4.2.2. Классификация операционных систем**

Для классификации операционных систем может быть использовано несколько признаков. В настоящее время чаще всего используются следующие:

- 1) количество одновременно работающих пользователей;
- 2) число задач, одновременно выполняемых под управлением ОС;
- 3) тип пользовательского интерфейса;
- 4) способ использования аппаратных и программных ресурсов;
- 5) количество используемых в компьютере процессоров;
- 6) разрядность процессора.

Рассмотрим некоторые из особенностей ОС в зависимости от того или иного классификационного признака. В зависимости от количества пользователей, которые могут для решения своих задач одновременно обратиться к одному и тому же компьютеру, различают однопользовательские и многопользовательские ОС. Главным отличием многопользовательских ОС от однопользовательских является наличие средств защиты информации каждого пользователя от несанкционированного доступа других пользователей. Каждому пользователю выделяется свой сегмент оперативной памяти.

По числу задач, одновременно выполняемых под управлением ОС, они делятся на однозадачные и многозадачные. В многозадачном режиме каждой задаче (программе, приложению) по-



очередно выделяется какая-то доля процессорного времени. Поскольку процесс переключения идет очень быстро, а выделяемые задачам доли процессорного времени достаточно малы, то у пользователя создается впечатление одновременного выполнения сразу нескольких задач. При использовании многозадачной ОС можно одновременно организовать вычисления в среде табличного процессора, включить принтер для печати текста, запустить проигрыватель музыкальных произведений, вести поиск вирусов и рисовать в графическом редакторе.

Современные операционные системы обеспечивают реализацию двух типов пользовательского интерфейса — текстового и графического. Текстовые (неграфические) операционные системы предоставляют пользователю интерфейс командной строки. Основным устройством управления в данном случае является клавиатура. Управляющие команды вводят в поле командной строки, где их можно и редактировать. Исполнение команды начинается после ее утверждения, например нажатием клавиши **ENTER**.

Графические операционные системы дают возможность использовать более сложный тип интерфейса, в котором в качестве органа управления, кроме клавиатуры, может использоваться устройство позиционирования (мышь, трекбол, сенсорная панель). Работа с графической операционной системой основана на взаимодействии активных и пассивных экранных элементов управления. В качестве активного элемента управления выступает указатель манипулятора — графический объект, перемещение которого на экране синхронизировано с перемещением органа управления. В качестве пассивных элементов управления выступают графические объекты на экране монитора (экранные кнопки, значки, переключатели, флажки, раскрывающиеся списки, строки меню и многие другие). Характер взаимодействия между активными и пассивными элементами управления выбирает сам пользователь.

По способу использования аппаратных и программных ресурсов различаются сетевые и локальные операционные системы. Сетевые операционные системы предназначены для эффективного решения задач распределенной обработки данных. Такая обработка ведется не на отдельном компьютере, а на нескольких компьютерах, объединенных сетью. Сетевые операционные системы поддерживают распределенное выполнение процессов, их взаимодействие, обмен данными между ЭВМ,

доступ пользователей к общим ресурсам и другие функции, которые превращают распределенную в пространстве систему в целостную многопользовательскую систему.

Все сетевые операционные системы делятся на две группы: одноранговые ОС и ОС с выделенными серверами. В одноранговых сетях каждая ЭВМ может выполнять как функции сервера, так и рабочей станции, а в сетях с выделенными серверами реализуется разделение функций: рабочие станции не предоставляют свои ресурсы для других ЭВМ. Услуги предоставляют только серверы.

Что касается других классификационных признаков, то здесь можно ограничиться только констатацией факта, что по количеству используемых в компьютере процессоров ОС могут быть одно- и многопроцессорными; а по разрядности процессора — 8-, 16-, 32- и 64-разрядными.

Очевидно, что процесс совершенствования аппаратных средств сопровождался эволюцией операционных систем. Из всего множества наибольшую известность получили следующие ОС: CP/M, MS-DOS, OS/2, Windows, UNIX и MacOS (для компьютеров Macintosh фирмы Apple).

ОС CP/M была создана для работы на 8-разрядных ПЭВМ. Первоначальный успех этой операционной системы в значительной степени был обусловлен ее предельной простотой и компактностью. Первая версия занимала всего 4 Кбайт. Компактность была весьма важна в условиях ограниченных объемов памяти первых персональных ЭВМ (ПЭВМ).

Операционная система MS-DOS (фирма Microsoft) появилась в 1981 г. В настоящее время существуют версии 6.22 и 7.0 (в составе Windows 95), а также ее разновидности других фирм-разработчиков (DR DOS, PC DOS). Начиная с 1996 г. MS-DOS распространяется в виде Windows 95 — 32-разрядной многозадачной и многопоточной операционной системы с графическим интерфейсом и расширенными сетевыми возможностями. Операционная система MS-DOS является промышленным стандартом для 16-разрядных ЭВМ на основе микропроцессоров 8086...80486. Все программы MS-DOS хранятся на магнитных дисках, поэтому она называется дисковой операционной системой (Microsoft Disk Operating System).

Операционная система OS/2 разработана фирмой IBM для персональных компьютеров на основе системной прикладной архитектуры, ранее используемой для больших ЭВМ. Это много-

задачная, однопользовательская, высоконадежная операционная система, обеспечивающая как текстовый, так и графический интерфейс пользователя, одновременную обработку нескольких приложений, 32-разрядную обработку данных. Важной особенностью операционной системы OS/2 является высокопроизводительная файловая система HPFS (High Performance File System), имеющая преимущества для серверов баз данных (в отличие от MS-DOS поддерживаются длинные имена файлов), поддержка мультипроцессорной обработки — до 16 процессоров типа INTEL и PowerPC. Версия OS/2 Warp работает с мультисредой и имеет встроенный доступ в сеть Интернет, систему распознавания речи Voice Type, интегрированную версию Lotus Notes Mail для передачи почты через Интернет.

Перспективной является многопользовательская и многозадачная операционная система Unix, созданная корпорацией Bell Laboratory. Данная операционная система реализует принцип открытых систем и широкие возможности по комплексированию в составе одной вычислительной системы разнородных технических и программных средств. Unix обладает наиболее важными качествами, такими, как переносимость прикладных программ с одного компьютера на другой и поддержка распределенной обработки данных в сети ЭВМ. Unix получила распространение для суперкомпьютеров, рабочих станций и профессиональных персональных компьютеров, имеет большое количество версий, разработанных различными фирмами. Согласно прогнозам объем мирового рынка вычислительных систем, базирующихся на ОС Unix, существенно будет возрастать, особенно с переходом к сетевым технологиям.

Операционная система MacOS была разработана для компьютеров Macintosh фирмы Apple. В настоящее время эти компьютеры с MacOS в России используются в основном для так называемой допечатной подготовки полиграфической продукции — книг, журналов, газет. Программы с мощными средствами обработки графики Adobe Photoshop, Adobe Illustrator, Adobe Pagemaker в свое время разрабатывались для этой ОС и только впоследствии были адаптированы под Windows.

Среди сетевых операционных систем в течение длительного времени наиболее широко используемыми были различные версии операционной системы Netware, разработанные фирмой Novell. К одноранговым операционным системам можно отне-

сти NetWare Lite и Personal NetWare, а к ОС с выделенным сервером — NetWare 2.2, NetWare 3.12, NetWare 4.0 и NetWare 5.0.

Фирма Microsoft выпустила несколько версий сетевых операционных систем: Windows NT 3.51 и 4.0, Windows 2000/ME/CE/XP.

Центральное место среди сетевых операционных систем занимает UNIX. Большая популярность пришла к UNIX в 1983 г., когда появилась версия 4.2BSD, имевшая сетевые средства TCP/IP, что позволяет использовать эту систему для работы в глобальной сети ARPANET. Классическая ОС UNIX дала жизнь многочисленным своим потомкам, число которых превышает несколько десятков (AIX, SCO, HP-UX, IRIX, Solaris, Linux и др.).

Особого упоминания заслуживает ОС Linux — свободно распространяемая версия операционной системы UNIX для платформ x86, Motorola 68k, Digital Alpha, Spare, Mips и Motorola PowerPC. В Linux не используются никакие части программного обеспечения, принадлежащие каким-либо коммерческим организациям. По этой причине она получила достаточно широкое распространение.

### 4.2.3. Состав операционных систем

Как правило, операционная система содержит в себе следующие компоненты: ядро, файловую систему, диспетчер задач (или планировщик задач), драйверы устройств и различные сервисные программы, упрощающие обслуживание, наладку и оптимизацию работы внешних устройств или оптимизирующие работу самой ОС.

#### 4.2.3.1. Ядро операционной системы

Ядро операционной системы состоит из базовой системы ввода и вывода (BIOS — basic input/output system), модуля расширения базовой системы ввода-вывода (BIO.COM), модуль обработки прерываний (DOS.COM) и командного процессора (COMMAND.COM).

В архитектуре ПЭВМ базовую систему ввода-вывода (BIOS) можно рассматривать, с одной стороны, как составную часть аппаратных средств; с другой стороны, BIOS является одним из

программных модулей ДОС. BIOS находится не на дисках, как все остальные модули, а в ПЗУ, которое установлено внутри системного блока. BIOS обеспечивает:

- автоматическое тестирование основных аппаратных компонентов при включении машины. При этом проверяются большие интегральные схемы (БИС) системного модуля, ОЗУ, клавиатура и ее адаптер, адаптеры накопителей на магнитных дисках, адаптеры параллельного и последовательного интерфейсов. Если обнаруживаются ошибки, BIOS выдает на экран соответствующие сообщения (обычно в виде условного кода ошибки) и еще извещает об этом пользователя звуковым сигналом. Дальнейшая работа машины при этом прекращается и пользователю нужно принимать меры к устранению выявленной ошибки;
- вызов блока начальной загрузки DOS. Загрузка DOS в память происходит в два этапа; сначала BIOS загружает с системного диска в память специальный блок начальной загрузки, а затем уже передает на него управление, и тот, в свою очередь, осуществляет загрузку всех остальных модулей DOS. Блок начальной загрузки предназначен для просмотра каталога системного диска и проверки того, что первые два файла являются модулями DOS. При попытке запуска системы с несистемного диска данный блок выдает сообщение об ошибке;
- обслуживание системных вызовов или прерываний.

Поскольку BIOS очень тесно связана с особенностями аппаратуры конкретного компьютера, она является общей и неизменяемой частью всех возможных операционных систем для данной модели ПЭВМ. Поэтому для обеспечения возможности управления с помощью BIOS определенным набором аппаратных средств создается дополнительный модуль, придающий гибкость операционной системе.

**Модуль расширения BIOS** — это обычная программа, которая при необходимости может быть заменена другой. Наличие модуля расширения дает возможность включения в состав ОС дополнительных подпрограмм, обслуживающих новые внешние устройства.

**Модуль обработки прерываний DOS** образует верхний уровень системы, с которым взаимодействует большинство прикладных программ. По этой причине этот модуль ДОС называют основным. Компонентами данного модуля являются подпрограммы,

обеспечивающие работу файловой системы, устройств, обслуживание некоторых специальных ситуаций, связанных с завершением программ, их искусственным прерыванием и обработкой ошибок. Некоторые из этих подпрограмм довольно велики по объему. Многие из функций, реализуемых данным модулем DOS, используются не только прикладными программами, но и командами, которые обрабатываются командным процессором.

**Командный процессор** представляет собой программу, осуществляющую перевод команд с языка программирования на язык машинных кодов. Основная функция командного процессора заключается в приеме, анализе и исполнении команд пользователя, обращенных к DOS, и в обработке командных файлов.

Команды пользователя иначе называют командами DOS. Они служат основным средством общения пользователя с ОС до тех пор, пока не будет вызвана какая-либо прикладная программа (задача), или «надстройка». Команды DOS позволяют готовить диски для работы, копировать файлы, переименовывать их, удалять из каталогов, сменять текущий каталог и текущий накопитель, изменять режим работы дисплея, выводить содержимое текстовых файлов на экран дисплея, на принтер или в коммуникационный канал.

#### 4.2.3.2. Файловая система

Файловая система — совокупность программ, входящих в состав операционной системы, которая управляет размещением и доступом к папкам и файлам на диске. Она обеспечивает подготовку магнитных носителей к работе (разбиение, форматирование, проверка), размещение на носителях файлов и обеспечение доступа к ним различных программ. Файловая система включает в себя программы доступа к магнитным носителям и специальные таблицы, в которых хранится информация о том, на какие части разбит диск, где на диске располагаются файлы и т. п. Файловая система во многом определяет скорость и эффективность работы магнитных носителей, надежность и секретность хранения информации на них.

Записать информацию на магнитный диск можно только тогда, когда он подготовлен определенным образом — отформатирован. Во время форматирования производится разметка диска, в процессе которой на него записывается служебная информация, которая затем используется для записи и чтения команд и

данных, коррекции скорости вращения диска. Запись информации осуществляется по дорожкам, причем каждая дорожка разбивается на секторы размером 512 байт.

В процессе форматирования на диске выделяется системная область, которая состоит из трех частей: загрузочного сектора, таблицы размещения файлов и корневого каталога.

**Загрузочный сектор** (Boot Record) размещается на каждом диске в логическом секторе с номером 0. Он содержит данные о формате диска, а также блок начальной загрузки DOS. Каждый жесткий диск может быть разбит на несколько логических дисков. На жестком диске имеется область, которая называется главной загрузочной записью MBR (Master Boot Record) или главным загрузочным сектором. В MBR указывается, с какого логического диска должна производиться загрузка операционной системы.

**Таблица размещения файлов** (File Allocation Table — сокращенно FAT) располагается после загрузочного сектора и содержит описание порядка расположения всех файлов в секторах данного диска, а также информацию о дефектных участках диска. За FAT-таблицей следует ее точная копия, что повышает надежность сохранения этой очень важной таблицы.

Необходимость в таблице расположения файлов вызвана тем, что данные и программы хранятся на носителях информации в виде файлов. У каждого файла есть свой адрес, который записывается двоичным числом. Очевидно, что количество адресов, которые могут быть присвоены файлам, зависит от разрядности этого числа. В ряде операционных систем для записи адреса файла использовались двухбайтные (16-разрядные) двоичные числа. Известно, что с помощью 16 битов можно выразить  $2^{16}$  (65 536) разных значений. В этом случае файлам на жестком диске не может быть предоставлено более чем 65 536 разных адресов, следовательно, на такой диск нельзя записать более 65 536 файлов. Необходимость присвоения каждому файлу собственного уникального адреса привела к появлению нового понятия «минимальный размер адресуемого пространства». Для обозначения этого пространства используется термин «**кластер**». Очевидно, что размер кластера зависит от объема диска и количества адресов, используемых для записи файлов (табл. 4.2, 4.3). Иногда для обозначения файловой системы используют разрядность адресов FAT. Так, достаточно широко распространены файловые системы FAT-16 и FAT-32.

Таблица 4.2. Размер кластера при использовании FAT-16

Объем диска	Размер кластера
Менее 32 Мбайт	512 байт
32 Мбайт...64 Мбайт	1 Кбайт
64 Мбайт...128 Мбайт	2 Кбайт
128 Мбайт...256 Мбайт	4 Кбайт
256 Мбайт...512 Мбайт	8 Кбайт
512 Мбайт...1 Гбайт	16 Кбайт
1 Гбайт...2 Гбайт	32 Кбайт

Таблица 4.3. Размер кластера при использовании FAT-32

Объем диска	Размер кластера
513 Мбайт...8 Гбайт	4 Кбайт
8 Гбайт...16 Гбайт	8 Кбайт
16 Гбайт...32 Гбайт	16 Кбайт
Более 32 Гбайт	32 Кбайт

Наличие FAT приводит к существованию противоречия: с одной стороны, чем меньше кластеры, тем их больше на диске и тем больше требуется места для хранения самой таблицы; с другой — чем больше кластеры, тем меньше таблица, но зато тем больше «теряется» дискового пространства, потому что размер файла обязан быть кратен размеру кластера. Это не вполне рациональный расход рабочего пространства, поскольку любой файл (даже очень маленький) полностью оккупирует весь кластер, которому соответствует только одна адресная запись в таблице размещения файлов. Даже если файл достаточно велик и располагается в нескольких кластерах, все равно в его конце образуется некий остаток, нерационально расходующий целый кластер.

Для современных жестких дисков потери, связанные с неэффективностью файловой системы, весьма значительны и могут составлять от 25 до 40 % полной емкости диска в зависимости от среднего размера хранящихся файлов. С дисками же размером более 2 Гбайт файловая система FAT-16 вообще работать не может.



Начиная с операционной системы Windows 98 используется более совершенная организация файловой системы — FAT32 с 32-разрядными полями в таблице размещения файлов. Для дисков размером до 8 Гбайт эта система обеспечивает размер кластера 4 Кбайт (8 секторов).

**Корневой каталог** (Root Directory) всегда находится за копией FAT. В корневом каталоге содержится перечень файлов и директорий, находящихся на диске. Непосредственно за корневым каталогом располагаются данные, организованные в файлы. **Файл** — это набор взаимосвязанных данных, воспринимаемых компьютером как единое целое, имеющих общее имя, находящихся на магнитном или оптическом диске, магнитной ленте или на другом носителе информации. Файл обычно отождествляют с участком памяти (ВЗУ, ОЗУ, ПЗУ), где размещены логически связанные данные, имеющие общее имя. Файл записывается на носителе информации в двоичных кодах. Для операционной системы файл представляется как совокупность связанных байтов.

#### 4.2.3.3 Файлы и их характеристики

Файлы могут содержать в себе любую информацию. Это могут быть как программы, выполняемые под управлением какой-либо операционной системы, либо файлы с данными для этих программ. Независимо от операционных систем персональных компьютеров все файлы можно разделить на текстовые и бинарные (двоичные) файлы. Текстовыми файлами называют файлы, в которых используются в качестве информационных символы с шестнадцатеричными кодами 20h—7Eh (32—126 десятичными) и 80h—7Eh (128—254 десятичными).

Среди всех текстовых файлов можно выделить подмножество чистых ASCII-файлов, информационные символы которых имеют только коды с номерами 20h—7Eh. Двоичные же файлы представляют собой последовательность любых символов.

С точки зрения содержания записанной в них информации файлы можно разделить на исполняемые (программы) и неисполняемые (файлы данных и документов). Исполняемые файлы могут запускаться операционной системой на выполнение, а неисполняемые файлы могут только изменять свое содержимое в процессе выполнения прикладных программ, с помощью которых они были созданы.

Кроме того, файлы можно разделить на:

- основные, наличие которых обязательно для работы как операционной системы, так и программ, работающих под ее управлением;
- служебные, хранящие конфигурацию и настройки основных файлов;
- рабочие, которые создаются и обрабатываются прикладными программами;
- временные файлы, создающиеся в момент работы прикладных программ и хранящие промежуточные результаты.

Следует отметить, что существует специальный вид файла, называемый папкой (directory) или каталогом (catalog). В таком файле содержатся ссылки на другие файлы. Поскольку ссылки на эти файлы содержатся лишь в одной из папок, для пользователя эти файлы как бы расположены в этой папке. На самом деле, конечно, все файлы находятся в секторах диска. Но это истинно лишь на физическом уровне, а на уровне представления данных файлы находятся в папках. Папки используются для упорядочения программ и документов на диске и могут вмещать как файлы, так и другие папки. Папки появились не сразу и не во всех операционных системах. Они возникли там, где требовался большой объем хранимой информации (например, в файловых системах жестких дисков) и, следовательно, возникли сложности с организацией и размещением большого числа файлов.

Любой файл имеет служебные поля, в которых записывается информация о нем. Она включает имя, дату и время создания и модификации, свой размер и атрибуты: только для чтения — **Read only**; скрытый файл — **Hidden**; системный файл — **System**; архивный файл — **Archive**.

То же относится и к папкам. Имя и тип любого файла (и папки) должны быть уникальными в пределах той папки, в которой они находятся. Операционные системы предъявляют определенные требования к именам, которые присваиваются файлам.

#### 4.2.3.4. Другие компоненты операционной системы

**Диспетчер задач** (или планировщик задач), обеспечивает запуск различных заданий и распределение между ними ресурсов компьютера.

**Драйверы устройств** — специальные программы, управляющие работой устройств компьютера. Помимо стандартных (внут-

ренных) драйверов существуют и загружаемые (внешние) драйверы в виде исполняемых файлов. Главное назначение внешних драйверов — обслуживание дополнительного (нестандартного) оборудования, установленного на ПЭВМ. Как правило, эти драйверы запускаются сразу после загрузки ОС. Подключая к ПЭВМ новое внешнее устройство (например, звуковую плату), необходимо установить поддерживающее его программное обеспечение (соответствующие драйверы).

**Утилиты** — это различные сервисные программы, упрощающие обслуживание, наладку и оптимизацию работы внешних устройств или оптимизирующие работу самой ОС. Многие из этих программ дублируются программами независимых производителей, которые зачастую удобнее и производительнее того, что предлагается в комплекте той или иной ОС.

### **4.3. Особенности операционной системы MS Windows**

Операционные системы, разработанные фирмой Microsoft, появились на рынке программных продуктов в 1981 г. Первой из них была MS-DOS — однопользовательская, однозадачная, локальная ОС, с интерфейсом командной строки. Операционная оболочка Windows 3.0 (1992 г.) обеспечила графический пользовательский интерфейс, а последующие ее версии 3.1 и 3.11 (1994 г.) — поддержку мультимедиа и работу в сети. Одновременно с этим фирма разрабатывала и другие операционные системы, предназначенные для работы в локальных компьютерных сетях. Первой из таких ОС была Windows NT (1993 г.).

В 1995 г. была создана графическая, многозадачная 32-разрядная ОС Windows 95. Принципиальная особенность этой ОС заключалась в том, что она могла автоматически распознать большое число комплектующих (технология Plug & Play). Дальнейшая работа привела к созданию ОС Windows 98, которая отличалась от предыдущей версии наличием встроенного браузера (Internet Explorer), а также она могла корректно работать с новыми комплектующими — процессором Pentium II, новыми моделями видеокарт, материнских плат, шинами USB. Кроме того, эта ОС могла обеспечить работы с файловыми системами FAT-16 и FAT-32.

В настоящее время широкое распространение получили новые версии сетевых и локальных операционных систем этой фирмы — Windows 2000/ME/XP.

#### 4.3.1. Общая характеристика операционной системы Windows XP

Windows XP является многопользовательской, многозадачной, сетевой операционной системой с графическим интерфейсом. В этой ОС могут быть использованы три варианта файловой системы: FAT-16, FAT-32 и NTFS.

Данная ОС предоставляет возможность использования одного компьютера разными пользователями, обеспечивая конфиденциальность информации каждого из них. Это свойство обеспечивается наличием средств защиты информации каждого пользователя от несанкционированного доступа других пользователей. Каждому пользователю выделяется свой сегмент памяти.

Возможность одновременного выполнения нескольких задач обеспечивается планировщиком задач, который управляет распределением оперативной памяти. В процессе работы компьютера в оперативной памяти выделяются области, предназначенные для размещения в них необходимых для выполнения в данное время файлов операционной системы и соответствующих программ. При этом постоянно обеспечивается постоянный обмен информацией между оперативной и внешней памятью. Для документов, обрабатываемых с помощью того или иного приложения, также выделяются соответствующие области оперативной памяти. Кроме того, ОС Windows выделяет специальную область оперативной памяти, в которой временно хранятся отдельные объекты при выполнении команд **Правка/копировать** или **Правка/вырезать**. Эта область называется буфером памяти. Из буфера памяти этот объект может быть помещен в любой другой документ, открытый в данное время, по команде **Правка/вставить**. Содержимое буфера памяти обновляется при выполнении последующего копирования (вырезания). На рис. 4.2 представлена схема распределения оперативной памяти и взаимодействия двух видов памяти в процессе работы. Предполагается, что данная схема отражает такое состояние устройств памяти компьютера, когда запущены два приложения, например графический редактор MS Paint (ПП1) и текстовый процессор MS Word (ПП2), и

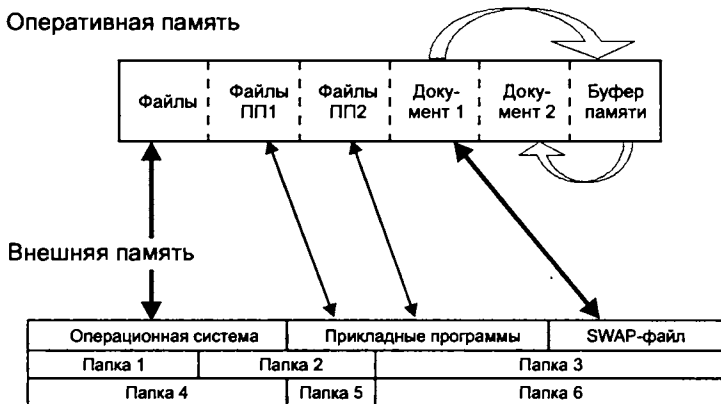


Рис. 4.2. Схема распределения оперативной памяти

по одному документу в каждом из них. При этом объект из графического документа 1 (фрагмент рисунка) копируется и переносится в текстовый документ 2.

Особого упоминания заслуживает тот факт, что при недостатке объема оперативной памяти на жестком диске выделяется специальная область, называемая файлом подкачки (SWAP-файл), куда временно помещаются данные, к которым программа обращается редко.

Для хранения программ и данных в ОС Windows XP используется иерархическая структура. При этом следует помнить, что файл является наименьшей единицей хранения, а для удобства их хранения и поиска файлы объединяются в папки. Эту структуру создает программа установки операционной системы. Вершиной (элементом высшего, нулевого уровня) является папка **Рабочий стол** (рис. 4.3, где представлен фрагмент файловой системы некоторого компьютера). Основные вершины следующего (первого) уровня представлены системными папками **Мой компьютер**, **Мои документы**, **Сетевое окружение** и **Корзина**. Рассмотрим подробнее назначение этих папок.

**Мой компьютер.** В эту папку помещаются корневые каталоги [1] всех устройств внешней памяти компьютера и системные папки второго уровня — **Панель управления**, **Принтеры** и **Удаленный доступ**. В этих папках размещаются служебные программы настройки компьютера, установки принтеров и управления ими и подключения к Интернет с использованием выделенного канала.

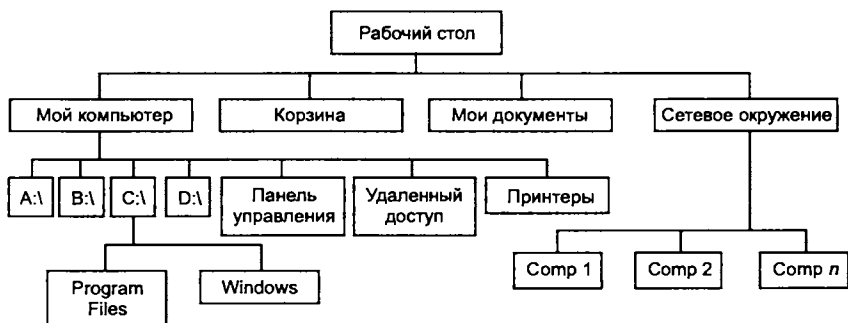


Рис. 4.3. Структура файловой системы ОС Windows XP

**Корзина.** Данная папка предназначена для временного хранения удаленных файлов и папок. В операционной системе Windows XP предусмотрено удаление файлов и папок в два этапа. По команде **Правка/удалить** выделенный объект помещается в **Корзину**. Окончательное удаление этого объекта осуществляется по команде **Очистить корзину**. Такой порядок удаления позволяет пользователю восстановить любой объект, ошибочно или преждевременно помещенный в папку.

**Мои документы.** В данную папку помещаются документы, создаваемые с помощью какого-либо приложения, по команде **Сохранить**, если пользователь не указал ранее другое место для их хранения.

**Сетевое окружение.** Это папка программы, позволяющей установить связь между компьютерами, включенными в одну локальную сеть.

Что касается файловой системы, то Windows XP предоставляет пользователю возможность выбрать тот или иной вариант файловой системы, каждый из которых имеет свои особенности (табл. 4.4).

Следует отметить, что в состав ОС семейства MS Windows включен ряд прикладных программ и утилит, позволяющих пользователю выполнять некоторые прикладные задачи и обслуживать компьютер. Так, текстовые редакторы **Блокнот** и **WordPad** позволяют создавать текстовые документы, не требующие сложного форматирования. Графический редактор **Paint** дает пользователю возможность обрабатывать графические файлы. С помощью **Калькулятора** можно выполнять достаточно сложные вычисления. Для удобства управления файловой системой служит

Таблица 4.4. Особенности файловых систем

NTFS	FAT16	FAT32
Компьютеры, работающие под управлением Windows XP или Windows 2000, имеют доступ к файлам, находящимся в разделе NTFS. Компьютер, работающий под управлением Windows NT 4.0 с пакетом обновления версии 4 или более поздней, может получить доступ к некоторым файлам. Для других операционных систем доступ отсутствует	Доступ возможен из MS-DOS, любых версий Windows, Windows NT, Windows 2000, Windows XP и OS/2	Доступ возможен только из Windows 95 OSR2, Windows 98, Windows Millennium Edition, Windows 2000 и Windows XP
Рекомендуемый минимальный размер тома равен примерно 10 Мбайт. Допускаются размеры томов свыше 2 Тбайт. Не может использоваться для гибких дисков	Тома объемом от 1,44 Мбайт (гибкий диск) до 4 Гбайт. Домены не поддерживаются	Тома объемом от 512 Мбайт до 2 Тбайт. Windows XP позволяет форматировать в FAT32 тома объемом только до 32 Гбайт. Домены не поддерживаются
Максимальный размер файла ограничен только размером тома	Максимальный размер файла равен 2 Гбайт	Максимальный размер файла равен 4 Гбайт

программа **Проводник**, представляющая собой диспетчер файлов (File Manager).

Из числа служебных программ в составе ОС можно отметить программы форматирования, проверки и очистки дисков, архивации и резервного копирования.

#### 4.3.2. Особенности технологии работы в среде Windows

При создании операционной системы семейства Windows фирма Microsoft использовала **объектно-ориентированный** подход. Для пользователя это выражается в том, что интерфейс представляет собой подобие рабочего стола, на котором расположены определенные объекты, обладающие соответствующими свойствами. При этом для решения задачи пользователь должен выполнить действия над этими объектами. Основными объектами являются файлы и папки. Так, папки можно открывать,

закрывать, перемещать. Документы можно просматривать, исправлять, переключать с одного места на другое, уничтожать или выбрасывать в корзину. Особым типом файла, используемым в ОС Windows, является ярлык, который представляет собой ссылку для загрузки в систему другого объекта — файла или каталога. Ярлык является мощным средством для обустройства под свои нужды среды Windows.

Достоинством ОС семейства MS Windows является **унифицированный пользовательский интерфейс** (оболочка), благодаря которому в различных программах сохраняются одинаковые принципы управления их работой. При этом пользовательский интерфейс каждой программы размещается в отдельной прямоугольной области, которая называется **окном**. Окна также являются объектом, свойства которого позволяют перемещать его по поверхности рабочего стола и изменять размеры. Унификация пользовательского интерфейса (оболочки) программ, работающих под управлением операционных систем этого семейства, значительно упрощает освоение новых программ. Например, в любой программе, работающей под управлением этих ОС, нажатие клавиши **F1** вызывает помощь, а сочетание клавиш **Alt + F4** приводит к завершению (закрытию) работы программы.

Реализация принципа **Plug and Play** (подключи и используй) позволяет без ручной настройки подключать новые устройства к компьютеру. ОС семейства MS Windows, поддерживающие этот принцип, автоматически подбирают драйвер, необходимый для работы нового подключенного к ЭВМ устройства, например принтера или накопителя на оптических дисках.

В Windows используется технология **Drag and Drop** (перетащи и положи). Эта технология позволяет легко изменить положение любого окна и его размеры. Кроме того, эту технологию можно использовать для удаления объектов. Для этого достаточно удаляемый объект перетащить в папку **Корзина**.

Наличие буфера обмена обусловило возможность реализации в ОС семейства MS Windows технологии **OLE** (Object Linking and Embedding), что означает «связывание и внедрение объектов». Эта технология применяется для включения в документ, созданный с помощью одного приложения, объектов из другого. При этом приложение, в котором создается объект, называется родительским, приложением-источником или приложением-сервером. Приложение, в котором размещается объект, называется приложением-приемником или приложением-клиентом.



Технология OLE предполагает два варианта ее возможного использования. Первый заключается в установлении связи документа, созданного с помощью приложения-клиента с объектом из родительского приложения, а второй — во внедрении (встраивании) объекта из родительского документа в документ-приемник. Основное различие между связыванием и внедрением объектов состоит в способе хранения объектов, созданных в родительском приложении.

В первом случае объект хранится в файле, созданном в родительском приложении. В документе-приемнике сохраняются лишь данные о местоположении объекта и данные, необходимые для визуального (графического) представления объекта на экране. Преимущества этого способа заключаются в следующем:

- увеличение объема документа-приемника незначительно;
- имеется возможность установления множественных связей с одним объектом — один экземпляр объекта служит при этом источником данных для многих документов.

Недостаток этого способа в том, что при переносе файла-приемника на другой носитель эта связь может быть потеряна, что приведет к утрате возможности воспроизведения связанного объекта.

При использовании внедрения механизм OLE включает в составной документ копию оригинального объекта или его фрагмента вместе с данными для управления им, при этом внедренный объект становится частью документа-приемника. В этом случае связь копии объекта с ее оригиналом не поддерживается, но сохраняется связь приложения-клиента с приложением-сервером. Это дает возможность при необходимости внесения изменений во внедренный объект активизировать связь между приложениями и редактировать объект средствами приложения-сервера. Внедрение объектов в составной документ приводит к значительному увеличению размера последнего. Тем не менее внедрение имеет некоторые преимущества по сравнению со связыванием:

- порожденный документ с внедренными объектами может быть перенесен на другой компьютер, где его можно открыть в своем приложении;
- внедренный объект редактируется родительским приложением по месту его нахождения в приложении-клиенте.

### 4.3.3. Особенности интерфейса операционной системы Windows XP

Внешне графический интерфейс операционной системы Windows XP, отображаемый на мониторе компьютера, выглядит как поверхность рабочего стола. Рабочий стол — это графическая среда, на которой отображаются объекты и элементы управления. Нижняя часть рабочего стола, представляющая собой полосу относительно небольшой ширины, называется панелью задач и является одним из основных элементов управления. Ширина и положение панели задач могут быть изменены пользователем.

Объекты, размещаемые на рабочем столе, составляют два основных класса — пиктограммы и окна. Пиктограмма представляет собой графическую миниатюру, соответствующую папке или файлу, непосредственно подчиненному вершине **Рабочий стол**. На примере (рис. 4.4) представлены образцы пиктограмм папок **Мои документы**, **Мой компьютер**, **Сетевое окружение**, **Корзина**, программы **Internet Explorer**. При настройке операционной системы пользователь может вдобавок к системным объектам разместить и другие. Особую разновидность таких объектов представляют ярлыки. Их отличают от остальных наличие черной стрелки в белом квадрате, расположенном в левом нижнем углу пиктограммы (см. ярлык программы Ad-Aware SE Personal на рис. 4.4). Ярлык соответствует специальному файлу с расширением **.lnk**. Этот файл находится в папке **Рабочий стол** и содержит ссылку на соответствующую папку или файл, расположение которых определяется операционной системой при их установке (для программ) или пользователем при создании документов или папок для их хранения. Наличие пиктограмм позволяет пользователю запускать программы или открывать файлы непосредственно с рабочего стола.



Рис. 4.4. Образцы пиктограмм

На панели задач расположены: кнопка **Пуск**; область отображения задач; область уведомлений.

Основным способом управления работой операционной системы является выбор одной из команд, представленных в перечне, называемом меню. Сама операционная система и программы, работающие под ее управлением, имеют главное меню, в котором содержатся основные команды, управляющие данной программой. Кроме главного меню той или иной программы, нажатием правой кнопки манипулятора можно открыть контекстное меню. Такое название этот вид меню получил потому, что его содержание зависит от положения курсора в момент его вызова.

**Кнопка Пуск** является элементом управления и содержит главное меню операционной системы, которое раскрывается при нажатии на эту кнопку. С помощью команд, содержащихся в главном меню ОС Windows, можно выполнять следующие действия:

- запускать программы;
- открывать файлы;
- настраивать систему с помощью панели управления;
- получать необходимые сведения о работе ОС с помощью команды **Справка и поддержка**;
- находить файлы и папки на своем компьютере или в Интернете с помощью команды **Поиск**.

Рядом с некоторыми пунктами главного меню отображается направленная вправо стрелка. Это означает наличие еще одного, вложенного, меню. Если поместить указатель на такой пункт меню, раскроется другое меню.

Другой разновидностью команд меню являются такие, которые сопровождаются многоточием. Для исполнения таких команд программе, содержащей ее, необходимы дополнительные сведения. Если для выполнения выбранной команды программе требуются какие-либо дополнительные сведения, будет отображено диалоговое окно. Эти сведения пользователь должен ввести в специальные поля, содержащиеся в диалоговых окнах, открывающихся после выбора такой команды. Для ввода сведений может потребоваться выполнение одного из следующих действий в зависимости от того, какие элементы управления расположены в этом диалоговом окне:

- для ввода данных в текстовое поле необходимо указать его и ввести текст;
- для выбора одного из значений в списке следует сначала нажать на кнопку со стрелкой, чтобы открыть список, а затем выбрать нужный элемент;

- для выбора одного варианта из нескольких нужно нажать соответствующий переключатель;
- для выбора одного или несколько доступных вариантов рядом с ними необходимо установить галочку.

**Область отображения задач** — это часть панели задач, расположенная правее кнопки **Пуск**. В ней отображаются в виде кнопок все запущенные программы или открытые документы. При закрытии окна программы или документа эта кнопка исчезает.

**Область уведомлений** занимает правую часть панели задач. В этой области отображаются системные часы и пиктограммы программ, которые автоматически запускаются при открытии операционной системы. Кроме того, после отправки документа на печать в области уведомлений отображается значок принтера, пока этот документ печатается. Также здесь отображаются сообщения, когда на веб-узле корпорации Майкрософт появляются обновления Windows, доступные для загрузки.

**Окна** — это графический объект, внутри которого на рабочем столе отображается содержимое файлов и программ. Внутри окна располагается главное меню программы. Имя каждого окна отображается в его верхней части, заголовке и соответствует названию программы или документа, открытого в нем. Перемещение окна может осуществляться путем перетаскивания с помощью нажатия и удержания в этом состоянии левой клавиши манипулятора. Уменьшение окна до размера значка осуществляется нажатием кнопки **Свернуть** в правой части заголовка окна. При этом окно будет свернуто в кнопку на панели задач. Развертывание окна осуществляется нажатием кнопки **Развернуть**, расположенной справа от кнопки свертывания. При этом окно разворачивается на весь экран. При следующем нажатии этой кнопки будет восстановлен исходный размер окна.

## Контрольные вопросы

1. Классификация программного обеспечения по назначению.
2. Назначение и состав системного программного обеспечения.
3. Назначение и признаки классификации операционных систем.
4. Файловая система. Понятие файла и папки. Свойства файлов.
5. Структура файловой системы ОС MS Windows.
6. Особенности технологии работы в среде ОС MS Windows.
7. Особенности интерфейса ОС MS Windows.

## Глава 5

# ОСНОВЫ ТЕХНОЛОГИИ ОБРАБОТКИ ТЕКСТОВ

---

---

Обработка текстов — один из наиболее распространенных видов работ, выполняемых на персональном компьютере. Это обусловлено тем, что деятельность любого учреждения неизбежно требует создания многих видов документов, без которых невозможно решать задачи планирования, финансирования, кредитования, бухгалтерского учета и отчетности, оперативного управления, кадрового обеспечения деятельности учреждения.

Именно документы, т. е. зафиксированная на материальном носителе информация с реквизитами, позволяющими ее идентифицировать, служат основным доказательством того или иного факта при решении хозяйственных споров или при рассмотрении трудовых конфликтов с работниками.

Для создания документов используются специальные программы — текстовые редакторы. Существует большое количество текстовых редакторов, отличающихся набором предоставляемых пользователю функций, легкостью освоения, операционными системами в которых они могут работать, быстродействием, стоимостью, распространенностью и другими параметрами. По своим возможностям эти программы можно разделить на три класса:

- **текстовые редакторы** — программы для создания, редактирования, форматирования, сохранения и организации печати текстовых документов. Пример — приложения Wordpad, Notepad, входящие в состав операционной системы MS Windows;
- **текстовые процессоры** — более совершенные текстовые редакторы, имеющие, помимо перечисленных возможностей, средства форматирования текста и документа. Пример — MS Word, входящая в пакет прикладных программ MS Office;

- **настольные издательские системы (программы компьютерной верстки)** — мощные пакеты программ, предназначенные для верстки сложных изданий. Пример — Adobe Page Maker, QwarkXpress, Ventura Publisher.


Несмотря на различия, в большинстве текстовых редакторов используются одни и те же принципы работы. Данное обстоятельство позволяет использовать в качестве примера для уяснения технологии обработки текстовой информации текстовый процессор MS Word.

## 5.1. Документ и его состав

Прикладная программа MS Word представляет документ как совокупность символов, логически связанных между собой и образующих завершённую лексическую конструкцию. Во внешней памяти компьютера документ хранится как файл с расширением \*.doc. Документ состоит из объектов, каждый из которых обладает своими свойствами. Основное содержание документа составляет, как правило, текст — набор символов, вводимых с помощью клавиатуры. Эти символы образуют текстовые объекты: слова, предложения и абзацы. Текстовые объекты располагаются на страницах так, как они будут отображаться на бумажном носителе при выводе документа на печать.

### 5.1.1. Текстовые объекты документа

Рассмотрим подробнее особенности текстовых объектов, составляющих документ.

**Символ** — наименьший элемент текста, вводимый в документ при однократном нажатии клавиши. В документе символы отображаются как буквы, цифры, знаки препинания. Нажатие некоторых клавиш вводит в текст непечатаемые символы (табл. 5.1). Эти символы не отображаются в документе при выводе его на печать. Для того чтобы увидеть эти символы на мониторе компьютера при подготовке документа к печати, следует использовать кнопку  на панели инструментов.

Следует отметить, что программы обработки текста позволяют включать в текст и специальные символы. Они создаются

Таблица 5.1. Перечень непечатаемых символов

Символ	Значение символа	Клавиша (сочетание клавиш)	Отображение символа в режиме показа непечатаемых символов
Пробел	Пустой промежуток между символами или словами	Пробел	•
Конец строки	Команда программе начать печать текста с новой строки	Shift + Enter	↵
Табулятор	Команда программе начать ввод символа с новой позиции табуляции	Tab	→
Конец абзаца	Команда программе перейти к следующему абзацу текста	Enter	¶

специальными программами, входящими в состав текстовых редакторов, и хранятся в специальных таблицах. Фрагмент такой таблицы представлен на рис. 5.1.

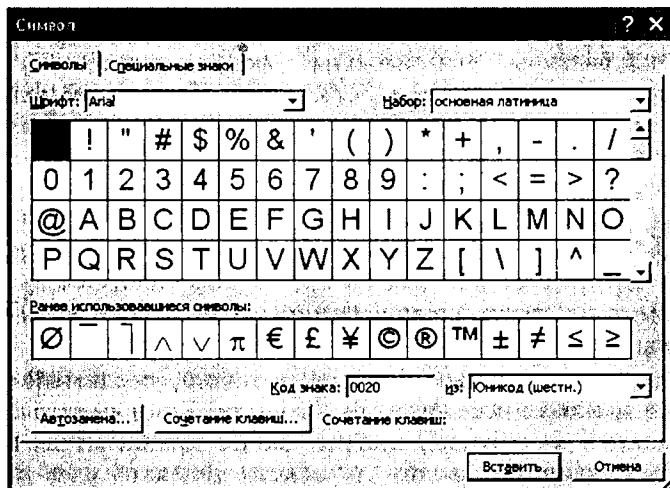


Рис. 5.1. Пример таблицы символов

**Слово** — набор символов, ограниченный с двух сторон пробелами или знаками препинания (точками, запятыми и т. д.).

**Строка** — набор слов или символов, расположенных в одну линию (без переносов).

**Предложение** — набор символов и слов, ограниченный с двух сторон знаками препинания (точками, восклицательными или вопросительными знаками, многоточием).

**Абзац** — это произвольная последовательность символов, замкнутая символом «Возврат каретки» (Клавиша **Enter**). В документе абзац представляет собой часть текста, содержащую законченную мысль. Необходимость принудительного обозначения конца абзаца обусловлена тем, что программы обработки текста не могут осуществить анализ смысла текста.

**Страница** — часть текста, ограниченная линиями разделения страниц. Страница представляет собой сложный объект, обязательными элементами которого являются поля (рис. 5.2).

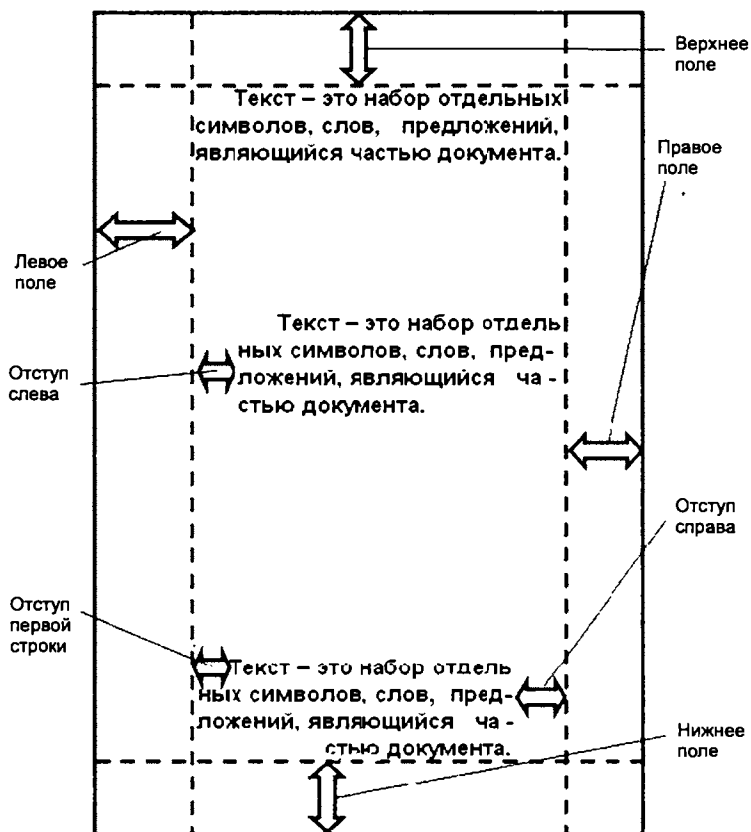


Рис. 5.2. Схема страницы документа



**Поля** — области страницы, где не может размещаться текст. Исключение составляют верхнее и нижнее поля, в которых может размещаться служебная информация. Эти элементы страницы называются колонтитулами. В качестве колонтитула может быть использован текст и/или рисунок (номер страницы, дата печати документа, логотип организации, название документа, имя файла, фамилия автора и т. п.). Область страницы, на которой может располагаться текст, называется полем текста или логической страницей. Текст может располагаться на некотором расстоянии от боковых (левого/правого) полей. Это расстояние называется отступом. При нулевом отступе границы текста совпадают с границами полей. Иногда при оформлении текста первую строку абзаца размещают с отступом от левого поля. Такую строку называют красной.

### 5.1.2. Свойства текстовых объектов документа

Свойства объекта, которые могут быть изменены пользователем, называются его атрибутами. Рассмотрим атрибуты текстовых объектов документа.

#### 5.1.2.1. Атрибуты шрифта

Любой текст может быть представлен как совокупность символов. Для обозначения размера и рисунка символов, используемых при создании документа с помощью программ обработки текста, применяется термин «шрифт». Любой текстовый редактор дает возможность использовать различные виды шрифтов, каждый из которых может иметь достаточно большое количество модификаций, различающихся своими атрибутами, основными из которых являются:

- **гарнитура шрифта** — совокупность наборного материала, имеющего одинаковый характер рисунка символов. По внешнему виду различают шрифты с засечками и рубленые (без засечек). Шрифты с засечками визуально как бы объединяют слово в единое целое, и это увеличивает скорость чтения на 10—15 %. Рубленые шрифты, как правило, используются в заголовках и подписях к рисункам.

По способу формирования изображения символов шрифты делятся на растровые и векторные. Изображение растрового символа кодируется по точкам в битовой карте (матрице), а затем без изменений отображается на экране или бумаге принтера. Матрица растрового шрифта, как правило, представляет собой квадрат (например,  $8 \times 8$  или  $16 \times 16$  пикселей). Существенный недостаток растровых шрифтов состоит в ухудшении качества при увеличении символа. В этом случае изображение символа приобретает ступенчатые очертания.

При создании векторного шрифта рисунок символа не кодируется по точкам, а описывается совокупностью геометрических фигур. Поэтому при воспроизведении символа его контур рассчитывается по определенным формулам. Это позволяет легко изменять масштаб векторных шрифтов без потери качества изображения.

В среде Windows для работы с документами, как правило, используются векторные шрифты специального формата TrueType. При этом один и тот же шрифт применяется и для вывода сообщений на экран, и для распечатки текста на принтере. Технология TrueType позволяет реализовать принцип WYSIWYG (What You See Is What You Get). Это означает, что документ, распечатанный на принтере, будет выглядеть так же, как и на экране монитора.

Каждый шрифт TrueType имеет название (имя), например Arial, Times New Roman, Symbol. Шрифты TrueType делятся на моноширинные и пропорциональные. В моноширинных шрифтах все символы имеют одинаковую ширину, например символ «.» (точка) занимает столько же места, сколько буква «ж». В пропорциональных шрифтах ширина каждого символа разная. Точка занимает совсем мало места, промежуточную ширину имеет буква «н», а буквы «ж», «ш» — шире всех других.

В качестве примеров различных шрифтов можно привести:

- шрифт Times New Roman — пропорциональный с засечками;
- шрифт Arial — пропорциональный рубленый;
- шрифт Courier New — моноширинный с засечками;

• **начертание шрифта.** Каждый шрифт имеет четыре варианта начертания: обычный, **полужирный**, **наклонный** (часто называемый *курсивом*) и **подчеркнутый**. Могут также использоваться комбинации начертания, например одновременно **полужирный, наклонный и подчеркнутый**;

- **размер символов.** Символы имеют размер, называемый кеглем. Величина кегля измеряется в пунктах (пт, pt).  $1 \text{ пт} = 1/72''$ . Данная запись означает, что 1 пт равен  $1/72$  части дюйма (2,54 см). Следовательно, символы размером 12 пт имеют высоту около 4,2 мм.

Кроме этих атрибутов, можно изменять и другие: цвет символов, их подчеркивание, видоизменения символов, интервал между ними. Представление о других атрибутах шрифта дает рис. 5.3, на котором изображено диалоговое окно форматирования шрифта в текстовом процессоре MS Word (Команда **Формат/Шрифт**).

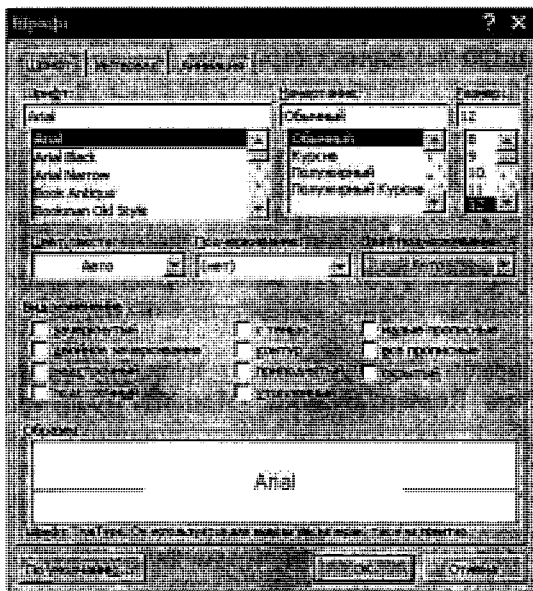


Рис. 5.3. Атрибуты шрифта в MS Word

#### 5.1.2.2. Атрибуты абзаца

Для текстового процессора MS Word абзац — это часть текста, введенная между двумя нажатиями клавиши **Enter**. Основными атрибутами абзацев являются выравнивание, отступы и интервалы.

Различается четыре вида выравнивания (рис. 5.4): по левому краю (1), по центру (2), по правому краю (3) и по ширине (4).

Представление об атрибутах абзаца дает рис. 5.5, на котором изображено диалоговое окно форматирования шрифта в текстовом процессоре MS Word (команда **Формат/Абзац**).

- 1 — Обработка текстов – один из наиболее распространенных видов работ, выполняемых на персональном компьютере. Это обусловлено тем, что деятельность любого учреждения неизбежно требует создания многих видов документов, без которых невозможно решать задачи планирования, финансирования, кредитования, бухгалтерского учета и отчетности, оперативного управления, кадрового обеспечения деятельности учреждения.
- 2 — Именно документы, т.е. зафиксированная на материальном носителе информация с реквизитами, позволяющими ее идентифицировать, служат основным доказательством того или иного факта при решении хозяйственных споров или при рассмотрении трудовых конфликтов с работниками.
- 3 — Для создания документов используются специальные программы – текстовые редакторы. Существует большое количество текстовых редакторов, отличающихся набором предоставляемых пользователю функций, легкостью освоения, операционными системами в которых они могут работать, быстродействием, стоимостью, распространенностью и другими параметрами. По своим возможностям эти программы можно разделить на три класса:
- 4 — Именно документы, т.е. зафиксированная на материальном носителе информация с реквизитами, позволяющими ее идентифицировать, служат основным доказательством того или иного факта при решении хозяйственных споров или при рассмотрении трудовых конфликтов с работниками.

Рис. 5.4. Пример различного выравнивания абзацев

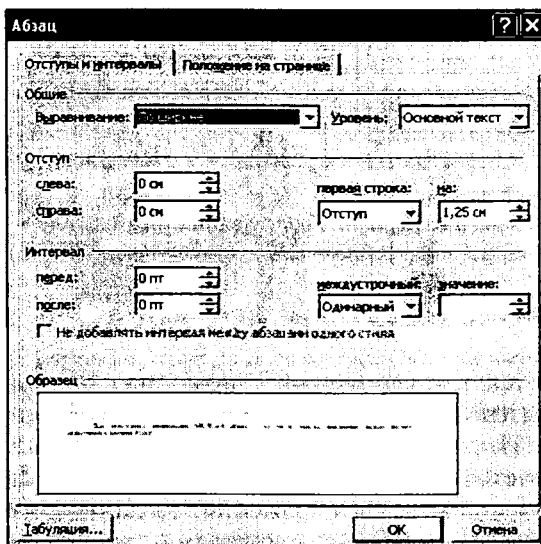


Рис. 5.5. Атрибуты абзаца в MS Word

Атрибут «отступ» характеризует расстояние от края текста до соответствующей внутренней границы бокового поля.

Что касается атрибута «интервал», то различают:

- интервалы междустрочные — это расстояние между строками внутри абзаца;
- интервалы между абзацами — расстояние от последней строки предыдущего абзаца до первой строки последующего.

### 5.1.2.3. Атрибуты страницы

Основными атрибутами страницы являются: ее размер, ширина полей и ориентация страницы, которая может быть книжной или альбомной (рис. 5.6).

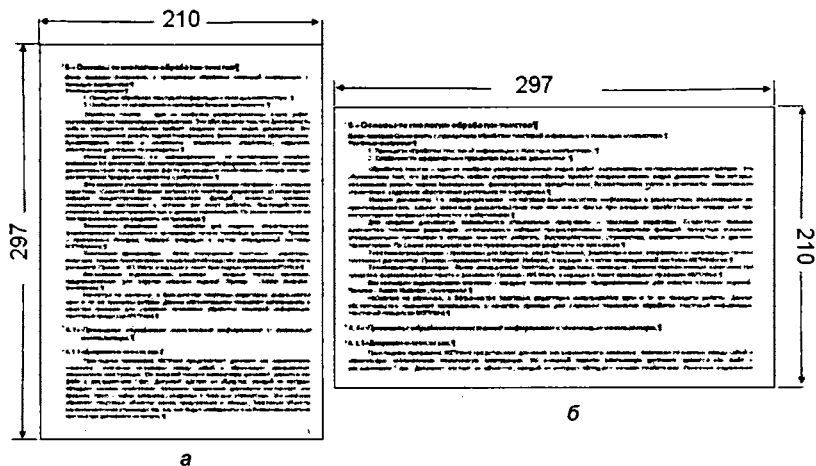


Рис. 5.6. Варианты ориентации страницы А4:

а — книжная; б — альбомная

### 5.1.3. Графические объекты документа

Зачастую текст, создаваемый с помощью текстовых процессоров, нуждается в пояснении с помощью наглядных примеров или образов. Изображение, которое включается в текст с этой целью, называется иллюстрацией. В качестве иллюстраций в документ могут включаться изображения, созданные как специальными приложениями, так и средствами, входящими в состав текстового процессора. Текстовый процессор MS Word позволя-

ет в полной мере реализовать эти возможности. Так, с помощью команды **Вставка/Рисунок** в текст документа можно включить графические объекты, созданные с помощью разных средств (рис. 5.7).

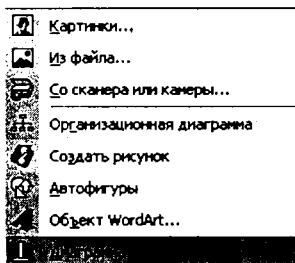


Рис. 5.7. Меню команды **Вставка/Рисунок**

Так, команда **Картинки** позволяет вставить в текст рисунки, входящие в комплект поставки пакета Microsoft Office. Они хранятся в папке **C:\MicrosoftOffice\ProgramFiles\media\cagcat**. При этом каждой картинке, хранящейся здесь, соответствует определенный набор ключевых слов. Поэтому выбор подходящей картинки существенно облегчается, так как пользователь может указать что именно ему требуется. Так, ключевому слову «компьютер» в текстовом процессоре MS Word отвечают картинки, часть которых приведена на рис. 5.8.

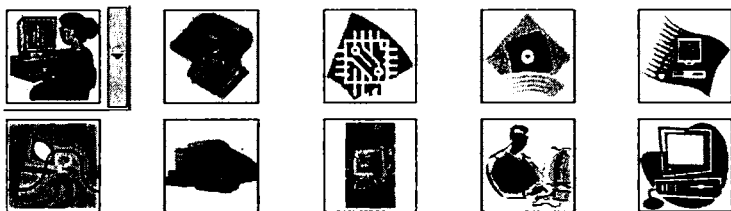


Рис. 5.8. Картинки, найденные по ключевому слову «компьютер»

С помощью команды **Из файла** можно вставить графические файлы, созданные в среде различных графических редакторов, например Adobe PhotoShop, CorelDraw, 3D Studio Max, GIMP и др. Кроме того, изображение в документ можно импортировать непосредственно со сканера или с цифровой камеры с помощью соответствующей команды.

Для включения в текст документа отдельных графических объектов могут быть использованы графические приложения, входящие в состав текстового процессора.

**Организационная диаграмма.** Данная команда позволяет включить в текст документа шесть типов диаграмм, выражающих отношения между объектами (рис. 5.9).

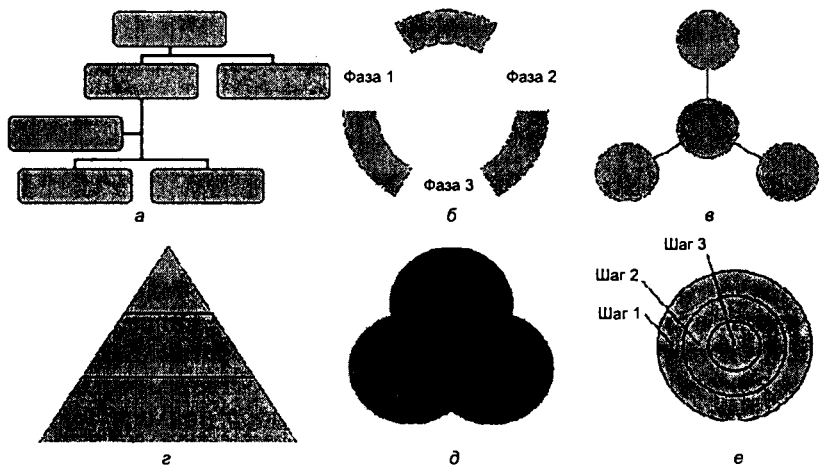


Рис. 5.9. Варианты организационных диаграмм в MS Word:

*а* — структура отношений подчинения; *б* — структура непрерывного процесса; *в* — структура с центральным элементом; *г* — иерархическая структура с базовым элементом; *д* — перекрывающиеся множества; *е* — последовательность достижения цели

**Объект WordArt.** С помощью этой команды в текст документа можно ввести художественно оформленную надпись. Программа позволяет выбрать один из тридцати вариантов ее оформления (рис. 5.10). Кроме того, пользователь может не только выбрать тип и размер шрифта, но и изменить некоторые свойства такой надписи.

**Диаграмма.** Диаграмма является способом графического представления числовых данных, хранящихся в таблицах. Это позволяет визуально представить соотношение различных значений и динамику их изменения. Программа включает большое количество различных типов диаграмм (рис. 5.11). Каждый тип диаграмм служит для определенных целей. Обычно пользователь не ограничен жесткими правилами использования какого-либо типа диаграмм.

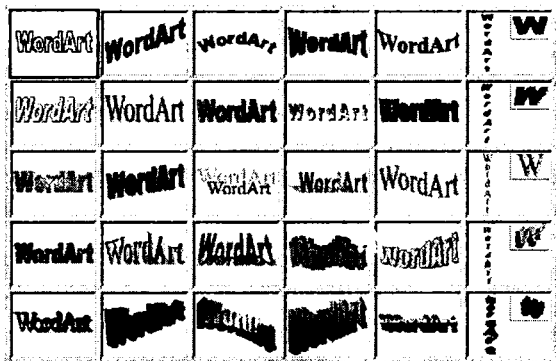


Рис. 5.10. Варианты начертания надписи WordArt



Рис. 5.11. Типы диаграмм MS Word

Кроме этих графических объектов, пользователь может создать и другие с помощью инструментов «Рисование». Доступ к ним можно получить с помощью соответствующей панели инструментов (рис. 5.12).



Рис. 5.12. Панель инструментов «Рисование»

#### 5.1.4. Свойства графических объектов документа

Ряд свойств графических объектов документа может быть изменен пользователем для придания документу наибольшей выразительности. Представление о них дает содержание диалогового окна «Формат рисунка» (рис. 5.13). Такие свойства, как цвета и линии, размер графического объекта, не требуют пояснений.



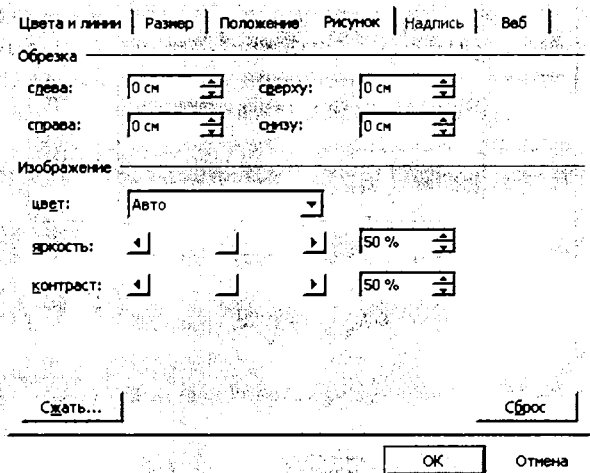


Рис. 5.13. Атрибуты графических объектов

Атрибут «Положение» определяет положение графического объекта по отношению к тексту.

Дело в том, что объекты в документе могут образовывать несколько слоев. Основным слоем является слой, где располагается текст. Графические объекты могут располагаться: а) в том же слое; б) в слое «перед текстом» или в) в слое «за текстом».

При расположении графических и текстовых объектов в одном слое рисунок может либо располагаться в тексте, либо обтекаться им. При расположении графического объекта в слое «за текстом» текст будет читаться на фоне рисунка.

## 5.2. Процедуры обработки текста

В процессе создания документа можно выделить несколько этапов. Очевидно, что сначала в документ вводится текст. Затем в нем исправляются орфографические, синтаксические и стилистические ошибки, удаляются ненужные повторы, при необходимости вводятся уточняющие слова. После этого необходимо структурировать текст, т. е. выделить в нем однородные по смыслу фрагменты: главы, параграфы, абзацы. Наконец, в текст вставляются иллюстрации — чертежи, рисунки, диаграммы, графики и т. п. На завершающем этапе следует расположить объек-

ты документа на одной или нескольких страницах определенным образом и придать им необходимые свойства.

Таким образом, можно утверждать, что в процессе обработки текста выполняются процедуры, изменяющие либо его содержание, либо его форму. Группу операций, изменяющих содержание документа, принято называть редактированием. К их числу относятся добавление, удаление, перемещение или исправление текста или графики. Другую группу операций составляют операции форматирования. При этом под форматированием понимаются преобразования, определяющие, в каком виде текст появляется на странице.

При обработке текста следует помнить, что:

- все операции выполняются над объектами;
- свойства объекта могут быть изменены только тогда, когда этот объект выделен;
- для выделения объектов используется курсор;
- при выполнении операций **Правка/копировать** или **Правка/вырезать** выделенный объект помещается в буфер обмена, откуда может быть вставлен в любое место этого же или другого документа с помощью команды **Правка/вставить**.

### 5.2.1. Редактирование документа

Компьютер, как и любое автоматическое устройство, способен решать только формализованные задачи. К числу таких задач, решаемых при редактировании, можно отнести:

- удаление, вставка, замена или перемещение любого фрагмента текста;
- копирование текстового фрагмента;
- поиск заданной последовательности символов и замена, если нужно, другой последовательностью;
- вставка графических объектов;
- орфографический контроль.

Современные текстовые процессоры, к числу которых относится MS Word, позволяют автоматизировать некоторые функции редактирования. Рассмотрим подробнее возможности MS Word по автоматизации процесса редактирования.

**Автозамена** — это функция автоматического обнаружения и исправления опечаток, неправильных слов, грамматических ошибок и неправильного регистра букв. Например, при вводе слова

**эффект** с последующим пробелом программа без вмешательства пользователя исправит неправильно введенный текст на «**эф-фekt**». Кроме того, автозамена позволяет быстро вставлять фрагменты текста, рисунки и символы. Например, для вставки символа © следует ввести (с), а с помощью сочетания тк можно вставить слова «так как». Эта функция реализуется с помощью списка автозамены. Фрагмент этого списка представлен на рис. 5.14.

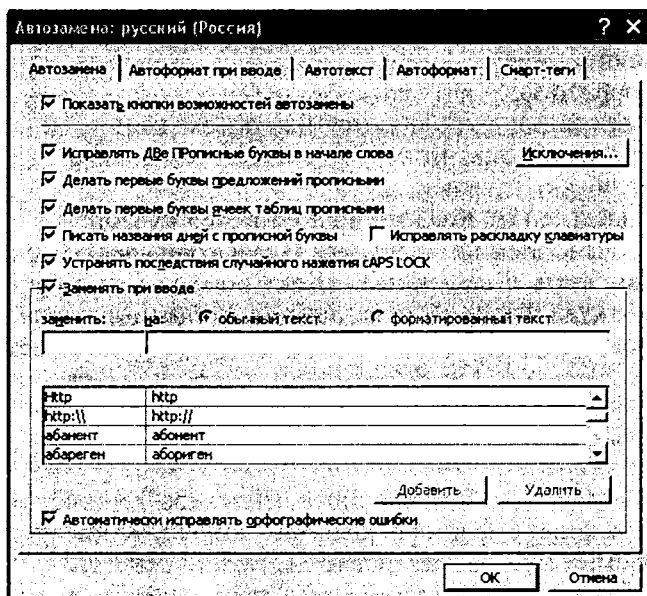


Рис. 5.14. Диалоговое окно «Автозамена»

Для автозамены используется таблица встроенных объектов, используемых при автозамене. Элементы этой таблицы позволяют обнаруживать и исправлять опечатки, неправильные слова, грамматические ошибки и общеупотребительные символы. В список автозамены нетрудно добавить новые элементы или удалить ненужные. Параметры автозамены устанавливаются с помощью команды **Сервис/параметры автозамены**. Использование данной функции позволяет:

- исправлять наиболее распространенные опечатки;
- вставлять специальные символы при вводе заранее заданной последовательности стандартных;

- быстро вставлять заранее определенные блоки текста или графики.

Последняя возможность реализуется в режиме **АВТОТЕКСТ**, который предназначен для быстрого и удобного ввода в текст отдельных слов или фрагментов текста. Автотекст служит одним из инструментов хранения и быстрой вставки текста, рисунков, полей, таблиц, закладок и других часто используемых элементов. В составе Microsoft Word поставляется обширный список встроенных элементов автотекста, классифицированных по различным категориям. Например, при создании письма можно воспользоваться такими элементами списка автотекста, как приветствия и прощания. Пользователь имеет также возможность создавать собственные элементы списка автотекста. Оформление часто используемых сложных или больших по объему элементов в виде элементов списка автотекста позволяет избежать необходимости их повторного ввода. Кроме того, автотекст удобен для хранения фрагментов текста с определенным форматированием или стилем. Например, если требуется ежемесячно отправлять клиентам отчет, содержащий один и тот же длинный текст заявления, удобно создать для этого заявления элемент списка автотекста.

**Поиск и замена.** Для быстрого поиска необходимых фрагментов текста служит функция **найти**. Программа, реализующая эту функцию, позволяет осуществлять не только поиск текста, но и его замену и переход к необходимому месту документа, которое может быть указано путем задания номеров страниц, разделов, строк, сносок, таблиц, рисунков, формул, а также задания имен закладок, примечаний, полей. Поиск и замена осуществляются по команде **Правка/найти (Правка/заменить)**. Параметры поиска (замены) задаются с помощью соответствующего диалогового окна (рис. 5.15).

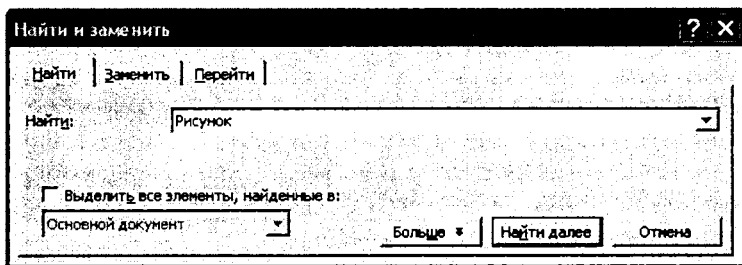


Рис. 5.15. Диалоговое окно «Найти и заменить»

Автоматическая проверка правописания включается с помощью команды **Сервис/Правописание**. При этом можно выбрать по отдельности орфографию и синтаксис путем установки соответствующих флажков в дополнительном диалоговом окне **Параметры** (рис. 5.16).

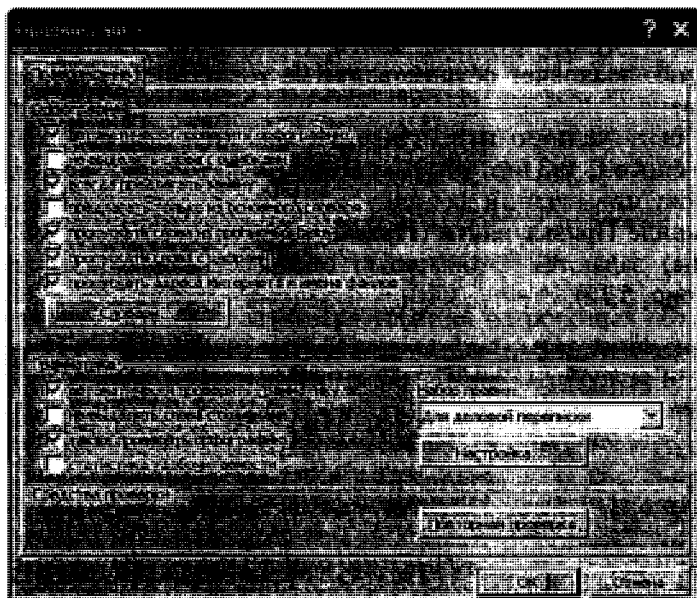
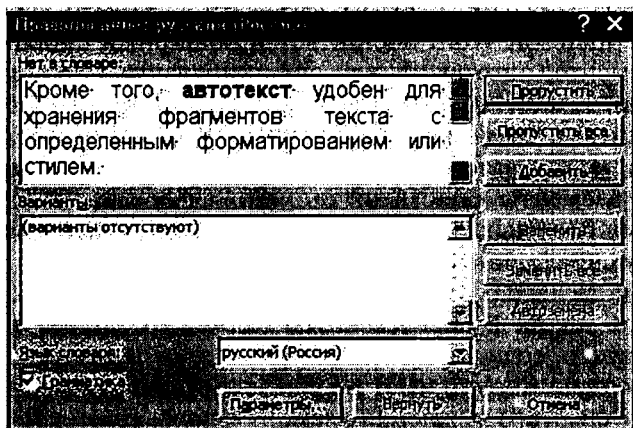


Рис. 5.16. Диалоговое окно «Правописание»

При включении автоматической проверки орфографии и грамматики флажки **не выделять слова с ошибками** в группах «**Орфография**» и «**Грамматика**» автоматически снимаются. Это позволяет выделять орфографические и грамматические ошибки волнистым подчеркиванием. Если отображение волнистых подчеркиваний нежелательно, можно установить эти флажки. При проверке правописания слова с возможными орфографическими ошибками подчеркиваются красными волнистыми линиями, а слова с возможными грамматическими ошибками — зелеными волнистыми линиями.

Для размещения в документе переменных данных и создания составных документов: конвертов, наклеек и т. п. используются **Поля**. По своей сущности поле представляет собой программу, обозначенную уникальным именем. Эта программа может быть включена пользователем в текст для автоматизации той или иной операции (рис. 5.17).

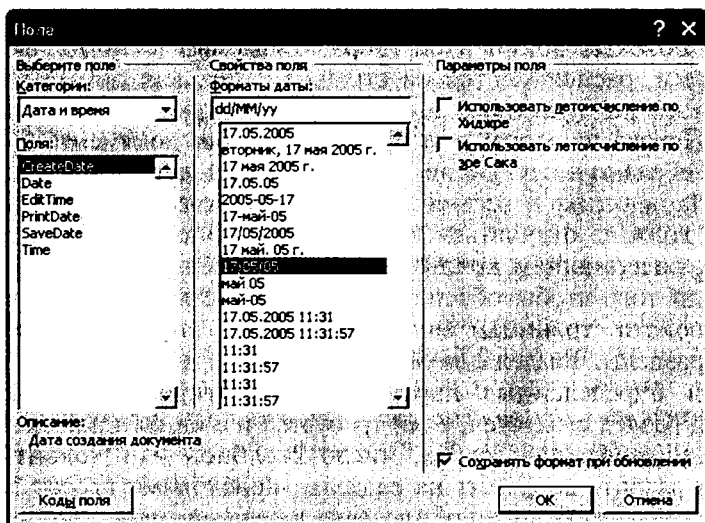


Рис. 5.17. Диалоговое окно «Поле»

Наиболее часто в документах встречаются поля:

**PAGE**, которое вставляется при добавлении номера страницы;

**DATE**, которое вставляется при выборе команды **Вставка/Дата и время** с последующей установкой флажка **Обновлять автоматически**.

Поля могут вставляться автоматически при создании предметного указателя и оглавления с помощью команды **Вставка/Оглавление и указатели**. Кроме того, использование полей позволяет организовать автоматическое обновление сведений о документе (фамилию автора, имя файла и т. п.), выполнить вычисления, установить связи с другими документами и объектами, создать перекрестные ссылки и многое другое. Коды полей в тексте документа отображаются внутри фигурных скобок ( { } ) в режиме показа непечатаемых символов.

### 5.2.2. Процедуры форматирования

Сущность форматирования заключается в изменении свойств документа в целом и/или объектов в его составе для придания им желаемой формы. Программы обработки текста разрабатывались в предположении, что, в конечном счете, текстовый документ будет перенесен на бумажный носитель. Поэтому объекты документов, обрабатываемых с помощью этих программ, располагаются на страницах так, как они будут отображаться на бумажном носителе при выводе документа на печать. Исходя из этого, при форматировании документа в целом целесообразно в первую очередь определить параметры его страниц. На практике известны случаи, когда страницы одного документа должны отличаться друг от друга ориентацией, шириной полей, содержанием колонтитулов, количеством колонок в тексте. Для того чтобы обеспечить возможность придания различных свойств страницам одного документа, внутри него выделяются разделы. **Раздел документа** — это его часть, которой присвоены определенные параметры форматирования страницы. Новый раздел создается, если требуется изменить такие параметры, как нумерацию строк, число столбцов или колонтитулы. Чтобы разбить документ на разделы, необходимо ввести символ конца раздела. Если конец раздела не указан, документ обрабатывается как один раздел.

При форматировании текстовых объектов изменяются свойства всех его фрагментов. Так, в случае выделения текста всего документа или одного из его разделов команда **Формат/шрифт** позволит изменить свойства всех символов, входящих в этот фрагмент, а команда **Формат/абзац** — всех абзацев данного фрагмента.

Текстовый процессор MS Word предоставляет возможность автоматизировать некоторые операции форматирования. Одна из таких возможностей реализуется командой **Сервис/Параметры автозамены/Автоформат при вводе** (рис. 5.18).

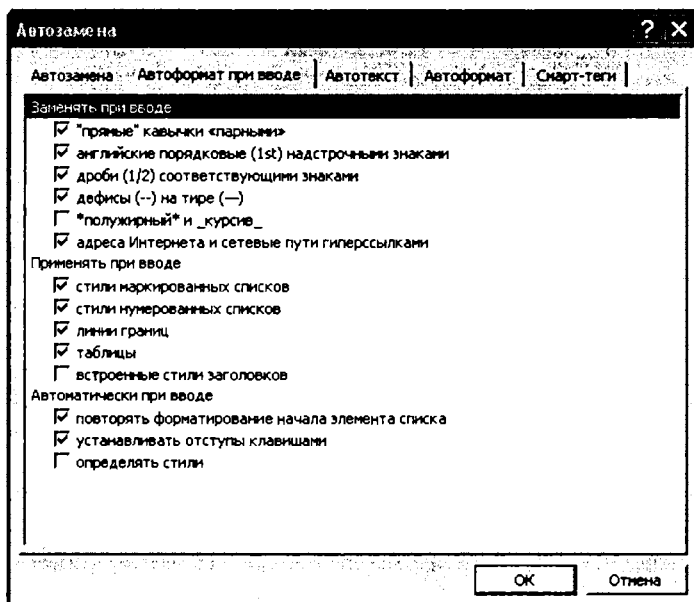


Рис. 5.18. Возможности автоформатирования

**Автоматическое форматирование документа** осуществляется либо непосредственно при вводе, либо по его завершении. При этом проводится анализ использования каждого абзаца в документе (например, является ли абзац заголовком или элементом нумерованного списка), а затем к каждому элементу применяется соответствующий стиль.

**Применение стиля** представляет собой еще одну возможность автоматизации оформления текста. Стиль — это поименованная совокупность параметров шрифта и абзаца, а также некоторых элементов оформления текста (линий и рамок). Использование стилей позволяет обеспечить однообразное оформление выделенных фрагментов текста. Пользователь может использовать встроенные стили, изменять их или создавать свои. Для этого используется команда **Формат/Стили и форматирование**. При



этом можно задать свойства текстовых объектов «шрифт» и «абзац» (рис. 5.19).

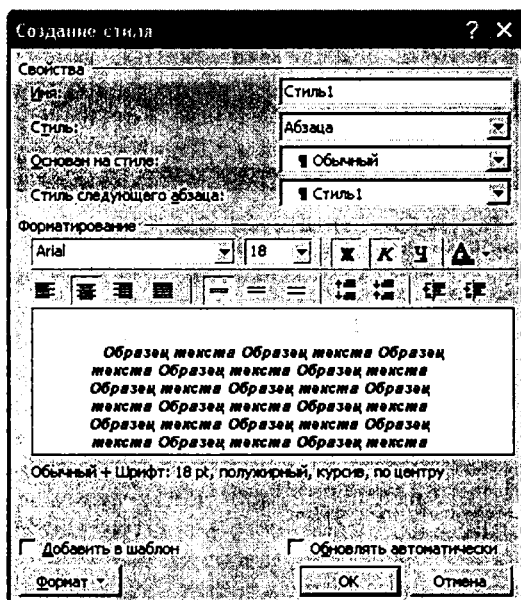


Рис. 5.19. Параметры форматирования, определяемые стилем

Кроме того, использование некоторых стилей позволяет автоматизировать некоторые операции по созданию документов. Например, применение стиля «Заголовок» позволяет автоматически создать оглавление документа, а стиля «Название объекта» — перечень иллюстраций (рисунков, таблиц, формул).

Текстовые процессоры обеспечивают возможность создания многоколонного текста. В этом случае текст в колонках непрерывно перетекает из нижней части одной колонки в верхнюю часть следующей колонки. Можно задать число колонок, промежутки между ними и вставить вертикальные разделители между колонками. Можно форматировать текст таким образом, чтобы колонки различались шириной или были одинаковыми (рис. 5.20). Многоколонный текст в документе выделяется как его раздел.

В некоторых случаях фрагмент документа может быть оформлен как список. Список — это перечисление однородных элементов текста, содержащих перечисление. Примером списков

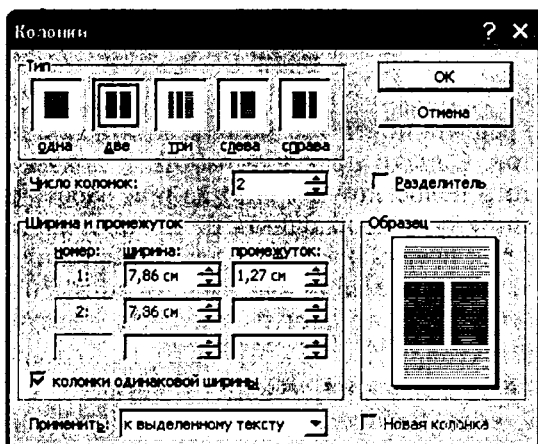


Рис. 5.20. Свойства многоколонного текста

могут служить перечни товаров, пунктов распорядка дня, действий и т. п. Списки в тексте оформляются так, что выделяется каждый элемент этого перечня. В зависимости от способа выделения различаются списки маркированные, нумерованные и многоуровневые. Для создания списков каждый элемент перечня должен оформляться как самостоятельный абзац. Этот способ оформления реализуется командой **Формат/список** (рис. 5.21).

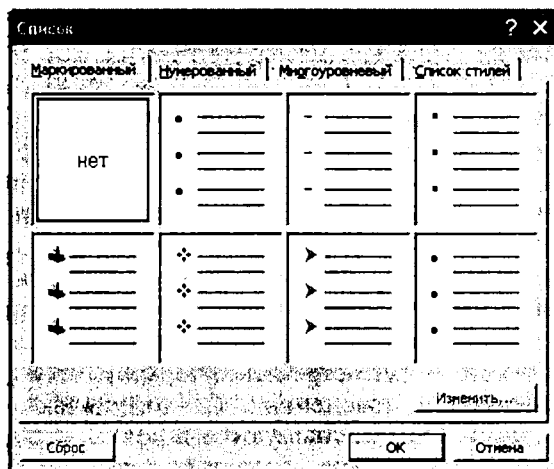



Рис. 5.21. Возможности оформления списков

### 5.3. Дополнительные объекты документа

Описанные возможности текстовых процессоров позволяют пользователю решить почти все задачи, возникающие при создании документов. Однако практика работы с документами показывает, что этого недостаточно. Иногда при работе с документами может возникнуть необходимость использования дополнительных объектов, предназначенных для повышения наглядности текста, облегчения понимания содержания документа, обеспечения возможности быстрого доступа к нужному месту документа.

К числу объектов, которые могут быть использованы для повышения наглядности текста, можно отнести:

а) *надписи*, которые представляют собой контейнер для текста или графики, местоположение и размеры которого можно изменять. Надписи используются для помещения на страницу нескольких блоков текста или для придания тексту ориентации, отличной от ориентации остального текста документа. При вставке надписи вокруг нее отображается полотно, но при необходимости надпись можно перетащить за пределы полотна. Для вставки надписи используется соответствующая кнопка , расположенная на панели инструментов **Рисование**;

б) *названия* — это нумерованные надписи (например, «Рисунок 1»), которую добавляют к таблице, рисунку, формуле и любому другому элементу. Название состоит из текста, обозначающего название объекта, и номера. Текст названия пользователь может выбрать из списка, а номер автоматически определяется программой;

в) *перекрестные ссылки* — это указание на объект, который находится в другой части документа, например «См. рис. 1». Можно создать перекрестные ссылки на заголовки, сноски, закладки, названия и нумерованные абзацы. Пример перекрестной ссылки приведен на рис. 5.22. В этом примере текст на странице 5 содержит перекрестную ссылку на объект (рисунок), расположенный на следующей странице. Для вставки перекрестной ссылки используется команда **Вставка/Ссылка/Перекрестная ссылка**.

Иногда в тексте могут встречаться термины или выражения, которые нуждаются в пояснениях. С этой целью в документ включаются *сноски*. Они используются в документе для пояснений, комментариев и ссылок на другие документы. При этом для подробных комментариев лучше использовать обычные, а

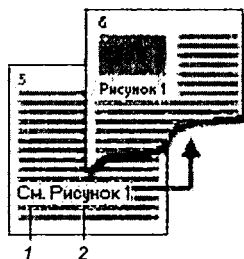


Рис. 5.22. Пример перекрестной ссылки:

1 — необязательный текст; 2 — позиция, выбранная в диалоговом окне  
**Перекрестная ссылка**

для ссылок на источники — концевые сноски. Ограничения на длину и оформление текста сносок отсутствуют. Сноска состоит из двух связанных частей: *знака сноски* и *текста сноски*. По умолчанию обычные сноски располагаются в конце каждой страницы, а концевые сноски — в конце документа. Можно изменить положение обычных сносок таким образом, чтобы они отображались непосредственно под текстом. Аналогично можно изменить положение концевых сносок, чтобы они располагались непосредственно после каждого раздела (рис. 5.23).

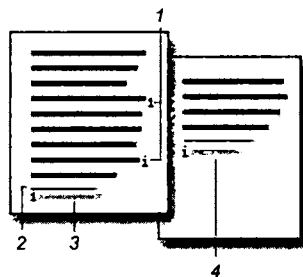


Рис. 5.23. Пример сносок:

1 — знаки обычной и концевой сноски; 2 — разделитель; 3 — текст обычной сноски; 4 — текст концевой сноски

В качестве знака сноски могут использоваться отдельные символы или их совокупность, которые указывают на наличие в сноске дополнительных сведений, содержащихся в тексте сноски. Допускается изменение разделителя сносок — линии, отделяющей текст документа от текста сноски. Сноски включаются в текст с помощью команды **Вставка/Ссылка/Сноска**.

Современные текстовые процессоры учитывают особенности взаимодействия исполнителей в процессе создания документа.

Предположим, руководителю организации необходимо подготовить доклад. В этом случае он определяет его замысел, который включает тему и сроки доклада, а также требования к нему. Затем он ставит задачу на подготовку этого доклада одному из сотрудников (референту), который готовит задания для разработчиков доклада, т. е. определяет тему, сроки исполнения и требования к отдельным частям доклада. Разработчики доклада после получения задания готовят назначенные им части доклада и в установленные сроки передают их референту. После этого референт компилирует доклад из подготовленных частей и редактирует его. Готовый доклад передается руководителю для окончательной редакции. Эти операции могут повторяться до тех пор, пока руководитель не примет решение о готовности доклада.

Очевидно, что в процессе такой работы возникает необходимость отобразить в тексте документа те изменения, которые вносят в него руководитель и референт. Кроме того, целесообразно предоставить разработчику или референту возможность включить примечания, чтобы обосновать тот или иной тезис доклада.

Для упрощения проверки документов в диалоговом режиме MS Word предоставляет пользователю возможность просто пометить и просмотреть записи исправлений и примечания.

Записанное исправление представляет собой пометку удаления, вставки и других изменений, внесенных в документ. Запись исправлений осуществляется с помощью панели инструментов **Рецензирование** (рис. 5.24).

Записанное исправление оформляется как выноска (рис. 5.25).

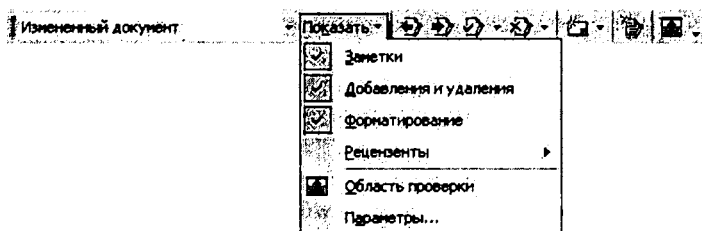


Рис. 5.24. Панель инструментов Рецензирование

референтом. Кроме того, целесообразно предоставить разработчику или референту возможность включить примечания, чтобы обосновать тот или иной тезис доклада. Для упрощения проверки документов в диалоговом режиме MS Word

Форматирование: Шрифт: полужирный

Рис. 5.25. Внешний вид записанного исправления

Примечание представляет собой заметку, добавленную автором или рецензентом в документ. В Microsoft Word примечания отображаются в выносках на полях документа и на панели рецензирования (рис. 5.26). Для вставки примечаний служит команда **Вставка/Примечание**.

Для упрощения проверки документов в диалоговом режиме MS Word предоставляет пользователю возможность просто пометить и просмотреть записи исправлений и примечания.

Рис. 5.26. Внешний вид примечания

## 5.4. Особенности работы с документами большого объема

В процессе работы с документом возникает необходимость обращения к различным его фрагментам. При небольшом объеме документа такое перемещение по тексту не вызывает затруднений. Для облегчения перемещения между различными частями многостраничных документов целесообразно прибегнуть к структурированию текста. Структура документа отражает его логическую организацию. Как правило, для этой цели используется иерархическая структура. Так, книга может быть разделена на главы. В свою очередь, главы делятся на параграфы, а последние — на абзацы. С целью идентификации таких фрагментов текста им присваиваются заголовки. В текстовых процессорах заголовком принято считать текст, оформленный как отдельный абзац, который служит для обозначения того или иного фрагмента документа.

Для того чтобы текстовый процессор отличал заголовки от других текстовых объектов документа, при их оформлении используется стиль заголовка. Текстовый процессор MS Word позволяет различать девять уровней заголовков. Пример оформления заголовков различного уровня приведен на рис. 5.27. Таким образом, с помощью заголовков создается иерархическая структура документа.

В качестве примера такой логической организации документа приведена структура данной главы (рис. 5.28). Созданные таким образом заголовки позволяют автоматически создавать оглавление документа, которое представляет собой список заголовков документа.

**6 Заголовок 1****6.1 Заголовок 2****6.1.1 Заголовок 3****6.1.1.1 Заголовок 4****6.1.1.1.1 Заголовок 5****6.1.1.1.1.1 Заголовок 6****6.1.1.1.1.1.1 Заголовок 7****6.1.1.1.1.1.1.1 Заголовок 8****6.1.1.1.1.1.1.1.1 Заголовок 9**

Рис. 5.27. Оформление заголовков в MS Word

- ❖ **6 → Основы технологии обработки текстов ¶**
  - ❖ **6.1 → Документ и его состав. ¶**
    - ❖ **6.1.1 → Текстовые объекты документа ¶**
    - ❖ **6.1.2 → Свойства текстовых объектов документа ¶**
      - ❖ **6.1.2.1 → Атрибуты шрифта. ¶**
      - ❖ **6.1.2.2 → Атрибуты абзаца. ¶**
      - ❖ **6.1.2.3 → Атрибуты страницы. ¶**
    - ❖ **6.1.3 → Графические объекты документа ¶**
    - ❖ **6.1.4 → Свойства графических объектов документа ¶**
  - ❖ **6.2 → Процедуры обработки текста. ¶**
    - ❖ **6.2.1 → Редактирование документа. ¶**
    - ❖ **6.2.2 → Процедуры форматирования. ¶**
  - ❖ **6.3 → Дополнительные объекты документа. ¶**
  - ❖ **6.4 → Особенности работы с документами большого объема. ¶**
  - ❖ **6.5 → Особенности создания документа. ¶**
  - ❖ **6.6 → Указатель ¶**
  - ❖ **6.7 → Литература: ¶**

Рис. 5.28. Структура данной главы

Современные текстовые процессоры представляют пользователю возможности быстро перемещаться по тексту многостраничных документов. В качестве ориентиров для быстрого обращения к той или иной части документа используются как текстовые, так и графические объекты. Быстрый переход из одной части большого документа в другую осуществляется с помощью

команды **Правка/Перейти**. В диалоговом окне, которое вызывается данной командой, пользователь должен указать необходимый ему объект (рис. 5.29).

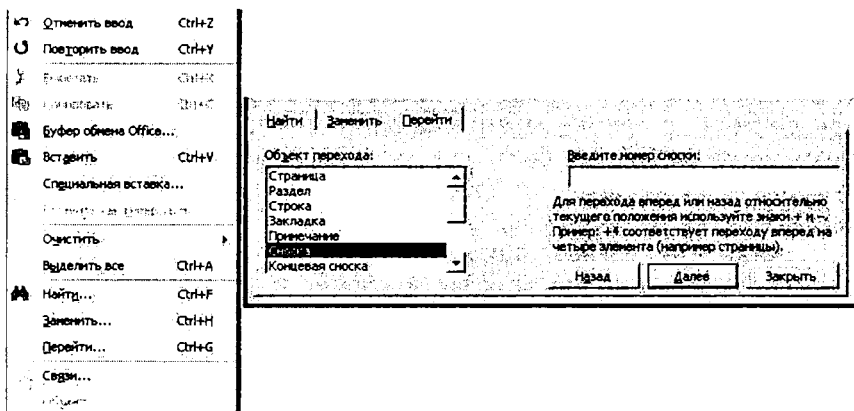


Рис. 5.29. Меню Правка и диалоговое окно Перейти

Для облегчения перемещения по тексту документа может быть использован и такой объект текстового процессора, как закладки. Закладка — это элемент документа, которому присвоено уникальное имя. Это имя можно использовать для последующих ссылок. Например, можно использовать закладку для определения текста, который необходимо проверить позже. Для создания закладок используется команда **Вставка/Закладка** (рис. 5.30).

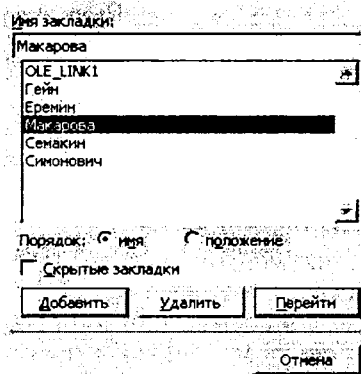


Рис. 5.30. Диалоговое окно Закладка



## 5.5. Особенности создания текстовых документов

### 5.5.1. Принципы создания документов

Принципиальная особенность текстового процессора MS Word заключается в том, что любой документ создается этой программой на основе заранее созданной заготовки. В качестве такой заготовки может быть использован уже существующий документ. Тогда процедура создания нового документа будет включать в себя следующие операции:

- 1) открытие нужного документа;
- 2) его обработка путем редактирования и форматирования;
- 3) сохранение обработанного документа под новым именем.

При необходимости создания документа, аналогом которого пользователь не располагает, в его основу должен быть положен шаблон. Это особый вид документа, предоставляющий специальные средства для создания нового. Другими словами, каждый документ Microsoft Word основан на шаблоне [1]. Шаблон определяет основную структуру документа и содержит настройки документа, такие, как элементы автотекста, шрифты, назначенные сочетания клавиш, макросы, меню, параметры страницы, форматирование и стили. По своей сущности шаблон представляет собой файл с расширением **.dot**.

Текстовый процессор имеет в своем составе общие шаблоны и шаблоны документов. Различие между этими двумя группами шаблонов заключается в том, что общие шаблоны содержат настройки, доступные для всех документов. В то же время шаблоны документов, например шаблоны записок или факсов, имеют настройки, доступные только для документов, основанных на соответствующих шаблонах.

Общие шаблоны, поставляемые в составе текстового процессора, позволяют создать новый документ, веб-страницу и сообщения электронной почты. В этом случае новый документ создается на основе шаблона **Normal.dot**.

При работе над документом, как правило, можно использовать только те настройки, которые сохранены в шаблоне, присоединенном к этому документу, или в шаблоне **Normal.dot**. Для использования настроек, хранящихся в другом шаблоне, можно загрузить нужный шаблон в качестве общего шаблона. После загрузки шаблона элементы, сохраненные в этом шаблоне, доступ-

ны в любом документе до окончания текущего сеанса работы с Microsoft Word. Следует помнить, что загруженные настройки и шаблоны выгружаются при завершении работы Microsoft Word. Настройки и шаблоны, которые требуется загружать при каждом запуске Microsoft Word, следует скопировать в подпапку Startup папки Microsoft Office.

При сохранении шаблона Microsoft Word автоматически выбирает расположение, заданное для типа файлов шаблоны пользователя (меню **Сервис**, команда **Параметры**, вкладка **Расположение**). По умолчанию используется папка **Шаблоны** и ее подпапки. Шаблоны, сохраненные в других папках, не будут отображаться в диалоговом окне **Шаблоны**. При необходимости пользователь может сохранить в этой папке и файлы с расширением **.doc**, которые в дальнейшем также могут применяться в качестве шаблона.

Шаблоны, сохраненные в папке **Шаблоны**, отображаются в диалоговом окне с тем же названием (рис. 5.31).

Данное окно содержит вкладки, соответствующие различным типам документов. Каждой такой вкладке соответствует папка с тем же названием, которая вложена в папку **Шаблоны**. Чтобы создать в этом окне пользовательские вкладки для хранения

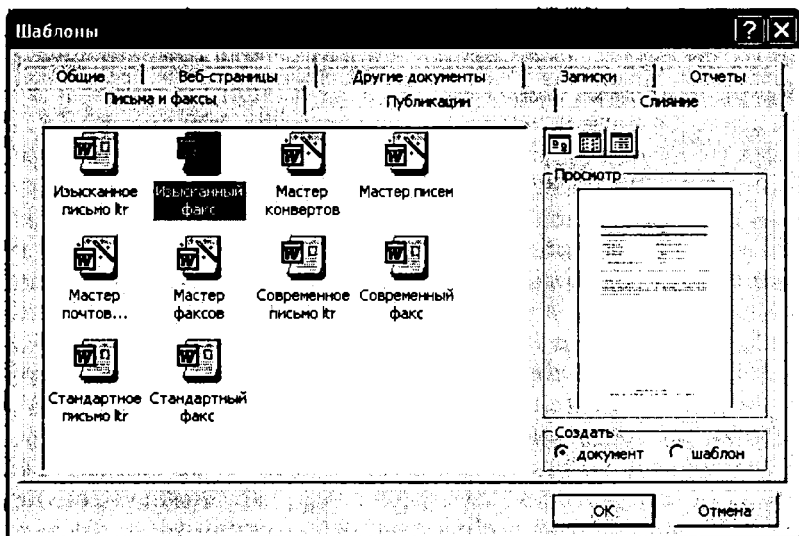


Рис. 5.31. Диалоговое окно **Шаблоны**

вновь созданных шаблонов, следует создать в папке **Шаблоны** новую вложенную папку. После этого появляется возможность сохранять новые шаблоны в ней. Любой файл с расширением **.doc**, сохраняемый в папке **Шаблоны**, также используется как шаблон.

Итак, шаблон может содержать следующие элементы:

- текст или формат, одинаковый для всех документов этого типа, например для служебной записки или отчета;
- стили;
- элементы автотекста;
- макросы;
- меню и присвоенные сочетания клавиш операции;
- панели инструментов.

Следовательно, для создания документа на основе шаблона необходимо:

1) выбрать шаблон из списка, представленного в диалоговом окне **Шаблоны**;

2) заполнить свободные поля шаблона;

3) сохранить документ в соответствующей папке, присвоив ему имя и расширение **.doc**.

Чтобы создать нужный ему шаблон, пользователь должен:

1) на основе шаблона **Normal.dot** сформировать заготовку, придав ей необходимое содержание и должную форму;

2) сохранить ее как шаблон. Для этого при сохранении данного файла следует выбрать место в папке **Шаблоны** и присвоить имя с расширением **.dot**.

### 5.5.2. Особенности создания больших документов

С помощью текстового процессора MS Word можно создавать и обрабатывать любые документы независимо от их объема. Однако при работе с документами, объем которых превышает 10—15 страниц, пользователь начинает испытывать неудобства при необходимости обращения к тексту, расположенному в разных частях такого документа. Текстовый процессор MS Word позволяет пользователю избежать это путем использования особой технологии работы с документами большого объема. Сущность этой технологии заключается в том, что большой документ создается как составной, образованный несколькими, объем которых не создает неудобств пользователю.

В этом случае большой документ представляет собой оболочку, в которую помещаются файлы, составляющие его содержание. Эта оболочка, или контейнер, называется главным документом. Отдельные файлы, объединенные главным документом, получили название вложенных документов.

В главном документе можно создать оглавление, предметный указатель, перекрестные ссылки и колонтитулы для всех вложенных документов. Для создания главного документа образуется его структура путем создания заголовков вложенным документам. В качестве вложенных документов в главный документ также можно добавлять существующие документы. Существенное значение при этом имеет расположение файлов главного и вложенных документов в одной папке. Работа с главным документом осуществляется в режиме структуры. В этом режиме заголовки документа смещены в зависимости от их уровня в структуре документа. Переход в данный режим осуществляется с помощью команды **Вид/Структура**. При переходе в этот режим появляется панель инструментов **Структура** (рис. 5.32).

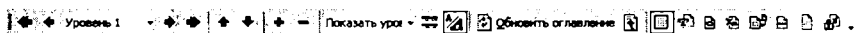



Рис. 5.32. Панель инструментов **Структура**

Для вновь создаваемого документа необходимо ввести заголовки для главного документа и всех вложенных документов. Каждый заголовок должен представлять собой один абзац, т. е. ввод каждого заголовка следует завершать нажатием клавиши **ENTER**. Затем требуется назначить соответствующий стиль каждому заголовку. Очевидно, что целесообразно использовать стиль «**Заголовок 1**» для заголовка главного документа, а стиль «**Заголовок 2**» — для каждого вложенного документа. При назначении стилей заголовкам можно использовать соответствующие кнопки на панели инструментов **Структура**. Также можно преобразовать в **Главный** уже существующий документ. Для этого необходимо создать структуру в существующем документе, используя приемы, описанные выше.

После создания Главного документа в него следует поместить вложенные документы. Для этого необходимо с помощью курсора указать место размещения вложенного документа внутри Главного. Это должна быть пустая строка между существующими вложенными документами. Затем на панели инструментов **Структура** нужно нажать кнопку **Вставить вложенный доку-**

мент . В результате выполнения данной команды появится диалоговое окно **Вставка вложенного документа**. В этом окне следует выбрать файл, который требуется вставить, и ввести его имя в поле **Имя файла**. Ввод этого файла осуществляется нажатием кнопки **Открыть**. Как объект текстового процессора, вложенный документ будет представлять собой отдельный раздел результирующего документа. Полезно вспомнить, что непечатаемый символ **Разрыв раздела** хранит сведения о таких элементах форматирования раздела, как поля, ориентация страницы, колонтитулы и последовательность номеров страниц. Вложенные документы отображаются в главном документе как гиперссылки. Для работы с содержимым вложенного документа его необходимо открыть из Главного. Следует иметь в виду, что после добавления вложенного документа в **Главный** его нельзя перемещать и удалять из папки, где он находился при включении в состав **Главного**, без предварительного удаления из **Главного** документа. Кроме того, переименовывать вложенные документы можно только из **Главного**.

Сохранение главного документа осуществляется как обычно с помощью команды **Файл/Сохранить как**. При сохранении главного документа каждому вложенному документу автоматически присваивается имя файла, которое составляется из первых символов заголовка вложенного документа. Например, вложенный документ, который начинается с заголовка «**Глава 1**», получит название «**Глава 1.doc**».

### 5.5.3. Особенности создания составных документов

В практической деятельности достаточно часто приходится сталкиваться с необходимостью создания документов, имеющих общее содержание, но различающихся реквизитами адресата, например при массовой рассылке документов на бланке, сообщений по факсу или по электронной почте, создании почтовых наклеек, конвертов, каталогов и т. п. Текстовый процессор MS Word позволяет легко решить подобную задачу с помощью так называемых составных документов. Такой документ создается путем слияния основного документа с источником данных.

В операции слияния основным называется документ, содержащий данные, которые при слиянии остаются неизменными во

всех производных документах, например обратный адрес или текст письма. Источник данных представляет собой файл, содержащий сведения, предназначенные для объединения с основным документом. В нем должны содержаться данные, свойственные каждому конкретному получателю. Так, в качестве источника данных могут быть использованы списки имен и адресов, которые должны быть включены в основной документ.

Механизм создания составных документов рассматривает адресата составного документа как объект, обладающий определенными свойствами. Хранилищем данных о свойствах адресата является источник данных. Он представляет собой таблицу, в столбцах которой размещены данные, характеризующие одно из свойств адресата. Каждый столбец источника данных должен иметь заголовок, соответствующий одному из свойств (реквизитов) адресата. Файл источника по умолчанию сохраняется как таблица базы данных (расширение **.mdb**) в папке **«Мои источники данных»**, которая вложена в системную папку **«Мои документы»**. Пользователь может самостоятельно выбрать и другое место хранения источника данных. Такая возможность предоставляется ему при создании этого файла.

Для того чтобы обеспечить создание составного документа, в основном документе должны быть указаны позиции, предназначенные для включения в него данных из источника. Для их обозначения используются поля (см. п. 6.2.1). В общем случае поле представляет собой набор кодов, обеспечивающих автоматическую вставку в документ текста, рисунков, номеров страниц и других сведений. Например, поле DATE вставляет текущую дату. В составном документе используются поля слияния, имена которых определяются заголовками столбцов источника данных. Поэтому для использования сведений из источника данных необходимо сначала подключиться к нему. Таким образом, при вставке данных полей в основной документ они отображают сведения из соответствующих столбцов в источнике данных. Схематически это представлено на рис. 5.33.

Остается отметить, что для создания составных документов используется команда **Сервис/Письма и рассылки**. Эта команда предоставляет пользователю сформировать итоговый документ одним из двух способов:

- с помощью **Мастера слияния**;
- с помощью команд, отображаемых на панели слияния.

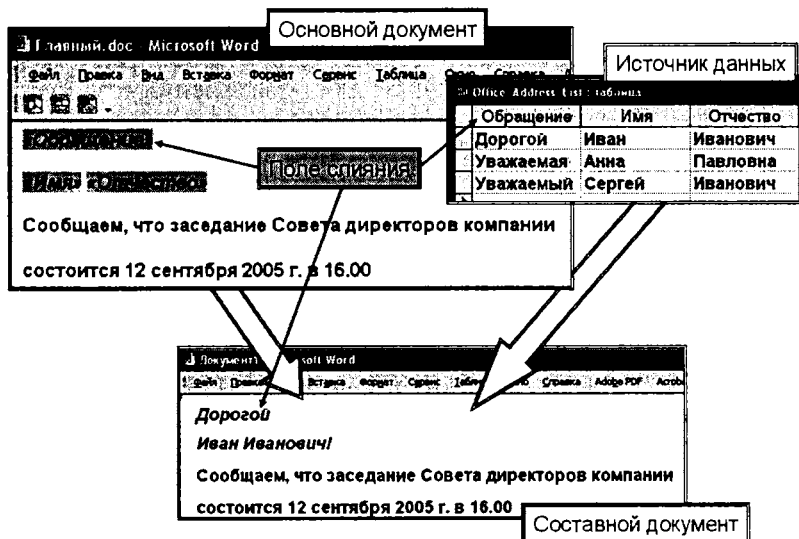


Рис. 5.33. Схема создания составного документа

Независимо от избранного способа последовательность действий пользователя включает в себя:

- 1) создание основного документа;
- 2) выбор, а при его отсутствии, создание источника данных;
- 3) включение в основной документ полей слияния;
- 4) непосредственно слияние основного документа с данными, расположенными в источнике.

## Контрольные вопросы

1. Документ как объект текстового процессора MS Word.
2. Текстовые объекты документа и их свойства: символ, абзац, страница, раздел.
3. Графические объекты документа и их свойства: рисунок, диаграмма, объекты WordArt.
4. Процедуры редактирования документа.
5. Процедуры форматирования документа.
6. Дополнительные объекты документа: надписи, названия, перекрестные ссылки, примечания.
7. Стиль и возможности его использования при создании документов.
8. Шаблоны и их использование для создания документов.
9. Создание составных документов.

## Глава 6

# ОСНОВЫ ФУНКЦИОНИРОВАНИЯ ТАБЛИЧНОГО ПРОЦЕССОРА MS EXCEL

---

---

### 6.1. Табличные структуры данных

Табличная форма представления данных нашла широкое применение при решении задач управленческой и интеллектуальной деятельности человека. Например, в естественных науках и технике используются таблицы функций (тригонометрических, логарифмов), таблицы для инженерных расчетов (таблицы механических свойств различных материалов); в финансах (таблицы кросс-курсов валют, курсов акций); в бухгалтерском учете (оборотные ведомости, баланс предприятия). Практически нет такой отрасли деятельности, где не использовались бы таблицы. В дальнейшем будем считать, что таблица — это перечень сведений, цифровых данных, записанных в известном порядке, по графам [11].

Табличные структуры данных являются простыми. Ими легко пользоваться, поскольку адрес каждого элемента задается числом (для списка), двумя числами (для двухмерной таблицы) или несколькими числами для многомерной таблицы. Они также легко упорядочиваются. Основным методом упорядочения является *сортировка*. Данные можно сортировать по любому избранному критерию, например по алфавиту, по возрастанию порядкового номера или по возрастанию какого-либо параметра.

Несмотря на многочисленные удобства, у простых структур данных есть и недостаток — их трудно обновлять. Если, например, перевести студента из одной группы в другую, изменения надо вносить сразу в два журнала посещаемости; при этом в обоих журналах будет нарушена списочная структура. Если пере-



веденного студента вписать в конец списка группы, нарушится упорядочение по алфавиту, а если его вписать в соответствии с алфавитом, то изменятся порядковые номера всех студентов, которые следуют за ним. Таким образом, при добавлении произвольного элемента в упорядоченную структуру списка может происходить изменение адресных данных у других элементов. Поэтому в системах, выполняющих автоматическую обработку данных, нужны специальные методы для решения этой проблемы. Таким образом, табличные структуры данных — это упорядоченные структуры, в которых адрес элемента определяется номером строки и номером столбца, на пересечении которых находится ячейка, содержащая искомый элемент.

В зависимости от назначения таблиц и способов обработки данных, внесенных в эти таблицы, различают два класса задач, для решения которых могут быть использованы электронные таблицы: вычислительные и информационные. Для задач обработки таблиц, содержащих относительно небольшое количество данных, характерны вычислительные процессы. Класс программ, предназначенных для решения вычислительных задач с использованием табличных данных, получил название табличных процессоров. При решении задач обработки больших наборов, содержащих до миллионов данных, более актуальны организация хранения и доступа к данным. Поэтому для решения информационных задач создаются табличные базы данных. Класс программ, предназначенных для решения информационных задач с использованием баз данных, называется системами управления базами данных.

При решении этих задач используются различные способы доступа к данным при реализации технологий обработки массивов структур. При решении вычислительных задач доступ к данным осуществляется по адресу. Для этого используются таблицы, ставящие в соответствие имя данного и адрес его местоположения в памяти. Такой способ доступа характерен для табличных процессоров. Ячейки таблицы именуется, имя складывается из буквенного обозначения столбца (А, В, С, ...) и номера строки. Например, А7, В3. При обработке больших наборов данных доступ по адресу теряет смысл.

В системах управления базами данных выбор информации осуществляется путем проверки для каждого элемента данных истинности логического выражения, описывающего запрос на доступ к данным. Такой способ доступа можно рассматривать

как разновидность контекстного поиска. Предположим, что из базы данных необходимо выбрать информацию о жителях муниципального образования, достигших пенсионного возраста. К этой категории относятся мужчины в возрасте старше 60 и женщины — старше 55 лет. Следовательно, для решения этой задачи требуется проверка на истинность для каждого из жителей следующего логического выражения:

**ИЛИ(И(пол=«муж»;год\_текущий-год\_рождения>=60);**

**И(пол=«жен»;год\_текущий-год\_рождения>=60));**

Компьютер позволяет представлять таблицы в электронной форме, а это дает возможность не только отображать, но и обрабатывать данные.

Таким образом, электронная таблица (ЭТ) — компьютерное представление обычной таблицы, в ячейках которой записаны данные различных типов: тексты, даты, формулы, числа, а табличный процессор (ТП) — комплекс программ, используемых для управления ЭТ. Особенность электронных таблиц заключается в возможности применения формул для описания связи между значениями различных ячеек. Расчет по заданным формулам выполняется автоматически. Изменение содержимого какой-либо ячейки приводит к пересчету значений всех ячеек, которые с ней связаны формульными отношениями, и, тем самым, к обновлению всей таблицы в соответствии с изменившимися данными. Пример таких вычислений приведен на рис. 6.1.

Применение электронных таблиц упрощает работу с данными и позволяет получать результаты без проведения расчетов

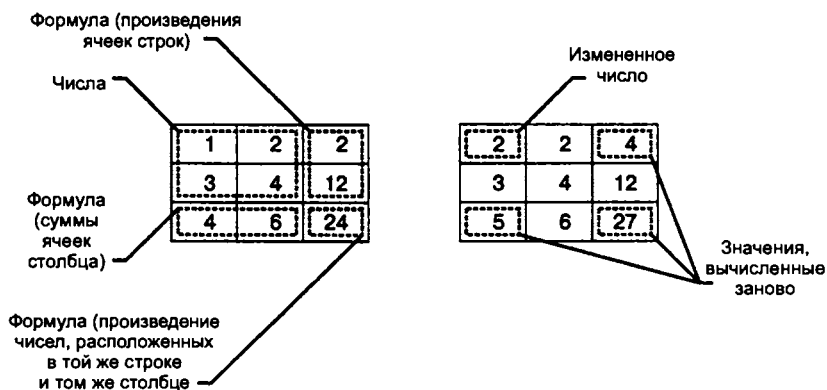


Рис. 6.1. Пример вычислений в таблице

вручную или специального программирования. Наиболее широкое применение электронные таблицы нашли в экономических и бухгалтерских расчетах, но и в научно-технических задачах электронные таблицы можно использовать эффективно, например для:

- проведения однотипных расчетов над большими наборами данных;
- автоматизации итоговых вычислений;
- решения задач путем подбора значений параметров, табулирования формул;
- обработки результатов экспериментов;
- проведения поиска оптимальных значений параметров;
- подготовки табличных документов;
- построения диаграмм и графиков по имеющимся данным.

Идея создания электронной таблицы возникла у студента Гарвардского университета (США) Дэна Бриклина (Dan Bricklin) в 1979 г. Выполняя скучные вычисления экономического характера с помощью бухгалтерской книги, он и его друг Боб Франкстон (Bob Frankston), который разбирался в программировании, разработали первую программу электронной таблицы, названную ими VisiCalc, которая очень быстро скоро стала одной из наиболее успешных программ. Первоначально она предназначалась для компьютеров типа Apple II, но потом была трансформирована для всех типов компьютеров. Многие считают, что резкое повышение продаж компьютеров типа Apple в то время и было связано с возможностью использования на них табличного процессора VisiCalc. Затем появились электронные таблицы-аналоги (например, SuperCalc), в которых основные идеи VisiCalc были многократно усовершенствованы.

Новым существенным шагом в развитии электронных таблиц было появление в 1982 г. на рынке программных средств Lotus 1-2-3. Этот продукт был первым табличным процессором, который дал возможность использовать графику и работать с базами данных. Поскольку Lotus был разработан для компьютеров типа IBM, он сделал для этой фирмы то же, что VisiCalc в свое время сделал для фирмы Apple. После разработки Lotus 1-2-3 компания Lotus в первый же год повышает свой объем продаж до 50 млн долларов и становится самой большой независимой компанией — производителем программных средств. Успех компании Lotus привел к ужесточению конкуренции, вызванной появлением на рынке новых электронных таблиц, таких, как VP

Planner компании Paperback Software и Quattro Pro компании Borland International, которые предложили пользователю практически тот же набор инструментария, но по значительно более низким ценам.

Следующий шаг — появление в 1987 г. табличного процессора Excel фирмы Microsoft. Эта программа предложила более простой графический интерфейс в комбинации с ниспадающими меню. При этом в значительной степени были расширены функциональные возможности пакета.

В настоящее время широкое применение в практической деятельности получили табличные процессоры: SuperCalc, Lotus 1-2-3, Author, Excel 4.0 (для Windows 3.1), Excel 5.0 (для Windows 3.11), Excel 7.0 (для Windows 95).

Для IBM-совместимых ПК, использующих операционную систему MS Windows (95, 97, 2000), фирма Microsoft разработала табличный процессор Excel, который включен в пакет прикладных программ для офиса Microsoft Office.

Все программы, предназначенные для решения одних и тех же задач похожими средствами, как правило, похожи друг на друга. Редакторы электронных таблиц не являются исключением. Достаточно беглого взгляда на изображения главных окон нескольких редакторов, чтобы увидеть сходство их оформления и выделить общие функциональные возможности: вычисления в таблицах, сортировку данных, графическое и текстовое оформление ячеек, создание диаграмм, стандартные операции с электронными документами (открытие, сохранение, печать и т. д.). Список наиболее распространенных табличных процессоров приведен в табл. 6.1.

Таблица 6.1. Наиболее распространенные табличные процессоры

Название ТП	Пакет прикладных программ, в состав которого входит ТП	Производитель	Базовая операционная система
Spreadsheet	Apple Works	Apple	Mac OS
Quattro Pro	Word Perfect Office	Corel	Windows
Star Gale	Star Office	Star Division	Linux, Windows
Excel	Microsoft Office	Microsoft	Windows
KSpead	Office	KDE Group	Linux

Электронные таблицы по своей природе являются более сложным типом документов, чем электронные тексты. Конечно,

практически во всех редакторах электронных таблиц есть программные компоненты, обеспечивающие преобразование документов из одного формата хранения в другой. Наиболее распространена возможность сохранения и открытия файлов, записанных в формате Excel (\*.xls). Однако на практике таблицы, содержащие сложные формулы или богатое текстовое оформление, нередко преобразуются с ошибками и утрачивают в лучшем случае респектабельный вид, а в худшем — данные и формулы. Информация, созданная в документе «надстройками» над таблицами, часто не перекодируется вовсе. Выбирая редактор электронных таблиц для решения той или иной задачи, следует учитывать эти особенности.

## 6.2. Принципы функционирования электронных таблиц ТП MS EXCEL

Документ, создаваемый и обрабатываемый с помощью табличного процессора MS Excel, хранится во внешней памяти компьютера как файл с расширением \*.xls.

### 6.2.1. Объекты рабочей книги

Документ MS Excel так же, как и документ MS Word, состоит из объектов, каждый из которых обладает своими свойствами. При создании нового файла табличный процессор самостоятельно присваивает файлу название «КнигаN.xls». Файл MS Excel представляет собой рабочую книгу — набор листов. Каждый из листов рабочей книги является местом для хранения и обработки данных. Каждый лист состоит из ячеек, образованных пересечением столбцов и строк. Иерархию этих основных объектов рабочей книги можно представить следующим образом (рис. 6.2).

**Объект Лист.** Основной вид листа, используемого табличным процессором, — это электронная таблица, называемая обычно рабочим листом (РЛ), которая состоит из ячеек, организованных по строкам и столбцам.

В состав рабочей книги могут включаться также листы диаграмм. Табличный процессор вставляет лист диаграмм в книгу

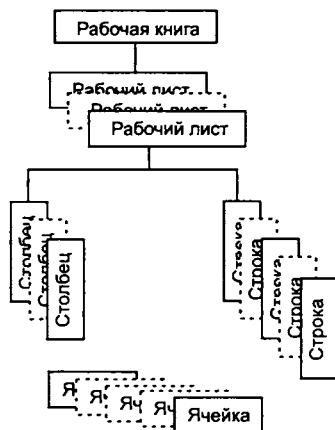


Рис. 6.2. Иерархия основных объектов рабочей книги Excel

слева от листа, содержащего данные, на которых основана диаграмма. Когда лист диаграмм активен, в него можно добавлять данные, выделять, форматировать, перемещать, изменять размеры входящих в него элементов. Листы диаграмм связаны с данными ЭТ и обновляются при изменении данных. Существуют также листы-модули. Модуль — это лист книги, содержащий набор макросов (инструкций) для выполнения конкретных действий.

Итак, рабочий лист представляет собой основной объект, предназначенный для размещения данных, подлежащих обработке с помощью табличного процессора. Лист состоит из ячеек, организованных в строки и столбцы. В каждом листе содержится  $2^8$  (256) столбцов и  $2^{16}$  (65 536) строк. Следует иметь в виду, что при удалении столбцов и/или строк ТП самостоятельно вставляет в конец листа новый столбец (строку), так что общее их число в таблице остается неизменным. Количество листов в новом файле может быть задано пользователем с помощью команды **Сервис/Параметры/Общие/Количество листов в книге**. В процессе работы листы могут добавляться в книгу или удаляться из нее.

Одним из свойств листа является его имя, которое отображается на ярлычках в нижней части листа рядом с линейкой горизонтальной прокрутки. Стандартные имена листов: **Лист1**, **Лист2**, ... . Эти имена ТП присваивает листам при открытии нового файла.

Листы можно переименовывать, вставлять, удалять, перемещать, копировать в пределах книги или из одной книги в другую.

**Объект Столбец.** Столбец представляет собой полосу из 65 536 ячеек, расположенных вертикально. Для идентификации столбцов используются символы латинского алфавита. Первые 26 столбцов обозначаются одним символом (от A до Z), а последующие — двумя (от AA — 27-й столбец до IV — 256-й).

Важным свойством столбцов является их ширина, которая измеряется в условных линейных единицах, кратных линейным размерам пикселя. Пиксель представляет собой единичный элемент изображений, создаваемых на экране аппаратными средствами монитора компьютера. Такие элементы, часто различимые как маленькие точки, образуют изображение на экране. Поэтому геометрические размеры пикселя на разных мониторах могут быть различными. Стандартная ширина столбца составляет 64 пикселя или 8,43 условных линейных единицы. С помощью команды **Формат/Столбец** можно установить необходимую ширину столбца, скрыть его (отобразить скрытый ранее) или установить режим автоподбора ширины.

**Объект Строка.** Строка представляет собой полосу из 256 ячеек, расположенных горизонтально. Для идентификации строк используются числа от 1 до 65 536, записанные с помощью арабских цифр.

Высота строк так же, как и ширина столбцов, измеряется в условных линейных единицах, кратных линейным размерам пикселя. Стандартная высота строки составляет 17 пикселей или 12,75 условных линейных единиц. С помощью команды **Формат/Строка** можно установить необходимую высоту строки, скрыть ее (отобразить скрытую ранее) или установить режим автоподбора высоты.

**Объект Ячейка.** Ячейка представляет собой основной объект рабочего листа, предназначенный для хранения данных. Идентификация ячеек осуществляется по их имени, которое отображается в крайнем левом окне строки формул. ТП присваивает стандартные имена ячейкам при открытии нового файла. Эти имена состоят из обозначения имени столбца и номера строки, например A32, BZ639 и т. п. Кроме этого, пользователь может присвоить другое, более легкое для запоминания, имя, которое впоследствии можно будет использовать при создании формул, например **Приход**, **НДС** и т. д.

Кроме самих данных, в ячейке хранятся сведения об их формате. Формат ячейки определяется (рис. 6.3):

- типом и форматом данных, хранящихся в ней;
- способом выравнивания символов, отображающих данные, в ячейке;
- параметрами шрифтового оформления данных, хранящихся в ячейке;
- начертанием границ ячейки;
- видом ячейки, который определяется цветом и характером заливки.

Для задания формата ячейки используется команда **Формат/Ячейка**. Тип данных, а следовательно, и виды операций, которые могут выполняться над ними, определяются числовым форматом (Закладка **Число** диалогового окна **Формат ячеек**), который может принять одно из ряда возможных значений.

Общий числовой формат используется по умолчанию. В большинстве случаев числа, имеющие общий формат, отображаются только способом, которым они были введены. Однако, если ширины ячейки недостаточно для отображения всего числа, общий числовой формат округляет число или использует экспоненциальное представление для больших чисел.

Кроме того, табличный процессор MS Excel содержит ряд встроенных числовых форматов. Они объединены в категории:

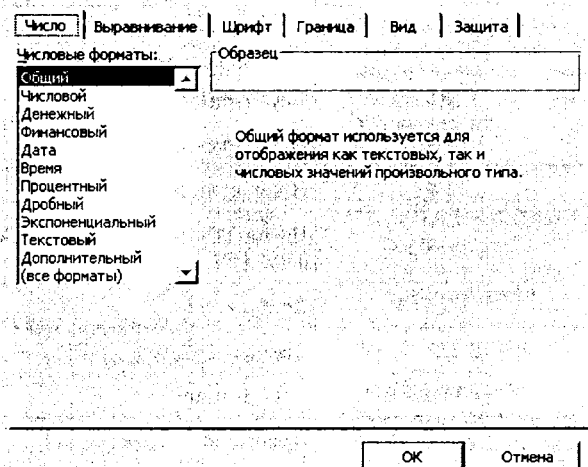


Рис. 6.3. Числовые форматы данных в ячейках



«Денежный», «Дата», «Время», «Дробный», «Текстовый» и «Экспоненциальный». Категория **Дополнительный формат** включает в себя почтовые индексы и телефонные номера. Данные, которым присвоен формат «Текстовый», обрабатываются как символьные, а остальные как числовые (См. гл. 2 «Теоретические основы обработки данных»).

Часто ТП автоматически присваивает правильный формат вводимым числам. Например, когда вводится число, содержащее символ \$ перед ним или знак % после него, ТП автоматически изменяет формат ячейки соответствующим образом. В случае, когда длина вводимого числа превышает ширину ячейки, ТП отображает его последовательностью знаков #####. Дата и время тоже могут быть представлены в различных форматах. Поскольку эти данные являются числовыми, с ними можно выполнять различные арифметические и логические операции.

Понимание зависимости формы отображения данных, введенных в таблицу, от используемого формата имеет существенное значение для правильной работы в ТП. Особенно актуально это для чисел, дат и времени. Поэтому ниже приведены табл. 6.2, 6.3, иллюстрирующие эти зависимости.

Таблица 6.2. Отображение чисел в таблицах

Вводимое число	Изображение введенного числа в таблице	Используемый формат
1.0	1	Общий
-1.1230	-1.123	Общий
1,581	1,581	Числовой#,##0
1.1%	1.10%	Процентный 0.00%
1.112%	1.11%	Процентный 0.00%
10e2	1.00E+03	Научный (экспоненциальный) 0.00E+00
0 1/2	1/2	Дробь # ?/?
3 3/5	3 3/5	Дробь # ?/?
3/4	4-Mar	Дата d-mmm
\$4.50	\$4.50	Валюта \$#,## 0.00;
-\$78.91	(\$78.91)	Валюта [Red](\$#,##0.00)

Таблица 6.3. Отображение дат в таблицах

Вводится	Формат Microsoft Excel
03/04/1994	m/d/yy
4-Mar-94	d-mmm-yy
3/4 or Mar-4	d-mmm
Mar-94	mmm-yy
8:50 PM	h:mm AM/PM
8:50:35 PM	h:mm:ss AM/PM
20:50	h:mm
20:50:35	h:mm:ss
03.04.94 20:50	m/d/yy h:mm

Более подробные сведения о назначении различных форматов данных и особенностях их использования приведены в табл. 6.4.

Таблица 6.4. Назначение форматов отображения данных

Наименование формата	Назначение	Примечания
Общий	Для отображения как текстовых, так и числовых констант произвольного типа	
Числовой	Наиболее общий формат для представления чисел	При использовании данного формата в отображаемое число может быть введен разделитель групп разрядов, а также различное число десятичных знаков
Денежный	Для отображения денежных значений чисел	
Финансовый	Для отображения денежных значений чисел	В отличие от денежного осуществляется выравнивание отображаемого числа по разделителю целой и дробной части
Дата	Для отображения дат	
Время	Для отображения времени	
Дробный	Для отображения дробей	Отображение может осуществляться простыми дробями (до трех знаков), долями (половинами, четвертями, восьмыми, шестнадцатыми, десятыми и сотыми)

Наименование формата	Назначение	Примечания
Экспоненциальный	Для отображения длинных чисел	7,89E+08 = 78 900 000 000
Текстовый	Для того чтобы данные обрабатывались как строка символов вне зависимости от их содержания (текст или число)	
Дополнительно	Для работы с базами данных и списками адресов	Например, почтовый индекс, № телефона и т. п.

При оформлении таблиц существенное значение приобретает их внешний вид, в частности, расположение данных в ячейке. Необходимый эффект достигается в результате использования закладки **Выравнивание** диалогового окна **Формат ячеек** (рис. 6.4).

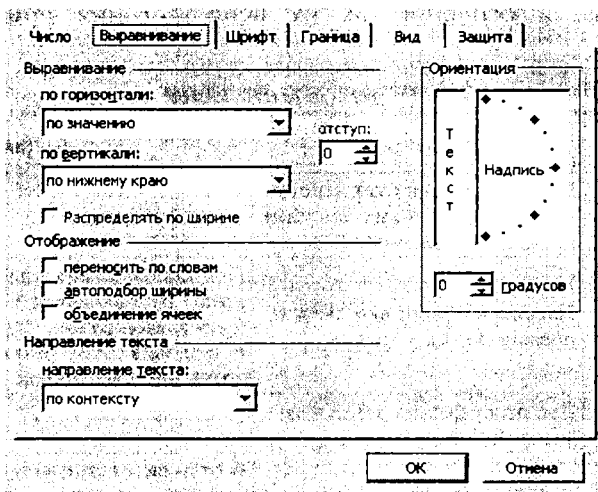


Рис. 6.4. Параметры выравнивания данных в ячейках

Представляется, что значение команд, отображаемых в данном диалоговом окне, не нуждается в пояснениях.

Шрифтовое оформление данных в ячейках при просмотре таблиц осуществляется так же, как и при работе в текстовом процессоре MS Word.

Итак, каждая ячейка имеет собственное имя и содержит данные, сведения об их форматах и оформлении ячейки. Кроме

того, практика показывает, что при работе с документом Excel возникает необходимость вставки в него примечаний. Для того чтобы обеспечить возможность сохранения всех этих сведений, ячейка представляет многослойный объект.

Самый верхний слой ячейки (слой значений) пользователь видит в таблице. В следующем (слой форматов) — хранятся сведения о форматах данных и оформлении ячейки. Третий слой (слой формул) содержит информацию о том, каким образом были введены данные в ячейку: непосредственно, с помощью клавиатуры или получены в результате вычисления по соответствующему алгоритму, заданному формулой, которая была введена в эту ячейку. Информация, записанная в этом слое, отображается в крайнем правом окне строки формул. Четвертый слой содержит примечания, а последний — имя ячейки.

**Диапазоны ячеек.** Для выполнения ряда операций в ТП Excel необходимо указать одну или несколько ячеек, содержащих данные для обработки. Это осуществляется путем их выделения с помощью курсора. Курсор — это выделенный прямоугольник, отмечающий активную ячейку. Две или более ячейки таблицы образуют диапазон. Диапазоны ячеек могут быть связными, когда ячейки примыкают друг к другу, или несвязными — расположенными порознь. Ячейки в диапазоне могут располагаться в одном столбце, в одной строке или в нескольких столбцах и строках, образующих прямоугольную область. Использование диапазона ячеек позволяет обеспечить существенную экономию времени и сил при работе в ЭТ. Например, можно одновременно форматировать все ячейки диапазона или вводить в них данные.

Для идентификации массивов могут использоваться их адреса или имена. Диапазоны именуются так же, как и ячейки. При записи адресов диапазонов необходимо руководствоваться следующими синтаксическими правилами:

- связный диапазон ячеек обозначается именами первой и последней ячеек, разделенными двоеточием — B2:D2 или B4:D7 (рис. 6.5);
- объединение связных (оператор запятая): (B2:D2,B4:D7);
- пересечение диапазонов (оператор пробел): (B11:D11 C10:C12) — значение C11;
- если в адресе диапазона лист не указан, то он определен на текущем листе. Пример указания листа в адресе: Лист6!A2:A5.

							Диапазон B2:D2		
							Диапазон B4:D7		
							Пересечение диапазонов B11:D11 и C10:C12		
							Обозначение B11:D11 C10:C12		
							Значение C:11		

Рис. 6.5. Обозначение диапазонов ячеек

Адреса ячеек и областей нужны для использования в формулах. Связный диапазон ячеек является массивом.

**Графические объекты.** Табличный процессор дает возможность пользователю применять для оформления таблиц те же самые средства, что и текстовый процессор MS Word. Ввиду того что диаграммы представляют собой мощное средство анализа табличных данных, особенности этой категории графических объектов будут рассмотрены ниже.

### 6.2.2. Ввод данных в таблицы

Табличный процессор MS Excel позволяет создавать таблицы для последующей обработки данных, содержащихся в них, двумя способами: путем импорта и вводом их с помощью клавиатуры.

Для импорта данных в Microsoft Excel из большинства источников нужно выбрать команду **Импорт внешних данных** в меню **Данные**, затем выбрать подпункт **Импортировать данные**, после чего в диалоговом окне **Выбор источника данных** выбрать данные, которые требуется импортировать.

Для ввода данных с помощью клавиатуры необходимо выделить ячейку, в которую эти данные следует поместить, набрать их и закончить ввод нажатием клавиши ENTER или клавиши табуляции. Данные, введенные таким способом, могут быть изменены только при последующем редактировании. Поэтому их иногда не совсем корректно называют константами. Изображения таких данных в слое значений и в слое формул совпадают.

**Особенности ввода текста.** В Microsoft Excel текстом является любая последовательность, состоящая из цифр, пробелов и нецифровых символов, например, приведенные ниже записи интерпретируются как текст:

**10AA109, 127AXY, 12-976, 208 4675.**

Текст, введенный в ячейку, выравнивается в ней по левому краю. Изменить выравнивание можно путем форматирования данных в ячейке. Для того чтобы начать в ячейке новую строку, следует нажать клавиши **ALT+ENTER**.

**Особенности ввода чисел.** Табличный процессор MS Excel интерпретирует все вводимые в ячейку числа как константы. При этом предполагается, что число может состоять только из следующих символов: **0 1 2 3 4 5 6 7 8 9 + - ( ) , / \$ % . E e**. Знак плюс (+), расположенный перед числом, игнорируется, а запятая служит разделителем десятичных разрядов. Все другие сочетания клавиш, состоящие из цифр и нецифровых символов, рассматриваются как текст.

При вводе числовых констант необходимо соблюдать следующие правила:

- перед рациональной дробью следует вводить **0** (ноль), чтобы избежать ее интерпретации как формата даты; например, **0 1/2**;
- перед отрицательным числом надо ввести знак минус (-) или заключать такое число в круглые скобки ( );
- введенные числа выравниваются в ячейке по правому краю. Изменить выравнивание можно путем форматирования данных в ячейке.

Пользователю необходимо учитывать, что формат числа, назначаемый ячейке, определяет способ просмотра числа на листе. В ячейках, имеющих общий числовой формат, отображаются целые числа (**789**), десятичные дроби (**7,89**) или числа, представленные в экспоненциальной форме (**7,89E+08**), если число длиннее ширины ячейки. Используя общий формат, можно просматривать до 11 разрядов, включая десятичные запятые и такие символы, как **E** и **+**. Кроме того, независимо от количества отображаемых разрядов числа хранятся с точностью до 15 разрядов. Если число имеет больше 15 значащих знаков, то разряды после 15-го преобразуются в нули (0).

В ряде случаев возникает необходимость обрабатывать числа, например, инвентаризационные номера, как текст. Для этого

необходимо назначить текстовый формат незаполненным ячейкам. Только после этого следует вводить числа.

**Особенности ввода дат и времени суток.** В табличном процессоре MS Excel предполагается, что даты и время суток представляют собой числовые данные. Внешнее представление времени или даты на листе зависит от числового формата, назначенного ячейке. При вводе значений даты или времени происходит их автоматическое распознавание, и общий формат ячейки заменяется встроенными форматами даты или времени. Если не происходит автоматического распознавания формата даты или времени, то введенные значения интерпретируются как текст, который выравнивается в ячейке по левому краю.

Независимо от формата, используемого для представления даты или времени, в Microsoft Excel все даты сохраняются как последовательные числа и время сохраняется в виде десятичной дроби. Поэтому время суток и даты можно складывать, вычитать и производить с ними другие вычисления. В формулы они могут быть введены в виде текста, заключенного в двойные кавычки. Например, приведенная ниже формула возвращает результат, равный 68:

**= "05.12.94" - "03.05.94"**

В Microsoft Excel используется система отсчета дат с начала столетия; в этой системе числу 1 соответствует дата **1 января 1900 г.**

**Ряды автозаполнения.** Табличный процессор MS Excel предоставляет пользователю возможность автоматического заполнения ячеек с помощью так называемых рядов автозаполнения нескольких типов. При этом ТП увеличивает значения данных, расположенных в исходной ячейке. Автоматическое заполнение осуществляется с помощью соответствующего маркера, расположенного в нижнем правом углу выделенной ячейки (рис. 6.6).

С этой целью выделяется ячейка, в которую введено начальное значение ряда данных, курсор устанавливается на маркер автозаполнения (при этом он принимает форму перекрестия), заполняются смежные ячейки путем перетаскивания маркера при нажатой левой клавише мыши. Пример рядов автозаполнения приведен в табл. 6.5, где в левой колонке указано начальное значение данных ряда, а в правой — его продолжение. Элементы продолжения ряда, разделенные запятыми, находятся в смежных ячейках.

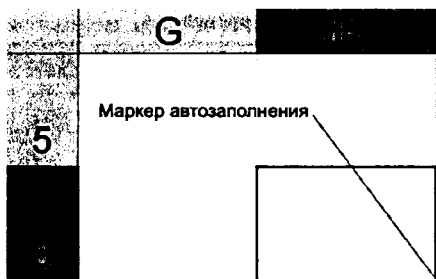


Рис. 6.6. Маркер автозаполнения (выделена ячейка H6)

Таблица 6.5. Пример рядов автозаполнения

Начальное значение	Продолжения ряда
пн	вт, ср, чт,...
кв.3	кв.4, кв.1, кв.2...
квартал 3	квартал 4, квартал 1, квартал 2
1-й период	2-й период, 3-й период,...
товар 1	товар 2, товар 3,...

В некоторых случаях для автозаполнения необходимо указать два начальных значения ряда или более, которые должны располагаться в смежных ячейках (табл. 6.6).

Таблица 6.6. Пример рядов автозаполнения с 2-мя и более начальными значениями

Начальное значение	Продолжения ряда
1-январь, 1-март	1-май, 1-июль, 1-сентябрь...
1; 2	3; 4; 5; 6...
1; 3; 4	5,66; 7,16; 8,66...

При использовании для автозаполнения рядов времени применяются приращения по дням, неделям или месяцам, которые можно указать, или predetermined последовательности, например, дни недели, названия месяцев или кварталы. Пример построения рядов времени с использованием функции автозаполнения приведен в табл. 6.7.



Таблица 6.7. Пример построения рядов времени

Начальное значение	Продолжение ряда
9:00	10:00, 11:00, 12:00
пн	вт, ср, чт
понедельник	вторник, среда, четверг
янв	фев, мар, апр
янв, апр	июл, окт, янв
янв-96, апр-96	июл-96, окт-96, янв-97
15-янв, 15-апр	15-июл, 15-окт
1994, 1995	1996, 1997, 1998

Очевидно, что с помощью автозаполнения легко создаются числовые ряды, представляющие собой арифметическую или геометрическую прогрессию. При создании арифметической прогрессии с помощью перетаскивания маркера заполнения значения увеличиваются или уменьшаются на постоянную величину, основанную на указанных начальных значениях, при создании геометрической — умножаются на постоянный множитель (табл. 6.8, 6.9).

Таблица 6.8. Примеры арифметических прогрессий

Начальное значение	Продолжение ряда
1; 2	3; 4; 5
1; 3	5; 7; 9
100; 95a	90; 85

Таблица 6.9. Примеры геометрических прогрессий

Начальное значение	Продолжение ряда
1;2	4; 8; 16
1;3	9; 27; 81
2;3	4,5; 6,75; 10,125

### 6.2.3. Использование формул для ввода данных в таблицы

Табличный процессор MS Excel позволяет вводить не только те данные, значения которых известны к моменту ввода, но и те, которые предстоит вычислить. С этой целью используются формулы. Формула представляет собой выражение, в соответствии с которым вычисляется значение той ячейки, в которой данная формула записана. Порядок, в котором вычисляется новое значение, называется синтаксисом формулы. Ввод формулы в ячейку должен начинаться со знака равенства (=). Вслед за ним вводится набор величин, над которыми должны быть выполнены определенные операции. Эти величины называются операндами, а символы, обозначающие действия, выполняемые над операндами, называются операторами. Таким образом, формула представляет собой последовательность операндов и операторов, которые создают в ячейках таблицы новое значение из уже введенных. В качестве операндов могут быть использованы значения (константы), ссылки на ячейки, имена ячеек и функции. В следующем примере представлена формула, в которой использованы различные операнды:

$$=A5/11-18*12+ \sin(A5) \quad (6.1)$$

Операндами в этой формуле служат ссылки на ячейку **A5**, константы **11**, **18** и **12** и функция  $\sin X$ , с аргументом, значение которого записано в ячейке **A5**.

Вычисления выполняются слева направо, начиная со знака равенства (=), в соответствии с приоритетом операций. Используя синтаксис написания формулы, можно управлять процессом вычисления. Например, следующая формула возвращает число 11, так как умножение имеет больший приоритет над сложением и поэтому выполняется в первую очередь: сначала происходит умножение 2 на 3 (результат умножения равен 6), а затем полученное значение складывается с 5:

$$=5+2*3 \quad (6.2)$$

Если для изменения синтаксиса воспользоваться скобками, то сначала произойдет сложение 5 и 2, а затем умножение полученного результата на 3. Следующая формула вернет число 21:

$$=(5+2)*3 \quad (6.3)$$

При вычислении формул табличный процессор MS Excel способен определенным образом воспринимать тип данных, включенных в нее. Рассмотрим несколько примеров.

#### Пример 1.

$$="1"+"2" \quad (6.4)$$

В результате выполнения действий над операндами будет получено число 3. Дело в том, что при использовании знака плюс (+) предполагается, что операнды являются числами. Даже если они записаны как числа в кавычках как текст ("1" и "2" — текстовые значения), Microsoft Excel автоматически преобразует их в числа.

#### Пример 2.

$$=1+"$4.00" \quad (6.5)$$

Результат вычисления — число 5. Это объясняется тем, что когда операнды должны быть числами, текстовые значения преобразуются, если это возможно, в числа.

#### Пример 3.

$$="6/1/2001"- "5/1/2001" \quad (6.6)$$

Результат вычисления — число 31. В данном случае Microsoft Excel интерпретирует текст как даты, записанные в формате "dd.mm.yy", а затем преобразует их в числа для вычисления количества дней, прошедших между этими двумя датами.

#### Пример 4.

$$=КОРЕНЬ("8+1") \quad (6.7)$$

Выполняя вычисления в соответствии с данной формулой, ТП вернет в ячейку значение ошибки #ЗНАЧ!. Это объясняется тем, что в данном случае текстовое значение «8+1» не преобразуется в число. Для правильного выполнения формулы необходимо указать следующие варианты: «9» или «8»+«1»; тогда формула выполнит преобразование текста в число и вернет в качестве результата 3.

**Пример 5.**

$$="A"&ИСТИНА \quad (6.8)$$

Результат вычисления — текст АИСТИНА. В том случае, когда операнды должны быть текстовыми значениями (в данном примере это определено оператором & — амперсанд), выполняется преобразование чисел и логических значений (ИСТИНА и ЛОЖЬ) в текст.

Очевидно, что для вычисления нового значения могут быть использованы данные, уже введенные в одну или несколько ячеек таблицы. В таких случаях табличный процессор MS Excel позволяет в качестве операндов формулы использовать имена этих ячеек или ссылки на них, используя их как идентификаторы ячейки (группы ячеек). В первом случае в качестве идентификатора ячейки (группы ячеек), используется текстовое обозначение, а во втором — ее адрес.

Ячейка, содержащая формулу, называется зависимой, так как ее значение зависит от значения другой ячейки, ссылка на которую является операндом. Преимущество использования ссылок заключается в том, что формула пересчитывается при изменении данных в той ячейке, ссылка на которую содержится в формуле. Так, пример, приведенный в следующей формуле, демонстрирует, что значение в зависимой ячейке будет получено путем умножения значения ячейки B15 на число 5. Формула будет пересчитываться при изменении значения ячейки B15:

$$=B15*5 \quad (6.9)$$

Таким образом, в формуле и имя ячейки, и ссылка могут быть использованы для указания того, какое значение данных используется для вычислений. При этом использование имени имеет некоторое преимущество, так как имя является уникальным идентификатором. Использование ссылок имеет некоторые особенности, которые нуждаются в пояснении.

**6.2.4. Относительные и абсолютные ссылки**

Рассмотрим пример. На рис. 6.7 приведен фрагмент таблицы, содержащей сведения о сотрудниках организации. Использование ссылок позволяет легко вычислить значение налога на

	A	B	C	D	E	F	G	H
1								
2	Номер отдела	Фамилия	Принадлежность к штату	Пол	Год рождения	Оклад	Налог на доходы физ.лиц	ФОРМУЛА
3	3	Бочин Г.Г.	с	м	1942	5 600,00р.	728,00р.	=13%*F3
4	3	Денисова О.К.	ш	ж	1944	7 800,00р.	1 014,00р.	=13%*F4
5	3	Куликова Р.А.	ш	ж	1944	9 200,00р.	1 196,00р.	=13%*F5
6	4	Смелов С.Д.	с	м	1944	9 700,00р.	1 261,00р.	=13%*F6

Рис. 6.7. Пример вычислений с использованием относительных ссылок

доходы физических лиц (ставка налога — 13 % от суммы дохода) для каждого из сотрудников.

Очевидно, что в этом случае формулу для вычисления значения величины налога на доходы физических лиц в ячейке G3 для Бочина Г. Г. можно записать:

$$=13\%*F3 \quad (6.10)$$

Табличный процессор позволяет с помощью маркера автозаполнения скопировать эту формулу в остальные ячейки этого столбца. Нетрудно убедиться, что в этом случае при копировании программа автоматически изменяет ссылку. Это показано в столбце «Формула» (см. рис. 6.7). Однако в случае изменения ставки налога, например на 11 %, в формуле (6.10) придется заменить константу, а затем повторить копирование формулы. Очевидно, что при достаточно большом количестве строк в таблице такое копирование может вызвать определенные трудности. На первый взгляд для решения этой проблемы достаточно записать величину ставки налога в отдельную ячейку и использовать для создания формулы не константу (величину ставки налога) и ссылку на адрес ячейки, где записано значение оклада сотрудника, а две ссылки с адресами ячеек (рис. 6.8):

$$=F1*F3 \quad (6.11)$$

Нетрудно убедиться, что копирование формулы из ячейки G3 в ячейки G4, G5 и т. д. не приведет нас к желаемому результату.

	A	B	C	D	E	F	G	H
1	Величина ставки налога на доходы физических лиц					13%		
2	Номер отдела	Фамилия	Принадлежность к штату	Пол	Год рождения	Оклад	Налог на доходы физ.лиц	ФОРМУЛА
3	3	Бочин Г.Г.	с	м	1942	5 600,00р.	728,00р.	=F1*F3
4	3	Денисова О.К.	ш	ж	1944	7 800,00р.	#ЗНАЧ!	=F2*F4
5	3	Куликова Р.А.	ш	ж	1944	9 200,00р.	61 620 000,00р.	=F3*F5
6	4	Смелов С.Д.	с	м	1944	9 700,00р.	75 660 000,00р.	=F4*F6

Рис. 6.8. Пример некорректного использования относительных ссылок

тату. Причиной этого является то, что при такой форме записи ссылок, табличный процессор автоматически изменяет порядковые номера обеих ссылок, что показано в столбце «Формула» (см. рис. 6.8). Для решения этой проблемы используется другая форма записи ссылок — так называемые абсолютные ссылки. В этом случае необходимая для решения нашей задачи формула должна иметь вид (рис. 6.9):

$$= \$F\$1 * F3 \quad (6.12)$$

	A	B	C	D	E	F	G
1	Величина ставки налога на доходы физических лиц					13%	
2	Номер отдела	Фамилия	Принадлежность к штату	Пол	Год рождения	Оклад	Налог на доходы физ. лиц
							ФОРМУЛА
	3	Бочин Г.Г.	с	м	1942	5 800,00р.	728,00р. = \$F\$1 * F3
4	3	Денисова О.К.	ш	ж	1944	7 800,00р.	1 014,00р. = \$F\$1 * F4
5	3	Куликова Р.А.	ш	ж	1944	9 200,00р.	1 196,00р. = \$F\$1 * F5
6	4	Смелов С.Д.	с	м	1944	9 700,00р.	1 261,00р. = \$F\$1 * F6

Рис. 6.9. Пример вычислений с использованием абсолютных и относительных ссылок

Последний пример показывает, что при копировании формулы, содержащей абсолютную ссылку, записанную с использованием символа \$ перед именем столбца и номером строки, адрес такой ячейки не изменяется. Преимущество использования формулы (6.12) заключается в том, что теперь нам для пересчета данных во всех ячейках столбца «Налоги на доходы физ.лиц» достаточно изменить только данные в одной ячейке F1.

Таким образом, при выполнении вычислений можно использовать два вида ссылок, различающихся формой записи.

Относительные ссылки используются в формулах для указания адреса ячейки, вычисляемого в относительной системе координат с началом в текущей ячейке. Форма записи относительных ссылок «Имя столбца — Номер строки». Относительные ссылки следует использовать при копировании в тех случаях, когда взаимное положение влияющих и зависимой ячеек остается неизменным.

Абсолютные ссылки используются в формулах для указания адреса ячейки, вычисляемого в абсолютной системе координат и не зависящего от текущей ячейки. Форма записи абсолютных ссылок «\$Имя столбца — \$Номер строки». Абсолютные ссылки следует использовать при копировании тогда, когда изменяется взаимное положение влияющих и зависимой ячеек.

При копировании формул в другие строки и столбцы иногда случается, что копируемые формулы содержат ссылки на ячейки, расположенные в этих строках или столбцах. При копировании относительные ссылки автоматически настраиваются, а абсолютные сохраняются. При использовании только относительных или только абсолютных ссылок, как правило, необходимо подправлять каждую формулу в отдельности. Однако при надлежащем применении смешанных ссылок Excel сможет правильно настроить формулы и без участия пользователя. При копировании смешанной ссылки ее часть с предшествующим символом доллара не изменяется, в то время как другая часть этой ссылки настраивается с учетом относительного положения копируемой формулы.

Итак, ссылка — это идентификатор ячейки (группы ячеек), использующий ее адрес. Ссылки делятся по расположению в книге на внутренние и внешние, а по способу определения местоположения — на относительные и абсолютные.

Использование ссылок для вычислений делает целесообразным присвоение постоянных имен листам и книгам при их сохранении. Дело в том, что если сначала создаются формулы, а потом сохраняется книга с новым именем, то формулы настраиваются соответствующим образом.

Например, если **Книга2** сохраняется как **Продажи.xls**, ссылка `=[Книга2]Лист2!$A$9` автоматически будет изменена на ссылку `=[Продажи.xls]Лист2!$A$9`. Если в этой же ссылке переименовать **Лист2** в **Февраль**, то она автоматически изменится на `=[Продажи.xls]Февраль!$A$9`. В том случае, если книга, на которую указывает ссылка, закрыта, то в ссылке появляется полный путь к папке, где хранится эта книга:

`'C:\Excel\[Продажи.xls]Февраль!$A$9`

Особого внимания заслуживает тот факт, что в последнем примере часть ссылки, указывающая на книгу и лист, заключена в одинарные кавычки. Объяснение этому находится в том, что при закрытии книги табличный процессор MS Excel автоматически заключает путь в одинарные кавычки. Поэтому при вводе новой ссылки на закрытую книгу следует добавлять одинарные кавычки самостоятельно. Во избежание возможных ошибок подобного рода целесообразно открыть закрытую книгу и с помощью мыши указать нужную ячейку. Такие действия обязывают Excel вставить внешнюю ссылку с соблюдением синтаксиса.

### 6.3. Понятие функции в электронных таблицах

Мощным инструментом, используемым в электронных таблицах, являются функции. Функции представляют собой заранее определенные формулы, которые выполняют вычисления по заданным величинам, называемым аргументами, и в указанном порядке. Например, функция СУММ суммирует значения в диапазоне ячеек, а функция ППЛАТ вычисляет величину выплаты за один период годовой ренты на основе постоянных выплат и постоянной процентной ставки. Функция входит в формулу в качестве операнда.

Для того чтобы табличный процессор MS Excel правильно распознал функцию и выполнил вычисления в соответствии с ее алгоритмом, необходимо строго соблюдать синтаксис — совокупность правил записи данной функции. Запись функции начинается с указания ее имени, затем вводится открывающая скобка, указываются аргументы, отделяющиеся точками с запятыми, а затем — закрывающая скобка. Если написание формулы начинается с функции, перед именем функции вводится знак равенства (=).

Аргументы функции — это величины, тип и порядок следования которых при записи строго должен соответствовать синтаксису функции. В качестве аргументов функция может содержать числа, текст, логические величины (например, **ИСТИНА** или **ЛОЖЬ**), массивы, значения ошибок (например, **#Н/Д**) или ссылки. Кроме того, аргументы могут быть как константами, так и формулами. Эти формулы, в свою очередь, могут содержать другие функции.

Всего в состав табличного процессора MS Excel включено более трехсот функций, которые для удобства объединены в 11 категорий (для Excel 2003).

1. Функции для работы с базами данных. Используются для анализа данных в списках. При этом в качестве базы данных рассматривается интервал ячеек, формирующий список или базу данных. В ТП MS Excel предполагается, что база данных представляет собой список связанных данных, в котором строки данных являются записями, а столбцы — полями. Верхняя строка списка содержит названия всех столбцов. Имена многих функций работы с базами данных часто начинаются с символа «Д».

2. Функции даты и времени. Служат для анализа и работы со значениями даты и времени в формулах.



3. Инженерные функции применяются для выполнения инженерного анализа. В этой категории можно выделить три группы функций: а) функции для работы с комплексными числами; б) функции для преобразования чисел из одной системы счисления в другую (десятичную, шестнадцатеричную, восьмеричную и двоичную); в) функции для преобразования величин из одной системы мер и весов в другую.

4. Финансовые функции позволяют выполнять типичные финансовые расчеты и используются для планирования и анализа финансово-хозяйственной деятельности предприятия, а также при решении задач, связанных с инвестированием средств.

5. Информационные функции предназначены для определения типа данных, хранимых в ячейке, проверяют выполнение какого-то условия и возвращают в зависимости от результата значение **ИСТИНА** или **ЛОЖЬ**.

6. Логические функции позволяют осуществить проверку выполнения одного или нескольких условий, что дает возможность реализовать алгоритмы выбора или цикла.

7. Функции просмотра служат для поиска нужных данных в списках или таблицах. С помощью данных функций можно получать текстовые или числовые значения, которые нецелесообразно или невозможно вычислить, но можно выбрать из таблицы на рабочем листе. Например, по названию товара или услуги можно определить их стоимость по прайс-листу и т. п.

8. Математические функции позволяют производить простые и сложные вычисления, например вычисление суммы диапазона ячеек, вычисление суммы ячеек диапазона, удовлетворяющих указанному условию, округление чисел и прочее. В их число входят арифметические, логарифмические и тригонометрические функции.

9. Статистические функции необходимы для выполнения статистического анализа диапазонов данных. Например, с помощью статистической функции можно провести прямую по группе значений, вычислить угол наклона и точку пересечения с осью  $Y$  и т. д.

10. Текстовые функции используются для выполнения действий над строками текста (например, изменения регистра или определения длины строки, объединения нескольких строк в одну и т. д.).

11. В ТП, кроме названных функций, могут использоваться внешние функции. Они функционируют в виде надстроек —

вспомогательных программ, служащих для добавления в Microsoft Office специальных команд или возможностей. Примером таких функций могут служить:

**EUROCONVERT** — осуществляет пересчет суммы в евро, пересчет из евро в национальную валюту страны, использующей евро, или пересчет из одной национальной валюты в другую с использованием евро в качестве промежуточной;

**SQL.REQUEST** — обеспечивает подключение к внешнему источнику данных и выполнение запроса из листа. Результат возвращается в виде массива. При этом дополнительное программирование не требуется.

В некоторых случаях необходимо использовать функцию как один из аргументов другой функции. Например, формула (6.13) использует вложенную функцию **СРЗНАЧ** и сравнивает результат со значением 50. При этом она содержит еще одну вложенную функцию **СУММ(G2:G5)**

$$=ЕСЛИ(СРЗНАЧ(F2:F5)>50,СУММ(G2:G5),0) \quad (6.13)$$

Вложенная функция, используемая в качестве аргумента, должна вычислять соответствующий этому аргументу тип данных. Например, если аргумент должен быть логическим, т. е. иметь значение либо **ИСТИНА**, либо **ЛОЖЬ**, то вложенная функция в результате вычислений тоже должна давать логическое значение либо **ИСТИНА**, либо **ЛОЖЬ**. Иначе появится сообщение об ошибке «#ЗНАЧ!».

В формулах можно использовать до семи уровней вложения функций. Если функция **Б** является аргументом функции **А**, то она имеет второй уровень вложенности. Например, в формуле (6.13) функции **СРЗНАЧ** и **СУММ** являются функциями второго уровня, потому что обе они являются аргументами функции **ЕСЛИ**. Функция, которая может оказаться вложенной в качестве аргумента в функции **СРЗНАЧ** или **СУММ**, будет функцией третьего уровня и т. д.

Во избежание синтаксических ошибок при вводе формул целесообразно использовать **Мастер функций** — пошаговую инструкцию, которая облегчает выполнение этой задачи. В этом случае ввод функции осуществляется в два этапа. По команде **Вставка/Функция...** появляется диалоговое окно **Мастер функций** (рис. 6.10), в котором пользователь должен выбрать функцию, необходимую ему для выполнения вычислений. При этом в

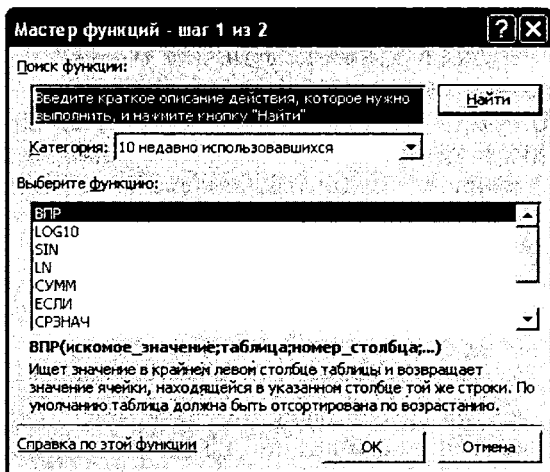


Рис. 6.10. Диалоговое окно Мастер функций (шаг 1)

нижней части окна отображается имя функции, ее описание и аргументы.

Кроме того, из этого окна с помощью гиперссылки можно вызвать справку по данной функции, в которой, кроме описания, приводятся примеры вычислений. После нажатия кнопки **OK** данного диалогового окна появляется следующее окно **Мастера функций** (рис. 6.11), где можно увидеть описание каждого аргумента, текущий результат функции и всей формулы. При выполнении этого шага справка по функции также остается дос-

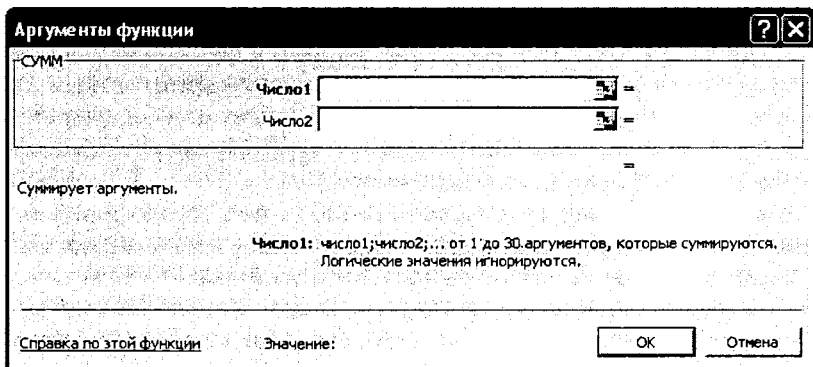


Рис. 6.11. Диалоговое окно Мастер функций (шаг 2)

тупной пользователю. Ввод функции заканчивается нажатием кнопки **ОК**.

При использовании ссылок в качестве аргументов пользователь должен учитывать возможность дальнейшего копирования создаваемой формулы. Для этого следует записывать их в диалоговом окне в абсолютной или относительной форме.

В тех случаях, когда пользователь не знает точно, какую функцию необходимо использовать для решения задачи, **Мастер функций** также может оказаться полезным. Достаточно на первом шаге ввести в поле «**Поиск функции**» краткое описание действия, которое нужно выполнить, и нажать кнопку **Найти**. Список функций, найденных в соответствии с составленным описанием, отображается в поле **Выберите функцию**.

## 6.4. Принципы вычислений с помощью функций

Безусловно, описать правила использования всех встроенных функций невозможно. Поэтому далее будут рассмотрены только наиболее часто используемые. Знание этих функций и использование справки MS Excel позволят пользователю решать практические задачи. В приложении приведен полный перечень функций табличного процессора MS Excel 2003 с указанием результатов их использования.

### 6.4.1. Использование математических функций

В категорию математических включено около шестидесяти функций, предназначенных для выполнения математических вычислений [1]. Для пользователя, знакомого с математическими дисциплинами (арифметикой, алгеброй, геометрией и тригонометрией) в объеме программы средней школы, применение этих функций не должно вызвать затруднений. Тем не менее представляется целесообразным отметить особенности использования некоторых функций, включенных в эту категорию.

Значительную часть функций, включенных в эту категорию, составляют тригонометрические функции. В их число входят прямые и обратные тригонометрические, а также гиперболические функции. Для вычисления этих функций следует ввести

только один аргумент — число. Для функций **SIN(число)**, **COS(число)** и **TAN(число)** аргумент **число** — это угол в радианах, для которого определяется значение функции. Если угол задан в градусах, его следует преобразовать в радианы путем умножения его на **ПИ()/180** или использования функции **РАДИАНЫ**.

Достаточно большое количество (примерно четверть) функций, включенных в эту категорию, позволяют обрабатывать такие массивы данных, как векторы и матрицы [1]. Объем данного пособия не позволяет подробно рассмотреть механизм работы всех функций. Однако представляется, что иллюстрации, приведенные ниже, позволят пользователю самостоятельно изучить действие тех функций, которые остались вне поля зрения:

**СУММЕСЛИ (диапазон;критерий;диапазон\_суммирования).**

**Диапазон** — диапазон ячеек, содержащий определенный признак.

**Критерий** — условие, записанное в форме числа, выражения или текста, определяющего требования к значению признака.

**Диапазон\_суммирования** — диапазон ячеек, значения данных в которых суммируются, если признак этих ячеек соответствует условию.

Таким образом, с помощью этой функции можно вычислить сумму значений, записанных в ячейках из «диапазона\_суммирования», если значения в соответствующих им ячейках «диапазона» удовлетворяют «критерию». Если «диапазон\_суммирования» опущен, то суммируются значения ячеек в «диапазоне». Пример действия этой функции показан на рис. 6.12.

Функция **СУММПРОИЗВ** также может оказать существенную помощь при обработке массивов данных. Ее действие за-

	А	В
1	Стоимость имущества	Величина налога на имущество
2	100 000р.	2 000р.
3	200 000р.	4 000р.
4	300 000р.	6 000р.
5	400 000р.	8 000р.
6	Итого надлежит собрать	18 000,00р.
7		
8	=СУММЕСЛИ(A2:A5;">160000";B2:B5)	

Рис. 6.12. Пример действия функции СУММЕСЛИ

ключается в вычислении суммы произведений соответствующих элементов заданных массивов. Синтаксис этой функции:

**СУММПРОИЗВ(массив1;массив2;массив3; ...)**

**Массив1, массив2, массив3, ...** — от 2 до 30 массивов, чьи компоненты нужно перемножить, а затем сложить.

Аргументы, которые являются массивами, должны иметь одинаковые размерности. Если это не так, то функция **СУММПРОИЗВ** возвращает значение ошибки **#ЗНАЧ!** При этом данная функция воспринимает нечисловые элементы массивов как нулевые. Пример использования функции приведен на рис. 6.13.

	A	B	C	D
	Вид	Цена	Количество	Итого
1	товара			
2	Товар1	10	5	
3	Товар2	20	6	
4	Товар3	30	7	
5	Общая стоимость товаров			380
6	=СУММПРОИЗВ(B2:B4;C2:C4)			
7				

Рис. 6.13. Пример вычислений с помощью функции **СУММПРОИЗВ**

В результате использования данной функции будет получено то же значение, что и при применении формулы **СУММ(B2\*C2; B3\*C3; B4\*C4)**.

В результате рассмотрения данных примеров можно прийти к выводу о том, что использование массивов дает более общее средство для выполнения вычислений.

## 6.4.2. Использование статистических функций

### 6.4.2.1. Прогнозирование с помощью статистических функций

Реальная деятельность практически в любой профессиональной области требует от руководителей принятия обоснованных решений. Для этого необходимы расчеты, связанные с прогнозами состояния рынка, эффективностью инвестиций, оценками возможных рисков и их последствий. Совокупность методов решения таких задач получила название математической статисти-

ки. Эти методы позволяют выявлять закономерности на фоне случайностей, делать обоснованные выводы и прогнозы, давать оценки вероятностей их осуществления или невыполнения. Наличие статистических функций в составе табличного процессора MS Excel дают подготовленному пользователю возможность использовать методы статистического анализа в профессиональной деятельности.

К числу распространенных задач математической статистики относятся задачи прогнозирования будущего поведения некоторого временного ряда: изменение курса валюты, цен и спроса на какие-либо виды сырья или продукции и т. п. К числу функций в составе электронных таблиц, предназначенных для решения задач такого рода, относятся функции **РОСТ** и **ТЕНДЕНЦИЯ**. Они рассчитывают возможное значение функций в будущем на основании имеющихся данных. При этом функция **РОСТ** предполагает наличие экспоненциальной зависимости значений функции (зависимой переменной) от величины аргумента (независимой переменной), а **ТЕНДЕНЦИЯ** — линейной. Рассмотрим пример (рис. 6.14), где приведены данные об изменении курса доллара в течение первых трех кварталов года, а нас интересуют те значения, которые он может принять в четвертом.

	А	В	С	Д	Е
1				Прогноз	
2	Месяц	№	Курс доллара	(функция РОСТ)	(функция ТЕНДЕНЦИЯ)
3	январь	1	34,02р.		
4	февраль	2	34,33р.		
5	март	3	34,42р.		
6	апрель	4	34,43р.		
7	май	5	34,55р.		
8	июнь	6	34,16р.		
9	июль	7	34,48р.		
10	август	8	35,01р.		
11	сентябрь	9	35,12р.		
12	октябрь	10		35,0271	35,0264
13	ноябрь	11		35,1333	35,1312
14	декабрь	12		35,2399	35,2361
15					
16				=РОСТ(С3:С11;В3:В11;В14)	
17				=ТЕНДЕНЦИЯ(С3:С11;В3:В11;В14)	
18					

Рис. 6.14. Пример использования статистических функций для прогнозирования

Графическая интерпретация данных представлена на диаграмме (рис. 6.15).

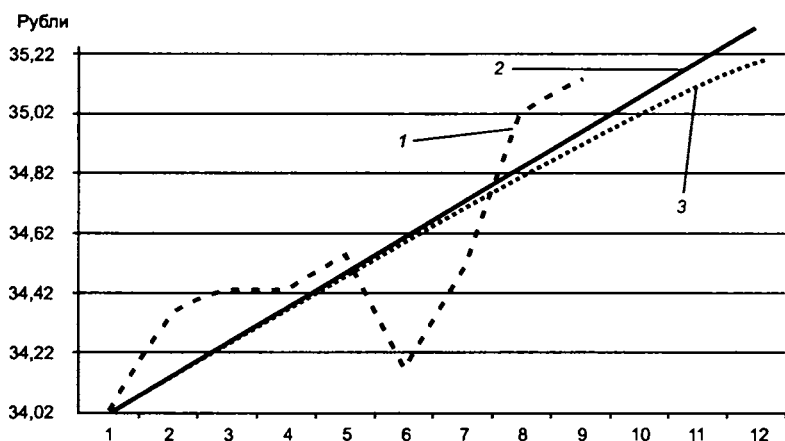


Рис. 6.15. Пример прогнозирования курса доллара с помощью статистических функций:

1 — фактический курс; 2 — рост; 3 — тенденция

Синтаксис этих функций практически совпадает:

**РОСТ**(известные\_значения\_y; известные\_значения\_x; новые\_значения\_x; конст);

**ТЕНДЕНЦИЯ**(известные\_значения\_y; известные\_значения\_x; новые\_значения\_x; конст).

При этом в качестве аргументов этих функций используются величины:

- **известные\_значения\_y** — это множество значений функции, которые уже известны для соотношения  $Y = f(X)$ ;
- **известные\_значения\_x** — это множество значений аргумента  $X$ , которые уже известны для соотношения  $Y = f(X)$ ;
- **новые\_значения\_x** — это новые значения аргумента, для которых функции **РОСТ** или **ТЕНДЕНЦИЯ** возвращают соответствующие значения функции  $Y$ ;
- **конст** — это логическое значение.

В случае использования функции **РОСТ** предполагается, что зависимость  $Y = f(X)$  описывается выражением  $y = b * m^x$ . Поэтому, если **Конст** имеет значение **ИСТИНА** или опущено, то **b** автоматически определяется в ходе вычисления функции, а если



**Конст** имеет значение **ЛОЖЬ**, то  $b$  считается равным 1, а значения  $m$  подбираются в процессе вычисления так, чтобы  $y = m^x$ .

Для функции **ТЕНДЕНЦИЯ** зависимость функции от аргумента описывается линейным уравнением  $y = mx + b$ . В этом случае, если **Конст** имеет значение **ИСТИНА** или опущено, то  $b$  автоматически определяется в ходе вычисления функции, а если **Конст** имеет значение **ЛОЖЬ**, то  $b$  полагается равным 0, и значения  $m$  подбираются таким образом, чтобы выполнялось соотношение  $y = mx$ .

#### 6.4.2.2. Определение частот наступления событий

Часто при анализе какого-либо явления или процесса может оказаться полезной группировка данных. Под группировкой следует понимать разбиение интервала, содержащего все результаты наблюдений, на некоторое число небольших интервалов, называемых интервалами группировки. Число наблюдений в некотором интервале группировки означает частоту их попадания в данный интервал. Графическое изображение зависимости частоты попадания элементов выборки в соответствующий интервал группировки называется гистограммой выборки. Табличная функция **ЧАСТОТА** позволяет вычислить частоту появления значений в интервале значений и ввести в таблицу массив чисел, отражающий результат этого вычисления. Функция **ЧАСТОТА** может быть использована, например, для подсчета количества результатов тестирования, попадающих в интервалы результатов. Существенная особенность данной функции заключается в том, что поскольку результатом ее вычисления является массив значений, она должна задаваться в качестве формулы массива. Пример вычисления этой функции и построения гистограммы на основе данных о рейтинговой оценке некоторой группы студентов приведен на рис. 6.16.

Синтаксис данной функции:

**ЧАСТОТА**(массив\_данных;массив\_интервалов),

где **массив\_данных** — массив или ссылка на множество данных, для которых вычисляются частоты. Для нашего примера это A2:A10;

**массив\_интервалов** — массив или ссылка на множество интервалов, в которые группируются значения аргумента **массив\_данных** (B2:B4).

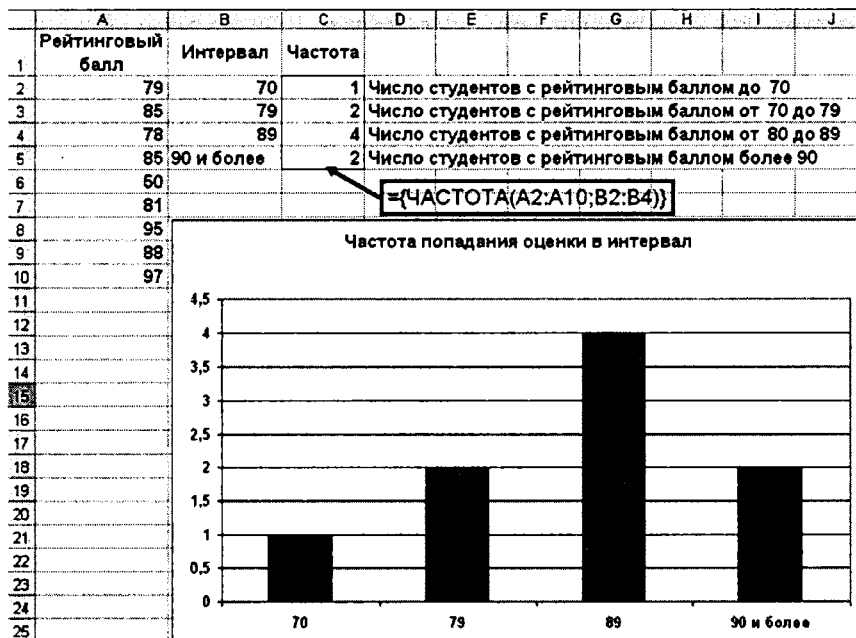


Рис. 6.16. Пример вычисления частоты событий и гистограмма

При использовании функции **ЧАСТОТА** ее следует вводить как формулу массива после выделения интервала смежных ячеек, в который будет размещен полученный массив распределения. Кроме того, количество элементов в возвращаемом массиве должно быть на единицу больше числа элементов в массиве **массив\_интервалов**. Это обусловлено тем, что дополнительный элемент в возвращаемом массиве содержит количество значений, превышающих максимальное значение в интервалах.

#### 6.4.2.3. Ранжирование данных

В деловой практике достаточно часто приходится принимать решения на основе данных, которые носят в той или иной степени условный характер. Это могут быть экспертные оценки, данные опроса и т. д. Объективную основу для выводов в подобных случаях могут дать только соотношения между такими данными. Например, при анализе данных с результатами опросов телезрителей можно сказать, что телепрограмма, получившая больший балл в анкете, данному зрителю нравится больше, чем

программа, отмеченная меньшим баллом. В то же время сказать, насколько (или во сколько раз) одна программа лучше другой, невозможно потому, что отсутствует объективная единица измерения зрительских симпатий. В подобных случаях в математической статистике исследуются только сведения о взаимной упорядоченности данных. С этой целью осуществляется переход от данных к их рангам. Рангом того или иного элемента данных называется тот порядковый номер, который получит этот элемент данных в упорядоченной совокупности подобных элементов после их упорядочения по определенному правилу. Процесс присвоения рангов называется ранжированием. Чаще всего упорядочение данных осуществляется по величине.

В табличном процессоре MS Excel для ранжирования используется функция РАНГ. Она присваивает ранг числам в списке. Синтаксис данной функции:

**РАНГ(число;ссылка;порядок),**

где **число** — число, для которого определяется ранг;

**ссылка** — массив или ссылка на список чисел. Нечисловые значения в ссылке игнорируются.

**Порядок** — число, определяющее способ упорядочения. Данный аргумент не является обязательным. По этой причине он обозначается обычным, а не жирным шрифтом. Если порядок равен нулю или опущен, то Microsoft Excel определяет ранг числа так, как если бы ссылка была списком, отсортированным в порядке убывания. В противном случае (порядок — любое нену-

	A	B	C	D	E
	Фамилия студента	Рейтинговая оценка	Ранг (порядок 0)	Ранг (порядок 1)	
2	Баранов	79	7	3	
3	Волкова	85	4	5	
4	Зайцев	78	8	2	
5	Иванова	85	4	5	
6	Лосев	50	9	1	
7	Львова	81	6	4	
8	Мамонтов	95	2	8	
9	Николаев	88	3	7	
10	Осипова	97	1	9	
11					
12	=РАНГ(B2:\$B\$2:\$B\$10;)				
13	=РАНГ(B2;\$B\$2:\$B\$10;1)				

Рис. 6.17. Пример использования функции РАНГ

левое число) ранг числа определяется так, как если бы ссылка была списком, отсортированным в порядке возрастания.

При использовании функции **РАНГ** повторяющимся числам присваивается одинаковый ранг. Однако наличие повторяющихся чисел влияет на ранги последующих чисел. Например, если в списке целых чисел, отсортированных по возрастанию, дважды встречается число 10, имеющее ранг 5, число 11 будет иметь ранг 7 (ни одно из чисел не будет иметь ранг 6). Пример использования данной функции приведен на рис. 6.17.

### 6.4.3. Использование логических функций

Опыт показывает, что специалисты различного профиля, использующие для решения своих профессиональных задач табличный процессор MS Excel, из всех логических функций чаще всего употребляют функции **И**, **ИЛИ** и **ЕСЛИ**. Объясняется это тем, что они позволяют в процессе решения задач организовать ветвление, т. е. реализовать выбор нескольких вариантов вычисления. Известно, что для организации ветвления используются высказывания. При этом простое высказывание содержит одно утверждение, что позволяет проверить выполнение только одного условия. Функции **И** и **ИЛИ** позволяют создавать сложные высказывания, с помощью которых можно проверить выполнение (или невыполнение) сразу нескольких условий. Рассмотрим примеры решения подобных задач.

Предположим, что в бюро трудоустройства, где ведутся списки желающих получить работу (фрагмент списка приведен на рис. 6.18) поступил запрос. Требования работодателя — образование высшее, возраст не более 25-ти лет.

Для отбора из этого списка кандидатов, соответствующих требованиям работодателя, можно использовать логическую функцию **И**. Ее действие заключается в том, что она присваивает значение **ИСТИНА**, если все аргументы имеют значение **ИСТИНА**. Если хотя бы один из аргументов имеет значение **ЛОЖЬ**, результатом будет значение **ЛОЖЬ**.

Синтаксис этой функции:

**И** (логическое\_значение1; логическое\_значение2; ...),

где логическое\_значение1, логическое\_значение2, ... — это от одного до тридцати проверяемых условий (простых высказыва-

	A	B	C	D
1	Фамилия	Пол	Образование	Год рождения
2	Беликов М.И.	м	в	1967
3	Бочкарева А.П.	ж	в	1982
4	Дерюгин С.С.	м	с/с	1985
5	Иванов П.П.	м	с	1976
6	Иванова С.В.	ж	с	1969
7	Бялко О.О.	ж	в	1970
8	Виноградова Т.Н.	ж	с/с	1981
9	Иванов П.П.	м	в	1983

Рис. 6.18. Фрагмент списка кандидатов на трудоустройство

ний), каждое из которых может иметь значение либо **ИСТИНА**, либо **ЛОЖЬ**.

Аргументы должны быть логическими значениями, массивами или ссылками, которые содержат логические значения. Если аргумент, который является ссылкой или массивом, содержит тексты или пустые ячейки, то такие значения игнорируются. Если указанный интервал не содержит логических значений, то **И** возвращает значение ошибки **#ЗНАЧ!**

Для решения нашей задачи следует ввести в ячейку E2 функцию **И(C2="в";2005-D2<=25)**. В этой формуле аргумент **C2="в"** соответствует высказыванию «Образование — высшее», а **2005-D2<=25** — «Возраст не более 25 лет». Копирование формулы в ячейки E3:E9 даст искомый результат (рис. 6.19).

Алгоритм исполнения этой функции можно представить следующим образом (рис. 6.20).

Рассмотрим действие функции **ИЛИ**. В то же бюро по трудоустройству поступил запрос на специалиста с высшим или средним специальным образованием. Для решения данной задачи следует использовать функцию **ИЛИ**. Эта функция вводит значение **ИСТИНА**, если хотя бы один из ее аргументов имеет значение **ИСТИНА**. Значение **ЛОЖЬ** вводится, если все аргументы имеют значение **ЛОЖЬ**. Синтаксис функции:

**ИЛИ (логическое\_значение1;логическое\_значение2; ...),**

где **логическое\_значение1**, **логическое\_значение2**, ... — как и в предыдущем случае, это от одного до тридцати проверяемых условий (простых высказываний), каждое из которых может иметь

E2	=И(C2="в";2005-D2<=25)				
	A	B	C	D	
1	Фамилия	Пол	Образование	Год рождения	кандидат
2	Беликов М.И.	м	в	1967	ЛОЖЬ
3	Бочкарева А.П.	ж	в	1982	ИСТИНА
4	Дерюгин С.С.	м	с/с	1985	ЛОЖЬ
5	Иванов П.П.	м	с	1976	ЛОЖЬ
6	Иванова С.В.	ж	с	1969	ЛОЖЬ
7	Бялко О.О.	ж	в	1970	ЛОЖЬ
8	Виноградова Т.Н.	ж	с/с	1981	ЛОЖЬ
9	Иванов П.П.	м	в	1983	ИСТИНА
10:					
11:			=И(C2="в";2005-D2<=25)		
12:					

Рис. 6.19. Результат отбора кандидатов на работу (Пример 1)

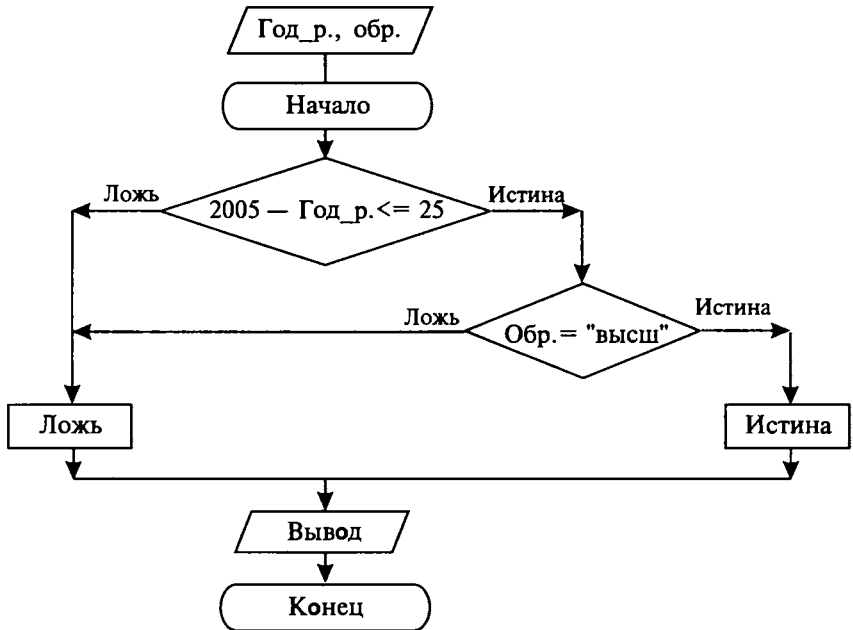


Рис. 6.20. Блок-схема алгоритма решения задачи отбора кандидатов (Пример 1)

значение либо **ИСТИНА**, либо **ЛОЖЬ**. Решение задачи представлено на рис. 6.21.

Блок-схема алгоритма решения приведена на рис. 6.22.

E2		=ИЛИ(C2="в";C2="с/с")			
	A	B	C	D	E
1	Фамилия	Пол	Образование	год рождения	кандидат
2	Беликов М.И.	м	в	1967	ИСТИНА
3	Бочкарева А.П.	ж	в	1982	ИСТИНА
4	Дерюгин С.С.	м	с/с	1985	ИСТИНА
5	Иванов П.П.	м	с	1976	ЛОЖЬ
6	Иванова С.В.	ж	с	1969	ЛОЖЬ
7	Бялко О.О.	ж	в	1970	ИСТИНА
8	Виноградова Т.Н.	ж	с/с	1981	ИСТИНА
9	Иванов П.П.	м	в	1983	ИСТИНА
10					
11			=ИЛИ(C2="в";C2="с/с")		
12					

Рис. 6.21. Результат отбора кандидатов на работу (Пример 2)

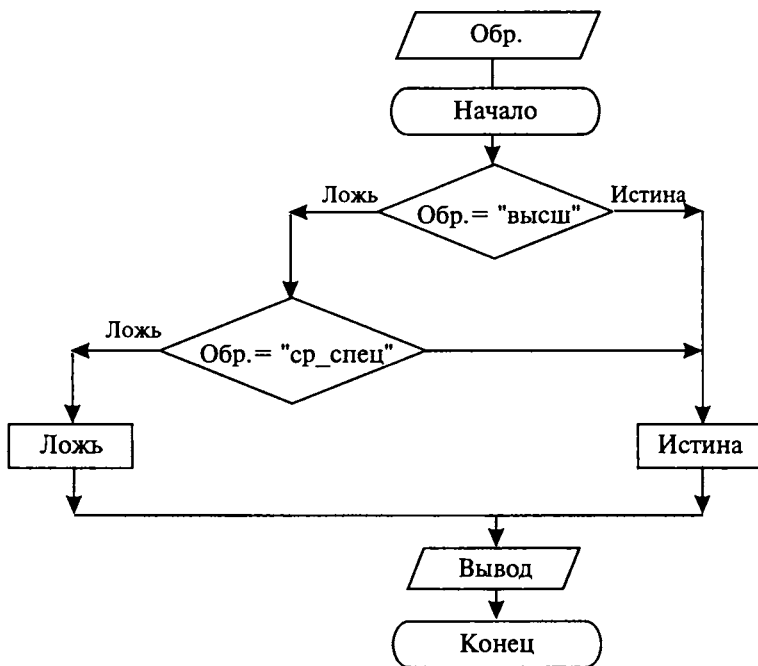


Рис. 6.22. Блок-схема алгоритма решения задачи отбора кандидатов (Пример 2)

При всех своих достоинствах функции **И** и **ИЛИ** имеют один существенный недостаток: в ячейки, где они вычисляются, вводятся логические значения. Данное обстоятельство иногда при-

водит к необходимости выполнения дополнительных операций. Поэтому при решении подобных задач удобнее использовать логическую функцию **ЕСЛИ**. Ее действие заключается в том, что она вводит в ячейку одно значение, если заданное условие при вычислении дает значение **ИСТИНА**, и другое значение — в противном случае. Как и другие логические функции, рассмотренные ранее, **ЕСЛИ** используется при проверке условий для значений и формул.

Синтаксис этой функции:

**ЕСЛИ(лог\_выражение;значение\_если\_истина;  
значение\_если\_ложь),**

где **лог\_выражение** — это любое значение или выражение (в том числе простые и сложные высказывания), принимающее значения **ИСТИНА** или **ЛОЖЬ**;

**значение\_если\_истина** — это значение, которое будет введено в вычисляемую ячейку, если **лог\_выражение** истинно. Это значение может быть формулой;

**значение\_если\_ложь** — это значение, которое будет введено в вычисляемую ячейку, если **лог\_выражение** ложно. Это значение может быть формулой.

Для уяснения действия этой функции рассмотрим очередной пример. На этот раз работодателю требуются женщины с высшим образованием и мужчины со средним специальным. Очевидно, что сложное высказывание, в котором сформулированы требования к кандидатам на работу, в данном случае можно записать следующим образом:

**ИЛИ(И(B2="м";C2="с/с");И(B2="ж";C2="в"))).**

Аргументами функции **ИЛИ** являются вложенные функции **И**, каждая из которых проверяет одновременное выполнение двух условий: пола и образования. Это сложное высказывание используется в качестве первого аргумента (**лог\_выражение**) функции **ЕСЛИ**. Второй и третий аргумент выбираются пользователем в зависимости от перспектив их последующего использования. В нашем случае, когда необходимо только обозначить в списке кандидатов, целесообразно в качестве второго аргумента (**значение\_если\_истина**) использовать текстовое значение «кандидат», а третьего (**значение\_если\_ложь**) — также текстовое значение « », оставляющее ячейку пустой. Результат решения задачи



представлен на рис. 6.23. При решении других задач можно использовать и другие значения. Например, при назначении премий, можно в качестве второго и третьего аргументов вписать формулы для вычисления размеров премии в зависимости от выполнения или невыполнения условий, записанных в первой строке шаблона функции.

Блок-схема алгоритма решения данной задачи приведена на рис. 6.24.

E2 =ЕСЛИ(ИЛИ(И(B2="м";C2="с/с");И(B2="ж";C2="в"));"кандидат";" ")					
	A	B	C	D	E
1	Фамилия	Пол	Образование	Год рождения	кандидат
2	Беликов М.И.	м	в	1967	
3	Бочкарева А.П.	ж	в	1982	кандидат
4	Дерюгин С.С.	м	с/с	1985	кандидат
5	Иванов П.П.	м	с	1976	
6	Иванова С.В.	ж	с	1969	
7	Бялко О.О.	ж	в	1970	кандидат
8	Виноградова Т.Н.	ж	с/с	1981	
9	Иванов П.П.	м	в	1983	
10	=ЕСЛИ(ИЛИ(И(B2="м";C2="с/с");И(B2="ж";C2="в"));"кандидат";" ")				
11					

Рис. 6.23. Результат отбора кандидатов на работу (Пример 3)

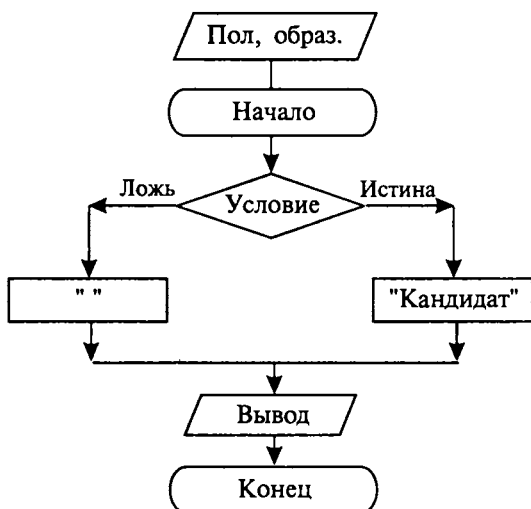


Рис. 6.24. Блок-схема алгоритма решения задачи отбора кандидатов (Пример 3)

#### 6.4.4. Использование финансовых функций

Все финансовые функции по характеру, выполняемым ими вычислениям можно разделить на три группы: а) функции для расчета доходов от ценных бумаг; б) функции для расчета амортизационных отчислений, т. е. отчислений, предназначенных для возмещения износа имущества и в) функции для расчета рентных платежей.

Некоторые функции могут использоваться для вычислений при операциях с ценными бумагами двух видов: твердопроцентными и беспроцентными. Твердопроцентные ценные бумаги приносят одинаковый доход на протяжении определенного периода времени. Беспроцентные ценные бумаги эмитируются с определенным дисконтом (скидкой), который выражается в процентах. Это означает, что курс при эмиссии ценных бумаг этого вида меньше номинальной стоимости, по которой происходит выкуп ценных бумаг в конце срока обращения.

Амортизация определяется как отчисления, предназначенные для возмещения износа имущества. Всякое имущество имеет определенную стоимость на начало периода амортизации и стоимость на конец периода амортизации — остаточную стоимость. Имущество амортизируется на протяжении определенного периода, который называется временем амортизации. Амортизационные отчисления используются в бухгалтерских расчетах для регулирования налоговых платежей в течение времени амортизации.

Под рентными платежами понимают регулярные платежи одинакового размера. Рента — вид дохода, не требующий от его получателя предпринимательской деятельности. Это платежи, обусловленные необходимостью возврата кредита, либо связанные с банковскими вкладами. С этим видом платежей приходится сталкиваться в повседневной жизни большинству граждан. Поэтому с действием финансовых функций будем знакомиться на примере этой группы функций.

Для понимания сущности рентных платежей рассмотрим схему, представленную на рис. 6.25.

Кредит — временная передача денег в долг. Однако на практике этот термин употребляется для обозначения денег, принадлежащих кредитному учреждению, которые передаются во временное пользование юридическим или физическим лицам. Те деньги, которые эти лица передают на хранение в банк, т. е. пре-



Рис. 6.25. Схема кредитных взаимоотношений

доставляют банку право пользования ими, называются вкладом. За пользование кредитом с получателя денег взимается плата в виде процента. Важнейшими параметрами, используемыми при расчете рентных платежей, являются сумма кредита (вклада), срок его выплаты, размер платы за пользование кредитом (процентная ставка), размер и периодичность платежей, погашающих кредит. Следует отметить, что финансовые функции расчета рентных платежей позволяют автоматизировать только простейшие случаи.

При работе с финансовыми функциями следует уметь правильно толковать некоторые понятия, использующиеся в повседневной финансовой деятельности. К числу таких понятий относится аннуитет, что следует воспринимать как ряд постоянных денежных выплат, делаемых в течение длительного периода. Например, заем под строительство дома или заклад квартиры являются аннуитетами. В функциях, связанных с аннуитетами, выплачиваемые денежные средства, такие, как депозит на сбережения, представляются отрицательным числом; полученные денежные средства, такие, как чеки на дивиденды, представляются положительным числом. Например, депозит в банк на сумму 1000 руб. представляется аргументом **-1000** — для вкладчика и аргументом **1000** — для банка. При расчете аннуитетов чаще всего используются следующие функции (табл. 6.10, которая подготовлена на основе [1]).

Таблица 6.10. Перечень наиболее часто употребляемых функций

Функция	Значение функции	Синтаксис функции
ПС	Вычисляет приведенную (к текущему моменту) стоимость инвестиции, которая представляет собой общую сумму, равноценную на настоящий момент ряду будущих выплат. Например, когда вы занимаете деньги, сумма займа является приведенной (нынешней) стоимостью для заимодавца	ПС(ставка; кпер; плт; бс; тип)

Окончание табл. 6.10

Функция	Значение функции	Синтаксис функции
БС	Вычисляет будущую стоимость инвестиции на основе периодических постоянных (равных по величине сумм) платежей и постоянной процентной ставки	БС(ставка; кпер; плт; пс; тип)
СТАВКА	Вычисляет процентную ставку по аннуитету за один период. СТАВКА вычисляется путем итерации и может давать нулевое значение или несколько значений. Если последовательные результаты функции СТАВКА не сходятся с точностью 0,0000001 после 20 итераций, то СТАВКА возвращает сообщение об ошибке #ЧИСЛО!	СТАВКА(кпер; плт; пс; бс; тип; предположение)
ПРПЛТ	Вычисляет сумму платежей процентов по инвестиции за данный период на основе постоянства сумм периодических платежей и постоянства процентной ставки	ПРПЛТ(ставка; период; кпер; пс; бс; тип)
КПЕР	Вычисляет общее количество периодов выплаты для инвестиции на основе периодических постоянных выплат и постоянной процентной ставки	КПЕР(ставка; плт; пс; бс; тип)
ПЛТ	Вычисляет сумму периодического платежа для аннуитета на основе постоянства сумм платежей и постоянства процентной ставки	ПЛТ(ставка; кпер; пс; бс; тип)

Данные этой таблицы показывают, что при вычислениях табличный процессор MS Excel учитывает возможность выразить одни финансовые аргументы через другие. Рассмотрим некоторые конкретные задачи, для решения которых используются эти функции.

#### 6.4.4.1. Задачи, связанные с получением кредита

**Задача № 1.** Предположим, что заемщику известны условия, на которых он может получить кредит. Необходимо оценить величину периодических платежей, которые ему придется выплачивать банку в погашение этого кредита.

При использовании финансовых функций вычисления удобнее выполнять, если записать условия задачи в определенном порядке (рис. 6.26).

	A	B	C	D
1	Известные данные	Значение	Примечания	Обозначение в формуле
2	Сумма кредита	1 200 000,00р.		ПС
3	Годовая процентная ставка	16,00%		
4	Срок	5 лет		
5	Периодичность погашения	4 ежеквартально, т.е. 4 раза в год		
6	Число периодов начисления	20 = Срок*число периодов погашения за весь срок		КПЕР
7	Ставка	4,00% = Годовая процентная ставка/число периодов погашения в году		СТАВКА
8	Периодический платеж	88 256,19р.	Тип=0 (выплата в конце квартала)	
9	Периодический платеж (1)	84 902,82р.	Тип=1 (выплата в начале квартала)	
10	=ПЛТ(В3;В5;В4*В5;В2;-9)			
11	=ПЛТ(В3;В5;В4*В5;В2;1)			
12				
13				

Рис. 6.26. Пример вычисления периодических платежей по кредиту

Для решения данной задачи следует использовать функцию ПЛТ, обязательными аргументами которой являются величины:

- **ставка** — процентная ставка по ссуде, равная годовой процентной ставке, деленной на число периодов платежей в году;
- **кпер** — общее число периодов платежей за весь срок кредита;
- **пс** — общая сумма кредита, которая на текущий момент равноценна ряду будущих платежей, называемая также основной суммой,

а необязательными:

- **бс** — остаток средств после последней выплаты. Если аргумент «бс» опущен, то он полагается равным 0 (нулю);
- **тип** — число, обозначающее, когда должна производиться выплата. Если выплата производится в конце периода, в качестве аргумента «тип» следует указать 0 (нуль) или опустить его. При выплате в начале периода в качестве этого аргумента следует указать 1.

**Задача № 2.** Заемщику известны условия, на которых он может получить кредит. Необходимо оценить максимально возможную сумму кредита, если ограничена величина периодических платежей, которые ему придется выплачивать банку в погашение этого кредита.

Запись условий и решение задачи представлены на рис. 6.27.

**Задача № 3.** Заемщику известны условия, на которых он может получить кредит. Необходимо оценить срок, на который можно взять кредит, если известна его сумма и величина периодических платежей.

Пример записи условий и решения задачи представлен на рис. 6.28.

	A	B	C	D
1	Известные данные	Значение	Примечания	Обозначение в формуле
2	Периодический платеж	-3 000,00р.		ПЛТ
3	Годовая процентная ставка	16,00%		
4	Срок	1 год		
5	Периодичность погашения	12	раз в год	
6	Число периодов погашения	12		КПЕР
7				
8	Возможная сумма кредита	33 505,69р.		ПС
9	=ПС(В3/В5;В4*В5;В2;;1)			

Рис. 6.27. Пример определения возможной суммы кредита при ограниченных выплатах

	A	B	C	D
1	Известные данные	Значение	Примечания	Обозначение в формуле
2	Сумма кредита	150 000,00р.		ПС
3	Периодический платеж	-5 000,00р.		ПЛТ
4	Годовая процентная ставка	16,00%		
5	Периодичность погашения	12	раз в год	
6				
7	Число периодов выплаты	38	месяцев	КПЕР
8	=КПЕР(В4/В5;В3;В2;;1)			
9				

Рис. 6.28. Пример определения сроков расчетов по кредиту

#### 6.4.4.2. Задачи, связанные с вкладом денежных средств

**Задача № 4.** Определить сумму накоплений при данных условиях.

Для решения задачи используется функция БС, аргументы которой определяются в ходе вычислений (рис. 6.29).

**Задача № 5.** Определить срок, в течение которого можно накопить необходимую сумму при данных условиях вклада. Особенность вычислений в данном случае состоит в том, что необходимая сумма накоплений вводится в формулу как аргумент БС (а не ПС, как в предыдущем случае) (рис. 6.30).

**Задача № 6.** Определить величину ежемесячного вклада для накопления необходимой суммы в заданный срок (рис. 6.31).

	А	В	С	Д
1	Известные данные	Значение	Примечания	Обозначение в формуле
2				
3	Периодический платеж	-2 000,00р.		ПЛТ
4	Годовая процентная ставка	10,00%		
5	Периодичность пополнения вклада	12 раз в год		
6	Срок	2 года		
7				
8	Сумма накоплений	52 893,83р.		БС
9	=БС(В4/В5; В6* В5; В3)			

Рис. 6.29. Пример определения суммы накоплений за заданный срок

	А	В	С	Д
1	Известные данные	Значение	Примечания	Обозначение в формуле
2				
3	Периодический платеж	-2 000,00р.		ПЛТ
4	Годовая процентная ставка	10,00%		
5	Периодичность пополнения вклада	12 раз в год		
6	Необходимая сумма накоплений	60 000,00р.		БС
7				
8	Срок, необходимый для накопления	27 месяцев		КПЕР
9	=КПЕР(В4/В5; В3;; В6)			

Рис. 6.30. Пример определения времени накопления заданной суммы

	А	В	С	Д
1	Известные данные	Значение	Примечания	Обозначение в формуле
2	Необходимая сумма накоплений	60 000,00р.		БС
3	Годовая процентная ставка	10,00%		
4	Периодичность пополнения вклада	12 раз в год		
5	Срок накопления	2 года		
6				
7	Периодический платеж	-2 268,70р.		ПЛТ
8	=ПЛТ(В3/В4; В5* В4;; В2)			

Рис. 6.31. Пример определения величины периодических платежей

## 6.5. Анализ данных в ЭТ

Методы научного исследования, анализ и синтез получили широкое распространение и в деловой практике. Анализ предполагает расчленение целого на составные элементы. При этом сами исследуемые предметы подвергаются фактическому расчле-

нению либо изучаемые вопросы подвергаются мысленно расчленению с помощью логической абстракции. Процедуры анализа входят органической составной частью во всякое исследование и обычно образуют его первую стадию, когда исследователь переходит от описания изучаемого объекта к выявлению его строения, состава, а также его свойств, признаков. Синтез же представляет собой сведение в единое целое данных, полученных в результате анализа. В табличном процессоре для анализа табличных данных можно использовать графическое их представление, списки и методы, которым разработчики MS Excel дали название «Что — если?».

### 6.5.1. Анализ данных с помощью диаграмм

#### 6.5.1.1. Типы диаграмм

Зачастую графическое представление данных позволяет наглядно понять смысл явления, которое описывается табличными данными. Одним из способов графического изображения зависимости между величинами является диаграмма. В табличном процессоре MS Excel для представления данных в графической форме можно использовать почти два десятка различных типов диаграмм, причем каждый тип содержит несколько форматов. Для построения диаграмм могут использоваться различные геометрические объекты — линии, плоскости или объемные тела.

Каждый тип диаграмм служит для определенных целей. Обычно пользователь не ограничен жесткими правилами использования какого-либо типа диаграмм для определенных данных. Например, чтобы отобразить итоги продажи по каждому отделу, одинаково хорошо подходят заполненная диаграмма, линейчатая диаграмма, гистограмма или круговая диаграмма. Предположим, что мы располагаем данными об объемах продаж четырьмя региональными торговыми организациями (табл. 6.11).

Таблица 6.11. Объемы продаж по регионам

Регион	Январь	Февраль	Март	Апрель	Май	Июнь
Север	26	20	22	28	24	24
Юг	36	28	34	40	36	30
Восток	31	24	28	34	30	27
Запад	27	27	23	37	30	28



Нетрудно убедиться, что для сравнения динамики изменения объемов продаж лучше использовать графики (рис. 6.32), а для сравнения показателей подразделений друг с другом больше подходит гистограмма (рис. 6.33).

Сравнение этих диаграмм достаточно убедительно показывает, что задача пользователя состоит в том, чтобы выбрать ту диаграмму, которая наилучшим образом отображала бы сущность

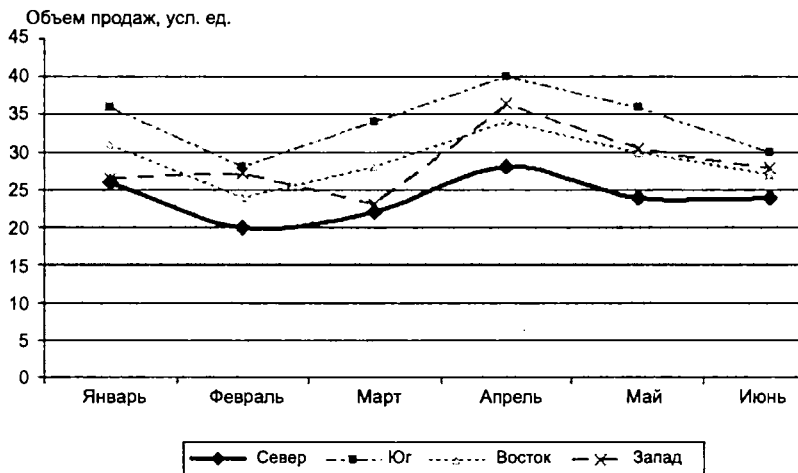


Рис. 6.32. График изменения объемов продаж

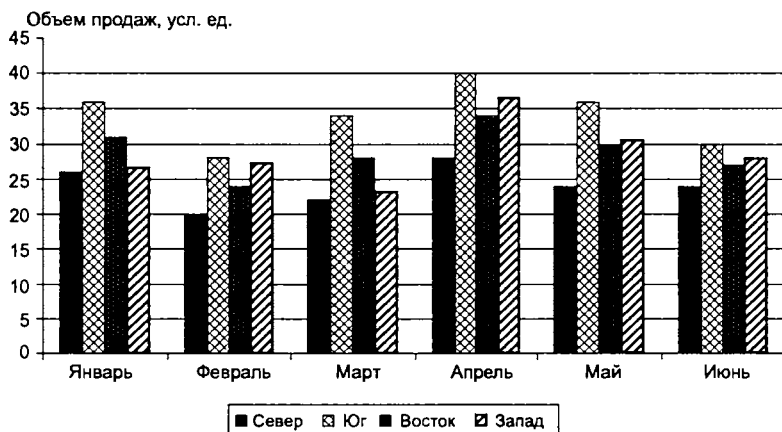


Рис. 6.33. Гистограмма продаж в первом полугодии

того, что следует представить потребителю информации. Поэтому представляется целесообразным рассмотреть краткие характеристики основных типов диаграмм.

График (см. рис. 6.32) представляет собой линию, соединяющую между собой точки, каждая из которых отображает данные, расположенные в одной ячейке. Эта совокупность точек соответствует одному ряду данных. Точки на графике располагаются через равные промежутки. Поэтому график полезен для отображения тенденций изменения данных за равные промежутки времени. Графики отображают данные через некоторые промежутки времени. Данные должны быть последовательными, без пропусков значений. На одном графике можно разместить либо один ряд данных, либо несколько. На представленной диаграмме четко видна взаимосвязь между объемами продаж по каждому региону, но подсчитать итоговый объем продаж по этой диаграмме практически невозможно.

Графики позволяют увидеть и предсказать тенденции развития какого-либо процесса или явления. Иногда важнее видеть тенденцию развития, чем проанализировать реальные данные. (Например, очень важно предугадать уровень инфляции, определить рынок сбыта, прибыль, состояние рынка акций и т. д.) Графики часто используются для отображения информации о деловых и финансовых операциях, таких, как учет прибылей/потерь, производства/продажи, отчеты по заработной плате служащих и др.

Гистограмма (см. рис. 6.33) позволяет представить ряд данных в форме вертикально расположенных столбцов, что позволяет сравнивать, например, изменение данных на протяжении отрезка времени. Диаграммы этого типа удобны также для наглядного сравнения различных величин. Ось категорий В на гистограмме располагается по горизонтали, ось значений — по вертикали. Такое расположение осей подчеркивает характер изменения значений во времени.

Заполненные диаграммы (рис. 6.34) отображают изменения объемов производства или потребления во времени, например количество кубометров газа, израсходованного за год; дневное потребление калорий в течение месяца; изменение стоимости коммунальных услуг за год (газ, электричество, телефон и т. д.). Заполненные диаграммы фактически представляют ту же информацию, что и графики. Но при этом она показывает сумму введенных значений и отображает вклад отдельных рядов в об-

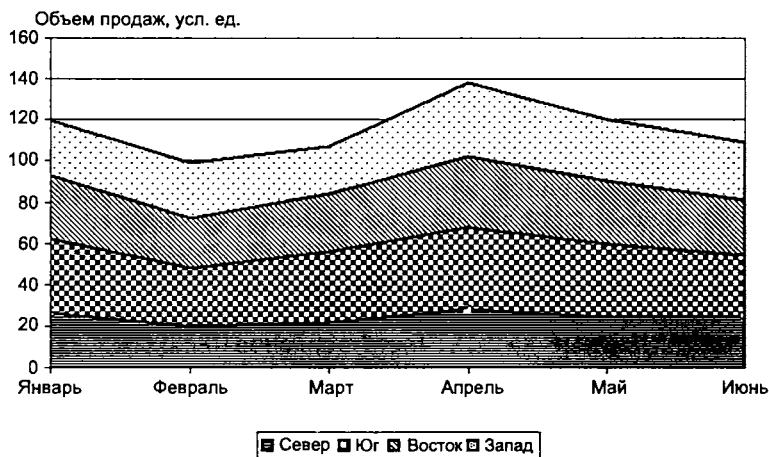


Рис. 6.34. Диаграмма с областями (заполненная диаграмма), построенная на основе данных табл. 6.11

щую сумму. Главное достоинство заполненных диаграмм состоит в том, что они позволяют увидеть соотношение отдельных составляющих и общего объема. На рис. 6.34 верхняя линия представляет общий объем продаж по месяцам. Каждая полоса на этой диаграмме — часть общего объема, которая приходится на один регион.

Эта диаграмма позволяет проследить, например, тенденцию к увеличению продаж в целом по всем четырем регионам. Заполненные диаграммы более наглядны, чем графики, поскольку они в значительной мере отображают картину в целом, а не отдельные изменения. С помощью заполненных диаграмм достигаются две цели: визуально представляются зависимости между сериями данных и указываются их точные значения. Эти диаграммы полезно применять в тех случаях, когда следует обратить внимание получателя информации на потребление или объем продажи, производства и т. п.

**Круговая диаграмма** в отличие от ранее рассмотренных показывает отношение размеров элементов, образующих только один ряд данных, к размеру их суммы. Используя круговую диаграмму, необходимо постоянно помнить, что они всегда отображают только один ряд данных. Такой тип диаграммы целесообразно использовать, когда необходимо подчеркнуть значение того или иного элемента в их совокупности (рис. 6.35).

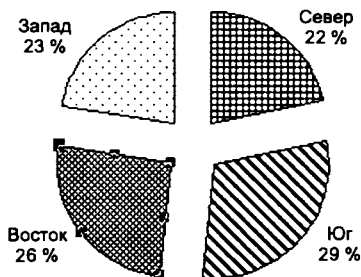


Рис. 6.35. Соотношение объемов продаж в регионах за январь (по данным табл. 6.11)

В том случае, когда необходимо решить аналогичную задачу для отображения нескольких рядов данных, следует использовать кольцевые диаграммы. Здесь каждому ряду данных соответствует одно кольцо (рис. 6.36).

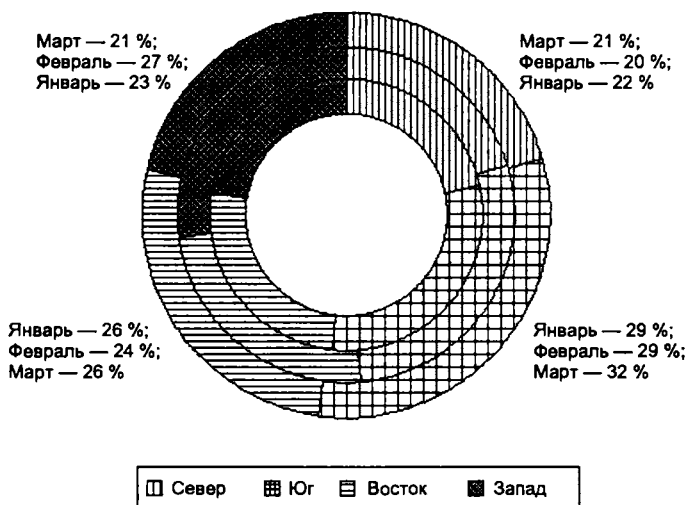


Рис. 6.36. Соотношение объемов продаж в регионах за первый квартал (по данным табл. 6.11)

**Лепестковая диаграмма** позволяет сравнивать совокупные значения нескольких рядов данных (рис. 6.37). В ней каждая категория данных имеет собственную ось координат. Эти оси расходятся лучами от начала координат. Линиями соединяются значения, относящиеся к одному ряду. Диаграммы такого типа по-

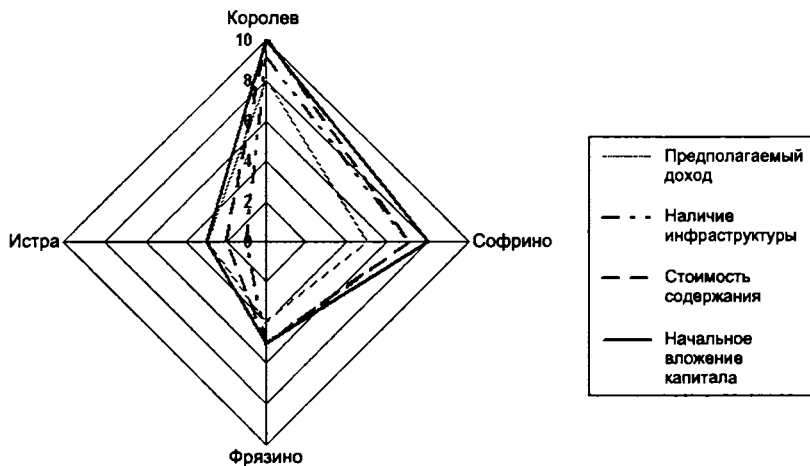


Рис. 6.37. Показатели оценки места возможного расположения филиала

лезны в тех случаях, когда необходимо выбрать один объект из нескольких, сравнивая численные показатели их характеристик. Предположим, предприятие планирует открыть филиал. Предварительная оценка каждого из возможных мест расположения филиала приведена в табл. 6.12.

Таблица 6.12. Оценка места возможного расположения филиала (по десятибалльной шкале)

Оценочный показатель	Королев	Софрино	Фрязино	Истра
Предполагаемый доход	8	5	4	3
Наличие инфраструктуры	9	7	5	1
Стоимость содержания	10	7	5	2
Начальное вложение капитала	10	8	5	3

На основании этих данных будет получена лепестковая диаграмма (рис. 6.37).

**Точечная диаграмма** (рис. 6.38) применяется для отображения отношений между численными значениями в нескольких рядах данных. Этот тип диаграмм особенно полезен, когда нужно сравнить данные, расположенные в разных рядах. Такая необходимость может возникнуть, например, при сравнении расчетных и фактических данных, полученных в ходе эксперимента (табл. 6.13), в котором измерялась температура объекта.

Таблица 6.13. Данные измерений

Время наблюдения	9:00	9:30	10:00	10:30	11:00	11:30	12:00	12:30
$T_{\text{расч}}$	21,00	21,05	21,10	21,15	21,20	21,25	21,30	21,35
$T_{\text{факт}}$	21,62	21,71	21,19	21,33	21,72	21,31	21,35	21,67

Температура объекта

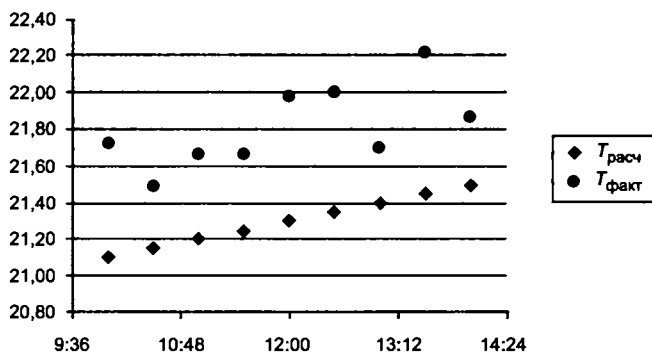


Рис. 6.38. Пример точечной диаграммы

Точечная диаграмма позволяет увидеть разницу между этими двумя рядами данных и оценить, какие обстоятельства вызвали это расхождение, что может оказаться полезным для практической деятельности. Диаграмма, построенная на основе данных табл. 6.13, приведена на рис. 6.38.

Существуют и другие типы диаграмм. Очевидно, что выбор типа диаграммы определяется характером информации, содержащейся в отображаемых данных.

Большинство диаграмм могут быть объемными. В этом случае диаграммы не только выглядят более эффектно, но часто помогают подчеркнуть различия между разными наборами данных. В случае, если диаграмма помещается на тот же рабочий лист, где находятся данные, использовавшиеся для ее построения, она называется внедренной. Внедренные диаграммы полезны для сравнения самих данных и их графического представления. Диаграмма может быть также помещена на отдельном листе диаграмм в рабочей книге, так что рабочий лист и диаграмма будут изолированы друг от друга.

### 6.5.1.2. Объекты диаграмм

Для правильного использования диаграмм при решении практических задач пользователь должен понимать следующее.

1. Данные, на основе которых осуществляется построение диаграммы, представляют собой массив, т. е. именованную последовательность переменных одного типа. Массив может быть одномерным (вектором), двумерным (матрицей), трехмерным. Чаще всего для построения диаграмм используются матрицы.

2. Программа, используемая для построения диаграмм, предполагает, что массив данных образован совокупностью векторов. При этом в зависимости от целей анализа данных в качестве такой совокупности могут рассматриваться строки либо столбцы, а их заголовки — как имена векторов.

Проиллюстрируем это положение следующим примером. В таблице на рис. 6.39 представлен двумерный массив (матрица) данных, содержащих сведения об объемах продаж в трех странах Азиатско-тихоокеанского региона (АТР). Таким образом, с точки зрения программы, обеспечивающей построение диаграммы, объектом анализа данных в этой таблице является объем продаж на протяжении трех лет (2002—2004 гг.) тремя странами АТР. При этом пользователю предоставлена возможность выбрать, какая характеристика этого объекта является более существенной: вклад каждой из стран в этот объем либо его изменение во времени. В первом случае следует считать, что количественные характеристики объекта содержат векторы **R** (строки, содержащие данные по странам), во втором — **C** (столбцы, содержащие данные по годам). Тогда графическая интерпретация этих данных в первом случае будет выглядеть так, как это изображено на рис. 6.40, а во втором — на рис. 6.41.

Продажи в АТР (\$ млн)

	C (столбцы)			
	2002	2003	2004	
R (строки)				
→	Япония	3,7	4,3	4,1
→	Малайзия	2,4	2,7	2,8
→	Н. Зеландия	3,2	3,9	3,8

Рис. 6.39. Пример двумерного массива данных

Объем продаж, млн долл.

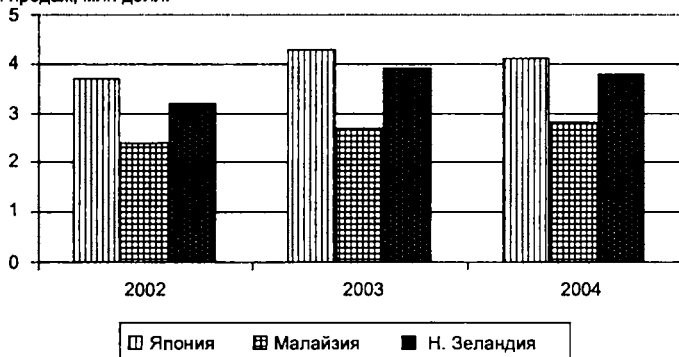


Рис. 6.40. Динамика объемов продаж в странах АТР

Объем продаж, млн долл.

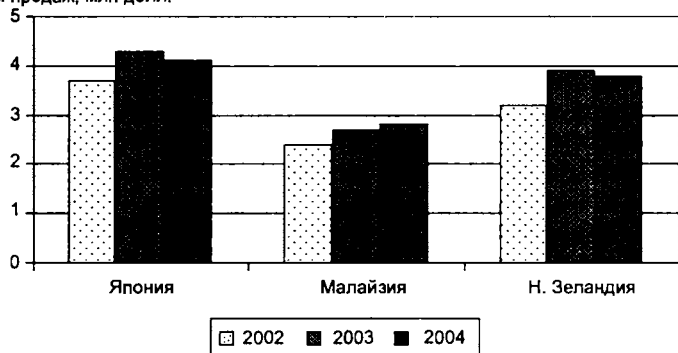


Рис. 6.41. Объемы продаж в странах АТР в 2002—2004 гг.

Эти примеры позволяют понять смысл одного из ключевых терминов, обозначающих основной объект диаграммы, — ряд данных. Ряд данных — это совокупность данных, содержащая количественные характеристики объекта. Эти данные содержатся в одном из векторов (в столбце или в строке), составляющих таблицу. В качестве имен рядов данных Excel использует заголовки столбцов или строк данных. Имена рядов отображаются в легенде диаграммы. Отображением рядов данных на диаграмме являются маркеры данных. Геометрические размеры маркера соответствуют численному значению отображаемых данных. Каждый маркер соответствует одному значению данных вектора, использованного в качестве ряда данных.



С понятием «ряды данных» тесно связано понятие «категории данных», отражающее качество (свойство) элементов в ряду. В качестве имен оси категорий Excel использует заголовки тех столбцов или строк таблицы, которые не используются в качестве рядов данных.

Другими объектами диаграмм являются: легенда, оси и линии координатной сетки, надписи, в число которых входят названия диаграммы и осей, категории, подписи данных и др. Минимальный набор объектов, образующих диаграмму, зависит от ее типа. Как правило, в диаграмму включаются следующие объекты:

- легенда — это текст, идентифицирующий отдельные элементы диаграммы;
- ось — это одна из сторон диаграммы. По горизонтальной оси обычно отображаются категории и/или названия рядов. По вертикальной оси — данные;
- сетка — множество линий, являющихся продолжением деления осей, которые способствуют лучшему восприятию данных на диаграмме и облегчают их анализ. Кроме того, сетка помогает определить точное значение данных.

Объекты можно добавлять в диаграмму и удалять из нее. Каждый из объектов обладает своими свойствами, которые можно изменить путем форматирования. Некоторые объекты диаграммы обозначены на рис. 6.42.

Следует отметить, что иногда значения данных в различных рядах существенно различаются по величине. Например, данные

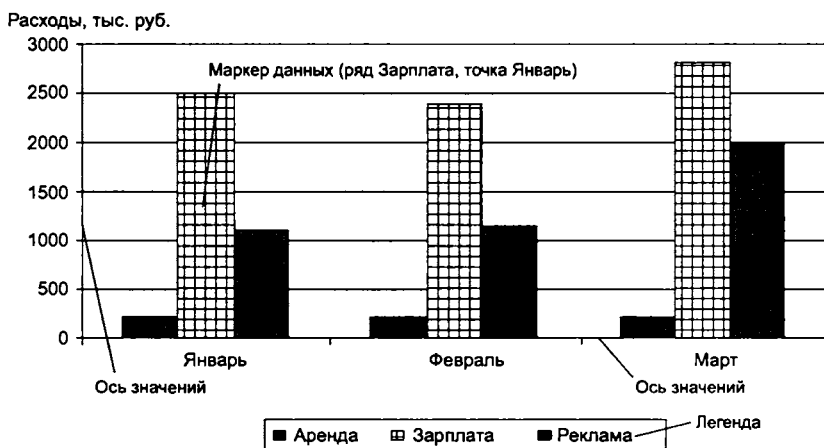


Рис. 6.42. Объекты диаграммы

рядов Зарплата и Аренда на рис. 6.42. В таких случаях для повышения наглядности диаграмм полезно использовать дополнительную ось значений, как это демонстрируется на рис. 6.43.

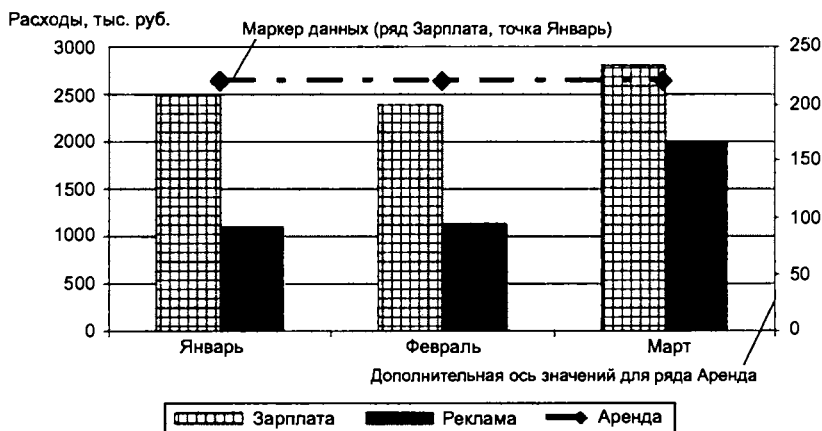


Рис. 6.43. Пример диаграммы с использованием дополнительной оси значений

Приемы работы с объектами диаграмм не отличаются от тех, которые используются во всех программах, входящих в пакет Microsoft Office. Все они сводятся к тому, что объект необходимо выделить, а затем удалять или изменять свойства с помощью команды **Формат/Объект**.

Для того чтобы придать диаграмме в целом и объектам в ее составе необходимые свойства, целесообразно для ее создания использовать **Мастер диаграмм**. Для этого следует выбрать команду **Вставка/Диаграмма...**, после чего появляется диалоговое окно (рис. 6.44).

На этом этапе пользователь имеет возможность выбрать тип диаграммы, в наибольшей степени соответствующий целям анализа. Следующий этап заключается в выборе или уточнении диапазона данных, используемых для построения диаграммы. Для этого используется вкладка **Диапазон данных** (рис. 6.45). Кроме того, с помощью команд, доступных при переходе к вкладке **Ряд**, можно добавить или удалить ряды данных (рис. 6.46).

Затем, на третьем шаге, пользователь может выбрать дополнительные объекты для включения в состав диаграммы и задать некоторые их характеристики и содержание надписей (рис. 6.47).

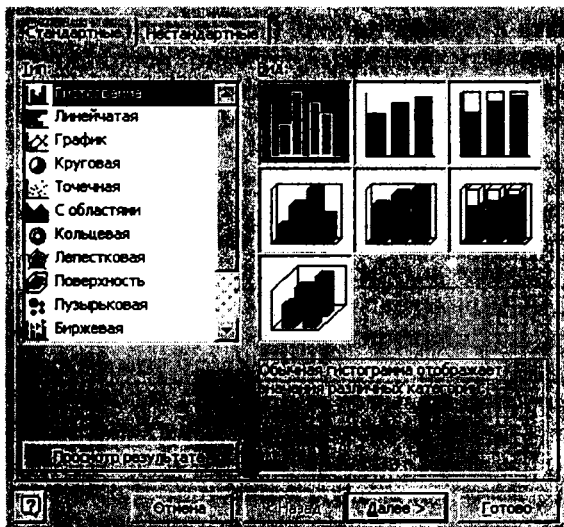


Рис. 6.44. Окно Мастера диаграмм (1-й шаг)

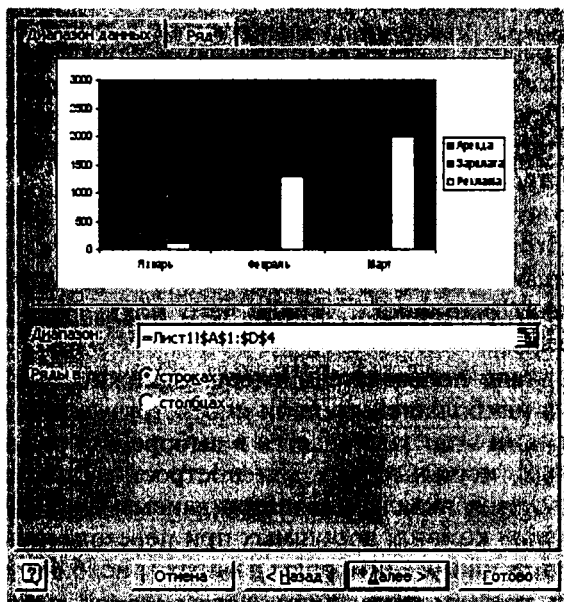


Рис. 6.45. Окно Мастера диаграмм (2-й шаг, вкладка Диапазон данных)

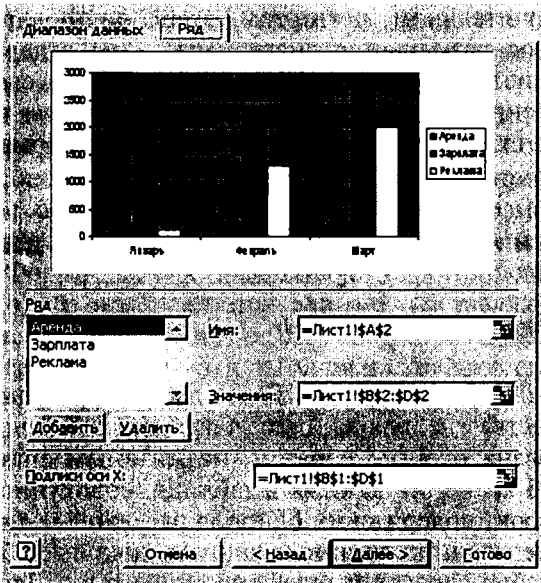


Рис. 6.46. Окно Мастера диаграмм (2-й шаг, вкладка Ряд)

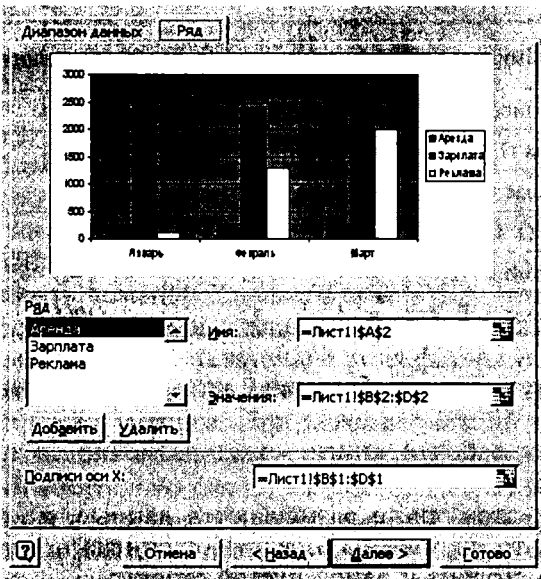


Рис. 6.47. Окно Мастера диаграмм (3-й шаг)

На заключительном, четвертом, этапе пользователь должен решить вопрос размещения диаграммы — на отдельном листе или на том листе, где расположены данные. Если свойства объектов, включенных в диаграмму, которая построена таким образом, не устраивают пользователя, то ее следует форматировать.

## 6.6. Списки и их использование для анализа данных

### 6.6.1. Списки в ТП MS Excel

Табличный процессор MS Excel дает в руки пользователя мощные средства анализа данных, если таблица, где они хранятся, организована особым образом, называемым списком. Список — способ хранения данных в таблице, совокупность поименованных строк, содержащих однородные данные (набор строк таблицы, содержащий связанные данные, например база данных счетов или набор адресов и телефонов клиентов). Данные, представленные в табл. 6.14, соответствуют этому определению.

Таблица 6.14. Пример списка

Номер отдела	Фамилия	Пол	Принадлежность к штату	Год рождения	Оклад, руб.
2	Кузьминов М. Ю.	м	Ш	1948	2325,0
1	Иванов П. П.	м	С	1952	3487,5
1	Решетова О. И.	ж	Ш	1959	2418,0
2	Белов О. И.	м	Ш	1963	2681,5
3	Куликова Р. А.	ж	Ш	1949	2852,0
4	Кравцова З. В.	ж	С	1966	3720,0

Существует ряд требований, которым должны отвечать списки:

- на листе рабочей книги может размещаться только один список;
- если на этом листе размещаются данные, не входящие в список, то их должны отделять от списка не менее одного пустого столбца и одной пустой строки;
- первая строка списка должна содержать заголовки столбцов;

- оформление заголовков столбцов должно отличаться от данных, для чего используется их выделение с помощью шрифта, выравнивания, форматов и рамок;
- во всех ячейках столбца размещаются однотипные данные, при этом используется один формат;
- перед содержимым ячейки не должно быть пробелов.

Одной из особенностей списков является возможность добавления записей в список, их удаления или изменения с помощью форм. Форма данных представляет собой диалоговое окно, в полях которого отображаются данные, составляющие запись (строку списка) (рис. 6.48). Для создания полей в форме Microsoft Excel использует эти заголовки столбцов списка.

Номер отдела:	2	1 из 6
Фамилия:	Кузьминов I	Добавить
Пол:	и	Удалить
Принадлежн. к штату:	Ш	Вернуть
Год рожд.:	1948	Назад
Оклад (руб):	2325	Далее
		Критерии
		Закрыть

Рис. 6.48. Пример формы данных

Для создания формы необходимо установить курсор внутри списка и выбрать команду **Данные/Форма...** Для поиска записи, подлежащей удалению или изменению, следует нажать кнопку **Критерии** и ввести в соответствующие поля формы условия поиска. Затем с помощью кнопок **Далее** и **Назад** найти записи, соответствующие этим условиям. При задании условий поиска можно использовать символы подстановки, приведенные в табл. 6.15.

Следует отметить, что для поиска необходимых данных можно так же, как и в текстовом процессоре MS Word, использовать команду **Правка/Найти**. В этом случае для организации поиска используется диалоговое окно **Найти и заменить** (рис. 6.49). При этом необходимо указать место расположения искомых данных

Таблица 6.15. Символы подстановки

Символы подстановки	Действие символа
? (знак вопроса)	Заменяет один любой знак. Пример: условию «бар?и» соответствуют результаты «барин», «баран» и «барон»
* (звездочка)	Заменяет любое количество любых символов. Пример: условию «*-восток» соответствуют результаты «северо-восток» и «юго-восток»
~ (тильда), за которой следует ?, * или ~	Заменяет вопросительный знак, звездочку или тильду. Пример: условию «ан91~?» соответствует результат «ан91?»

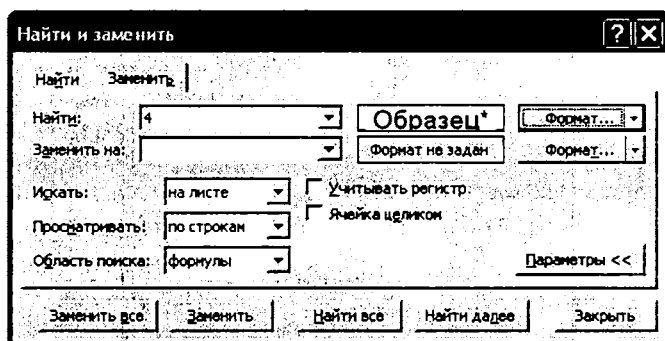


Рис. 6.49. Диалоговое окно Найти и заменить

(в книге или на листе), направление просмотра (по строкам или столбцам) и область поиска (значения, формулы или примечания). Дополнительно можно указать формат как тех данных, которые подлежат замене, так и тех, которые должны быть вновь введены в ячейку.

Существенную помощь при поиске необходимых данных в списках может оказать сортировка. Известно, что сортировка представляет собой способ упорядочения данных. Для сортировки данных в списках следует установить курсор внутри списка и выбрать команду **Данные/Сортировка...** После этого в диалоговом окне **Сортировка диапазона** необходимо задать параметры сортировки: столбец, данные которого должны стать основой для сортировки записей, и порядок сортировки (по возрастанию или убыванию) (рис. 6.50).

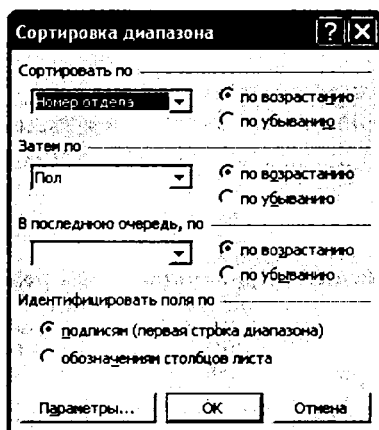


Рис. 6.50. Диалоговое окно Сортировка диапазона

Табличный процессор MS Excel позволяет производить сортировку по нескольким показателям (до трех). Очевидно, что в первую очередь сортировка производится по той категории, которая включает в себя наибольшее число записей, так как последующая сортировка осуществляется уже внутри нее. Для упорядочения данных в ячейках по значениям (без учета формата) в Microsoft Excel предусмотрен определенный порядок сортировки — по возрастанию или по убыванию. Причем этот порядок зависит от формата данных. При сортировке по возрастанию Excel использует следующий порядок:

- числа сортируются от наименьшего отрицательного до наибольшего положительного числа;
- при сортировке алфавитно-цифрового текста Excel сравнивает значения посимвольно слева направо. Например, если ячейка содержит текст "A100", Excel поместит ее после ячейки, содержащей запись "A1", и перед ячейкой, содержащей запись "A11";
- текст, в том числе содержащий числа, сортируется в следующем порядке: 0 1 2 3 4 5 6 7 8 9 (пробел) ! " # \$ % & ( ) \* , . / : ; ? @ [ \ ] ^ \_ ` { | } ~ + < = > A B C D E F G H I J K L M N O P Q R S T U V W X Y Z А Б В Г Д Е Ё Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я, апострофы (') и дефисы (-) игнорируются с единственным исключением: если две строки текста одинаковы, не считая дефиса, текст с дефисом ставится в конец;



- логическое значение **ЛОЖЬ** ставится перед значением **ИСТИНА**;
- все значения ошибки равны;
- пустые значения всегда ставятся в конец.

При сортировке по убыванию этот порядок заменяется на обратный.

### 6.6.2. Способы анализа данных в списках

К числу основных способов анализа данных в списках можно отнести следующие:

- фильтрация данных;
- получение промежуточных итогов;
- консолидация данных;
- сводные таблицы.

#### 6.6.2.1. Фильтрация данных

Фильтрация — это способ поиска подмножества данных в списке. Сущность этого способа заключается в том, что для данных одного или нескольких столбцов задаются условия отбора, после чего в отфильтрованном списке отображаются только те строки, которые содержат данные, отвечающие этим условиям. В табличном процессоре MS Excel используется два способа фильтрации списков: **Автофильтр** — для простых условий отбора и **Расширенный фильтр** — для более сложных условий. В отличие от сортировки при фильтрации порядок записей в списке не изменяется. При фильтрации временно скрываются строки, которые не требуется отображать. Строки, отобранные при фильтрации в Microsoft Excel, можно редактировать, форматировать и выводить на печать, а также создавать на их основе диаграммы, не изменяя порядок строк и не перемещая их.

Чтобы отфильтровать список с помощью **Автофильтра**, необходимо выделить одну из ячеек списка и выбрать команду **Данные/Фильтр/Автофильтр**. После выполнения этой команды в нижнем правом углу ячеек с заголовками столбцов появится черный треугольник, обращенный вершиной вниз, означающий появление в этой ячейке элемента управления «Поле со списком». Список в этом поле содержит условия отбора **Автофильтра**, указанные в табл. 6.16.

Таблица 6.16. Возможности Автофильтра

Условия отбора	Объект поиска
Все	Все строки списка
Первые 10	Заданное число строк с максимальными или минимальными значениями ячеек текущего столбца (например, можно отобразить 10 % строк, имеющих максимальные значения суммы покупки)
Условие	Строки, удовлетворяющие двум условиям или одному условию с оператором сравнения, отличным от И (оператор по умолчанию)
Пустые	Все строки, имеющие пустые ячейки в текущем столбце
Непустые	Все строки, имеющие непустые ячейки в текущем столбце

Чтобы отфильтровать список по двум значениям или более, встречающимся в столбце, или с использованием операторов сравнения, следует из развернувшегося набора значений выбрать строку «Условие...». Эта процедура применяется для фильтрации списка с наложением одного или двух условий отбора значений ячеек отдельного столбца. Для того чтобы наложить одно условие отбора, надо выбрать из разворачивающихся наборов в полях первого условия необходимые оператор сравнения и значение сравнения. Для задания второго условия следует установить переключатель в положение **И** или **ИЛИ** и выбрать из разворачивающихся наборов в полях второго условия необходимые оператор и значение сравнения.

В расширенном фильтре условия отбора вводятся в диапазон условий на листе книги. Команда **Расширенный фильтр** применяется, чтобы отфильтровать данные в тех случаях, когда для отбора записей требуется записать:

- условия в два или более столбца;
- более двух условий в одном столбце;
- условие, которое использует значение, вычисляемое формулой.

Таким образом, чтобы отфильтровать список с помощью расширенного фильтра, следует рядом со списком создать диапазон условий. Для этого в пустую строку создаваемого диапазона нужно ввести или скопировать заголовки фильтруемых столбцов, а в нижележащие строки ввести условия отбора. Диапазон условий и фильтруемый список должны быть разделены, по крайней мере, одним пустым столбцом или строкой. Затем для

запуска процесса фильтрации необходимо указать ячейку в фильтруемом списке и выбрать команду **Данные/Фильтр/Расширенный фильтр**. В результате появится диалоговое окно **Расширенный фильтр** (рис. 6.51). С помощью переключателя **Обработка**, расположенного в этом окне, пользователь должен указать программе, где следует размещать отфильтрованные записи — на месте или в другом диапазоне. Чтобы поместить отфильтрованные записи за пределами существующего списка, следует установить переключатель **Обработка** в положение **Скопировать результат в другое место**, а в поле **Поместить результат в диапазон** указать верхнюю левую ячейку области вставки. Затем необходимо ввести в поле **Диапазон критериев** ссылку на диапазон условий отбора, включая заголовки.

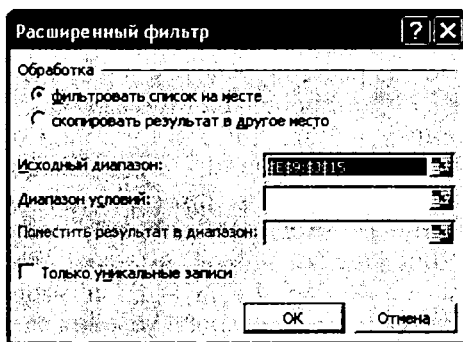


Рис. 6.51. Диалоговое окно **Расширенный фильтр**

Далее приведены примеры использования расширенного фильтра для отбора записей из списка. Очевидно, что в более простых случаях следует использовать автофильтр. В качестве исходного списка использован пример, приведенный в табл. 6.14.

**Пример 7.** В условии отбора содержатся ограничения по данным, записанным в два столбца и более.

Предположим, из списка сотрудников необходимо выбрать женщин-совместителей. В этом случае условия отбора должны быть записаны в диапазон условий следующим образом:

Номер отдела	Фамилия	Пол	Принадлежность к штату	Год рождения	Оклад, руб.
		ж	С		

**Пример 8.** В условии отбора содержится более одного ограничения по данным, записанным в одном столбце.

Из списка сотрудников необходимо выбрать мужчин в возрасте от 55 до 60 лет, учитывая, что задача решается в 2005 году. Для решения этой задачи условия в диапазон условий можно записать:

Номер отдела	Фамилия	Пол	Год рождения	Год рождения	Оклад, руб.
		м	>1945	<1950	

Особенность оформления диапазона условий заключается в том, что заголовок столбца «Год рожд.» включен в него дважды.

**Пример 9.** В условии отбора записей используется значение, вычисляемое формулой.

Из списка сотрудников необходимо выбрать тех, чей оклад ниже среднего по учреждению. Данная задача может быть решена, если условия отбора записать так:

Номер отдела	Фамилия	Пол	Принадлежность к штату	Год рождения	Оклад, руб.
					=F2<CP3HA4(\$F\$2:\$F\$6)

В данном примере следует обратить внимание на несколько обстоятельств:

- первое — столбец диапазона условий, в который должно быть записано ограничение, содержащее формулу, не должен иметь заголовок;
- второе — условие записывается как формула, содержащая логическое выражение, в котором значение первой ячейки столбца сравнивается со значением, вычисляемым формулой;
- третье — в формуле, вычисляющей среднее значение, используются абсолютные ссылки.

#### 6.6.2.2. Получение промежуточных итогов

Одной из функций ТП, используемых для анализа данных, является автоматический подсчет значений промежуточных итогов по группам данных, а также общих итогов. При получе-

нии итогов в списке Excel автоматически создает его структуру (рис. 6.52).

		Кв 1	Кв 2	Кв 3	Кв 4	Всего за год
Ручки	Копилярные	114 940 р.	126 076 р.	150 046 р.	141 066 р.	532 128 р.
	Перьевые	96 421 р.	152 211 р.	135 438 р.	163 313 р.	547 383 р.
	Шариковые	199 911 р.	178 082 р.	90 148 р.	169 892 р.	638 033 р.
		412 272 р.	456 369 р.	375 632 р.	474 271 р.	1718 544 р.
Клей	Канцелярский	200 019 р.	128 240 р.	200 375 р.	181 959 р.	710 593 р.
	Карандаш	164 581 р.	125 146 р.	148 991 р.	186 743 р.	625 461 р.
	ГВА	197 588 р.	168 595 р.	207 990 р.	155 414 р.	729 587 р.
		562 188 р.	421 981 р.	557 356 р.	524 116 р.	2065 737 р.
		974 460 р.	878 350 р.	932 988 р.	998 387 р.	3783 185 р.

Рис. 6.52. Пример структуры, созданной при вычислении промежуточных итогов

В рассматриваемом случае под структурой следует понимать объединение в группы строк и/или столбцов списка, содержащих однородные данные. Вид структуры можно изменять, скрывая или показывая соответствующие строки и/или столбцы. В приведенном примере такими группами являются **Ручки** и **Клей**.

При вычислении промежуточных итогов могут быть использованы функции, список которых приведен в табл. 6.17.

Таблица 6.17. Функции для вычисления промежуточных итогов

Функция	Результат
Сумма	Сумма чисел. Эта операция используется по умолчанию для подведения итогов по числовым полям
Количество	Количество данных. Эта операция используется по умолчанию для подведения итогов по нечисловым полям. Операция «Кол-во значений» работает так же, как и функция СЧЁТЗ
Среднее	Среднее чисел
Максимум	Максимальное число
Минимум	Минимальное число
Произведение	Произведение чисел
Количество чисел	Количество данных, являющихся числами. Операция «Кол-во чисел» работает так же, как и функция СЧЁТ

В качестве аргументов этих функций используются данные, включенные в соответствующую группу. Группы создаются пользователем перед подведением итогов путем сортировки данных списка по тем категориям, которые анализируются пользователем. Когда табличный процессор MS Excel подсчитывает промежуточные итоги, он вычисляет каждое итоговое значение, суммируя отдельные ячейки в выделенном диапазоне. Когда вычисляются общие итоги, Excel делает вычисления по отдельным ячейкам, а не на основании сумм промежуточных итогов. Эти нюансы несущественны при использовании функций **СУММ** и **СРЗНАЧ**, но это принципиально при использовании функций **МИН** и **МАКС**.

Итак, в случае необходимости вычисления промежуточных итогов следует сначала произвести сортировку данных таким образом, чтобы записи образовали однородные группы. Затем следует поместить курсор внутри отсортированного списка и выбрать команду **Данные/Итоги...**. В результате выполнения этой команды появится диалоговое окно **Промежуточные итоги**, с помощью которого нужно уточнить дальнейшие действия программы. Для этого требуется в поле **При каждом изменении в:** указать название того столбца списка, который содержит созданные при сортировке группы, в поле **Операция** — функцию, обрабатывающую данные внутри группы, и в поле **Добавить итоги по:** отметить те столбцы, данные которых обрабатываются с помощью выбранной функции. Пример диалогового окна для списка табл. 6.14 приведен ниже (рис. 6.53).

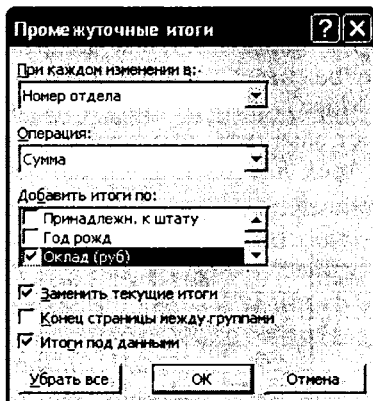


Рис. 6.53. Диалоговое окно **Промежуточные итоги**

В заключение следует отметить, что более совершенным способом анализа данных являются сводные таблицы, которые способны дать пользователю те же сведения, что и вычисление промежуточных итогов.

### 6.6.2.3. Консолидация данных

Консолидация данных — это способ получения итоговой информации, при котором данные, расположенные в нескольких различных областях, объединяются в соответствии с выбранной функцией обработки. При этом данные, расположенные в одной или нескольких исходных областях, обрабатываются и отображаются в одной итоговой таблице. Источники данных могут находиться на том же листе, что и итоговая таблица, на других листах той же книги или в других книгах. При консолидации для обработки данных используются те же функции, что и при подведении итогов. Место расположения исходных (консолидируемых) данных называется областью-источником. Та часть рабочего листа, где будут размещаться итоговые (консолидированные) данные — областью назначения.

Предусмотрены два способа консолидации данных: по расположению и по категориям.

**Консолидация по расположению.** Консолидация по расположению используется, если данные исходных областей расположены в одном и том же порядке и имеют одни и те же заголовки. Используется этот способ для консолидации данных листов, например бюджетов отделов предприятия, созданных на основе единого шаблона. Рассмотрим пример учета метизов, поступавших в три разных магазина в первом квартале (рис. 6.54). Для сравнения учетные данные, расположенные на разных листах (Январь, Февраль, Март) рабочей книги, на рисунке изображены рядом, что позволяет заметить, что взаимное расположение данных на каждом из трех листов полностью совпадает и по строкам, и по столбцам.

	Январь				Февраль				Март		
	Mar1	Mar2	Mar3		Mar1	Mar2	Mar3		Mar1	Mar2	Mar3
болты	12	14		болты	8		21	болты	21	8	
гайки		22	44	гайки	17	52		гайки	41		
шайбы	18	45		шайбы	4	11	5	шайбы		51	
шурупы		78	22	шурупы	74	14		шурупы	22		11

Рис. 6.54. Данные учета

Для консолидации по расположению пользователю следует заранее создать область назначения, где будет строка с заголовками столбцов (**Mar1**, **Mar2** и **Mar3**) и строка с названиями строк (болты, гайки, шайбы и шурупы). Разумеется, расположение этих заголовков и названий должно соответствовать их расположению в исходных таблицах. После этого необходимо выделить курсором верхнюю левую ячейку, находящуюся на пересечении столбца **Mar1** и строки **Болты**, и дать команду **Данные/Консолидация...**. Затем в диалоговом окне **Консолидация**, вызванном этой командой, следует уточнить функцию, с помощью которой должны обрабатываться консолидируемые данные, и ввести в поле **Список диапазонов** ссылки на области-источники. Для этого ссылку на диапазон-источник надо поочередно ввести в поле **Ссылка** и нажать кнопку **Добавить** (рис. 6.55). Консолидация завершается нажатием кнопки **ОК** после того, как будет заполнен список диапазонов.

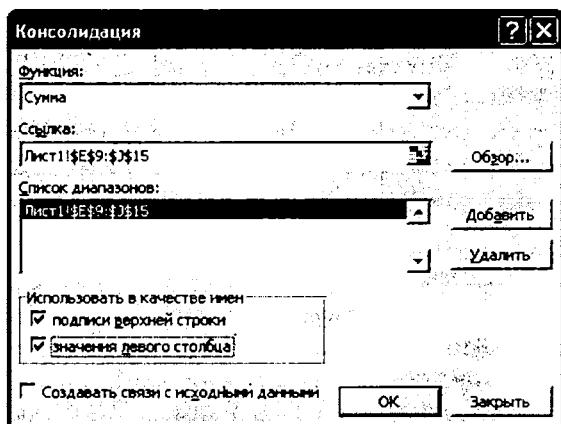


Рис. 6.55. Диалоговое окно **Консолидация**

Нетрудно заметить, что в случае консолидации по расположению табличный процессор MS Excel осуществляет обработку элементов массивов данных в соответствии с их индексами, соответствующими положению того или иного элемента в массиве. Очевидно, что размерность всех обрабатываемых массивов должна быть одинаковой.

**Консолидация по категориям.** Консолидация по категориям применяется, если данные исходных областей не упорядочены,



но имеют одни и те же заголовки. Используется этот способ для консолидации данных списков, имеющих разную структуру, но одинаковые заголовки. В этом случае табличный процессор MS Excel в процессе обработки данных присваивает им имена, используя при этом заголовки столбцов и названия строк. При этом способе консолидации первый выделенный столбец определяет поле группировки данных. Поясним это примером. Пусть на два склада (№ 1 и № 2), принадлежащих одной фирме, в разное время поступали товары от разных поставщиков (рис. 6.56). Используя эти данные как области источника, несложно получить различные варианты данных в области назначения.

Склад 1				Склад 2			
Наименование товара	Поставщик	Дата поставки	Стоимость (тыс. руб)	Наименование товара	Поставщик	Дата поставки	Стоимость (тыс. руб)
тюнер	ООО "Симфония"	мар	116	усилитель	ООО "Симфония"	мар	841
усилитель	ЗАО "Триумф"	янв	485	тюнер	ЗАО "Триумф"	апр	119
тюнер	ООО "Симфония"	фев	76,5	тюнер	ОАО "Альфа"	май	182
блок питания	ОАО "Альфа"	мар	208	усилитель	ООО "Симфония"	июн	537
усилитель	ЗАО "Триумф"	апр	580	блок питания	ОАО "Альфа"	мар	416
блок питания	ЗАО "Триумф"	май	683	блок питания	ОАО "Альфа"	апр	593
тюнер	ОАО "Альфа"	июн	118	усилитель	ЗАО "Триумф"	мар	759
блок питания	ОАО "Альфа"	янв	192,7				
усилитель	ОАО "Альфа"	фев	515				
усилитель	ОАО "Альфа"	апр	460				

Рис. 6.56. Сведения о поступлении товаров

**Пример 10.** Предположим, что нам необходимо определить суммарную стоимость каждого из наименований всех товаров, поступивших в течение полугодия на оба склада от всех поставщиков. Для решения этой задачи следует группировать данные по столбцу **Наименование товара**. В этом случае пользователь должен сообщить программе необходимые уточнения, указав их в диалоговом окне **Консолидация**. Это окно приведено на рис. 6.57, а результаты консолидации — на рис. 6.58. Нетрудно заметить, что в этом случае программе необходимо указать, какие элементы оформления списка следует использовать в качестве имен данных, так как в этом случае имена используются в качестве аргументов формул. Следует обратить внимание на то,

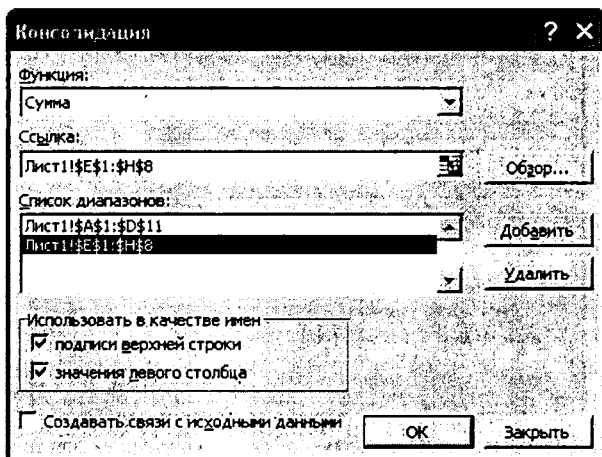


Рис. 6.57. Диалоговое окно с параметрами консолидации по категориям

	Дата	Стоимость
Поставщик	поставки	(тыс. руб)
тюнер		611,5
усилитель		4177
блок питания		2092,7

Рис. 6.58. Результаты консолидации для примера 10

что диапазоны, включенные в список консолидации, начинаются со столбца **Наименование товара**.

**Пример 11.** Необходимо оценить суммарную стоимость каждого наименования товара, поступившего на оба склада от разных поставщиков. В этом случае группировать данные следует по столбцу **Поставщик**, т. е. включать в диапазон консолидации данные, начиная со второго столбца — **Поставщик**. Следует помнить, что перед включением в диапазон консолидации новых данных из него необходимо удалить ссылки, использовавшиеся ранее. Для этого требуется выделить ссылку на диапазон в поле **Список диапазонов** и нажать кнопку **Удалить**. Диалоговое окно с указанием сведений, необходимых для консолидации в этом случае, приведено на рис. 6.59.

Результаты консолидации данных в этом случае можно увидеть на рис. 6.60.

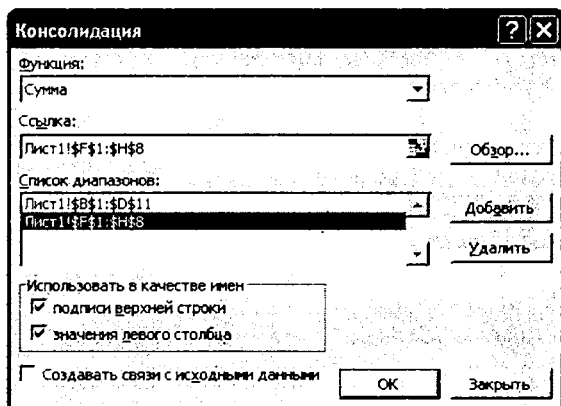


Рис. 6.59. Диалоговое окно с параметрами консолидации для примера 11

	Дата поставки	Стоимость (тыс. руб)
000 "Симфония"		1570,5
ЗАО "Триумф"		2626
ОАО "Альфа"		2684,7

Рис. 6.60. Результаты консолидации для примера 11

**Пример 12.** Для обоснования финансового плана на первое полугодие необходимо дать оценку распределению объемов поставок по месяцам. При этом нет необходимости учитывать вид товаров и реквизиты поставщика. Чтобы ответить на этот во-

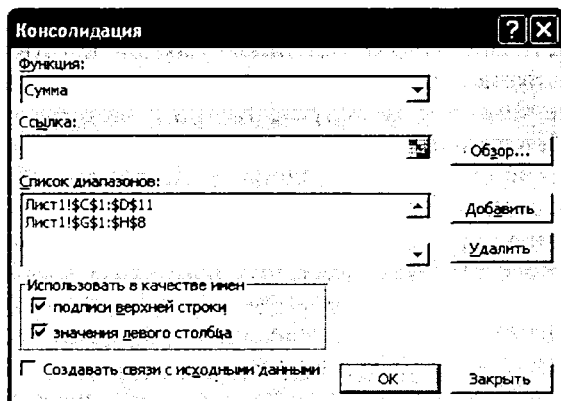


Рис. 6.61. Диалоговое окно с параметрами консолидации для примера 12

	Стоимость (тыс. руб)
мар	2340
янв	677,7
фев	591,5
апр	1 752
май	865
июн	655

Рис. 6.62. Результаты консолидации для примера 12

прос, для группировки данных нужно выбрать столбец **Дата поставки**, т. е. включать в диапазон консолидации данные, содержащиеся только в двух правых столбцах. Диалоговое окно для решения данной задачи приведено на рис. 6.61.

Результаты консолидации представлены на рис. 6.62.

#### 6.6.2.4. Сводные таблицы

Рассмотренные выше методы анализа данных трудно использовать, если список содержит большой объем данных. Например, в тех случаях, когда необходимо установить взаимосвязь между данными, расположенными в удаленных друг от друга участках рабочего листа. В этом случае табличный процессор MS Excel предоставляет пользователю возможность создать сводные таблицы. Сводная таблица — это интерактивная таблица, создаваемая пользователем на основе данных, содержащихся в списке, и отражающая обобщенные результаты обработки этих данных. Сводные таблицы используются для целенаправленного обобщения информации, представления взаимосвязей между категориями, изменения способа группировки данных, суммирования различных категорий данных и др.

Работа со сводными таблицами требует правильного понимания терминов, используемых создателями табличного процессора MS Excel при их описании (табл. 6.18). Значение некоторых терминов поясняется иллюстрацией (рис. 6.63).

Сводные таблицы создаются с помощью **Мастера сводных таблиц** (пошаговой инструкции по созданию сводных таблиц) по команде **Данные/Сводная таблица...**. Первый шаг заключается в том, что пользователь должен указать источник данных для сводной таблицы и вид создаваемого отчета. В качестве источника данных можно использовать списки MS Excel, внешние ис-

Таблица 6.18. Термины сводной таблицы [1]

Термин	Определение
Ось	Размерность сводной таблицы: столбец, строка, страница
Поле столбцов	Совокупность данных, размещаемая в столбцах сводной таблицы
Область данных	Часть сводной таблицы, которая содержит итоговые данные
Источник данных	Список (или таблица), используемый для создания сводной таблицы. Источник данных может быть внутренним (расположенным в Excel) и внешним (например, таблица базы данных Access)
Поле	Категория информации. В списке столбец является полем
Заголовок поля	Заголовок столбца, обычно содержащийся в первой строке списка
Общий итог	Строка или столбец, который создает сводная таблица для суммирования строк, столбцов или того и другого вместе
Группа	Совокупность данных, рассматриваемая как единое подмножество
Элемент	Имена, размещенные в заголовках столбцов или названиях строк
Поля страниц	Один из признаков, используемых для отбора данных. Страницы создают трехмерное представление данных
Вращение	Реорганизация сводной таблицы с помощью вращения ее относительно различных осей
Обновить	Пересчитать сводную таблицу (изменить ее содержимое) после модификации источника данных
Поле строк	Совокупность данных, размещаемая в строках сводной таблицы
Промежуточные итоги	Строки или столбцы, созданные с помощью сводной таблицы для суммирования итогов по каждой группе
Функции для вычисления промежуточных итогов	Функции, используемые для обработки значений, отображаемых в области данных. По умолчанию Excel использует функции СУММА для чисел и КОЛИЧЕСТВО ЗНАЧЕНИЙ для текста. Кроме них можно использовать функции: СРЕДНЕЕ, МАКСИМУМ, МИНИМУМ, ПРОИЗВЕДЕНИЕ, КОЛИЧЕСТВО ЧИСЕЛ и др.

точники данных (например, таблицы MS Access), диапазоны консолидации. Отчет может быть представлен в форме сводной таблицы или сводной диаграммы. Затем необходимо указать

Южный		Кв.2	Кв.3	Общий итог
Сумма заказов				
Хлебобулочные продукты	Дроздов	16 378 898	19 820 399	36 199 297
	Краснов	17 488 590	5 828 800	23 317 390
	Морозов	13 013 780	8 158 848	21 172 628
Мясопродукты	Всего	35 880 270	30 804 948	66 685 218
Морепродукты	Дроздов	30 759 780	39 248 800	69 998 580
	Краснов	14 379 180	10 053 000	24 432 180
	Морозов	28 027 580	12 318 080	40 345 660
Морепродукты	Всего	73 166 540	61 609 880	134 776 420
Общий итог		108 740 810	92 414 520	199 155 330

Рис. 6.63. Объекты сводной таблицы

программе диапазон, где размещены исходные данные. Если перед началом создания сводной таблицы курсор находился внутри списка, программа способна самостоятельно распознать этот диапазон. В последнюю очередь пользователь должен указать место, где должна размещаться создаваемая сводная таблица. В результате программа предоставляет в распоряжение пользователя макет и список полей сводной таблицы (рис. 6.64). Далее, в

	B	C	D	E	F	G	H	I	J	K	
1	Перетащите сюда поля страниц										
2											
3	Перетащите сюда поля столбцов										
4	Перетащите сюда поля строк	Перетащите сюда элементы данных								Перетащите элементы в сводную таблицу	
5				<input type="checkbox"/> Регистрационный номер дела				<input type="checkbox"/> Дата возбуждения уголовного дела			
6				<input type="checkbox"/> Дата закрытия дела				<input type="checkbox"/> Причина закрытия			
7				<input type="checkbox"/> Статья				<input type="checkbox"/> Тип правонарушения			
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											
21											
22											

Рис. 6.64. Макет и список полей сводной таблицы

зависимости от целей анализа пользователь должен указать место расположения на макете полей сводной таблицы.

Рассмотрим конкретный пример анализа с помощью сводных таблиц. Предположим, что имеется база данных учета уголовных дел, фрагмент которой представлен на рис. 6.65.

Регистр. номер дела	Дата возбужд-я уголовн. дела	Дата закр-я дела	Причина закрытия	Статья	Тип правонаруш-я
324	01.01.2001	21.01.02	Передано в суд	УКРФ-10	Мелкое хулиганство
329	02.01.2001	22.01.02	Передано в суд	УКРФ-25	Разбой
334	03.01.2001	23.01.02	Передано в суд	УКРФ-66	Вымогательство
339	04.01.2001	24.01.02	Передано в суд	УКРФ-10	Мелкое хулиганство
344	05.01.2001	25.01.02	Передано в суд	УКРФ-25	Разбой
349	06.01.2001	26.01.02	Передано в суд	УКРФ-89	Незаконный сбыт наркотиков
354	07.02.2001	27.01.02	Передано в суд	УКРФ-65	Кража

Рис. 6.65. Фрагмент базы данных учета уголовных дел

**Пример 13.** Оценить состояние раскрываемости преступлений, т. е. рассмотреть распределение количества дел по причинам закрытия. В данном случае анализ осуществляется только по одному признаку, поэтому достаточно в область полей строк поместить поле **Причина закрытия**, а в область элементов данных — поле **Регистрационный номер дела** и задать функцию **Количество по полю Регистрационный номер дела** (см. рис. 6.64). Результат представлен на рис. 6.66.

Количество по полю Регистрационный номер дела	
Причина закрытия	Итог
Закрью за недостаточностью улик	46
Отозвано	26
Передано в суд	306
(пусто)	141
Общий итог	519

Рис. 6.66. Состояние раскрываемости преступлений

**Пример 14.** Провести анализ раскрываемости преступлений в зависимости от типа правонарушения. Данная задача предполагает группировку данных уже по двум признакам — по причинам закрытия и по типам правонарушений. Поэтому необходимо в область полей строк макета сводной таблицы поместить поле **Причина закрытия**, в область полей столбцов — поле **Тип правонарушения**, а в область элементов данных — поле **Регистрационный номер дела** и задать функцию **Количество по полю Регистрационный номер дела**. В этом случае сводная таблица примет вид, представленный на рис. 6.67.

Количество по полю Регистрационный номер дела	Тип правонарушения													
Причина закрытия	Вымогательство	Заказное убийство	Коррупция	Кража	Мелкое хулиганство	Незаконное хранение оружия	Незаконный сбыт наркотиков	Разбой	Тяжкие телесные повреждения	Халатность	Хищение в особо крупных размерах	Хранение наркотиков	Общий итог	
Закрыто за недостатком доказательств	5	25			5						11		46	
Отозвано	2								24				26	
Передано в суд	15		5	43	90	28	23	12	18	35	5	32	306	
(пусто)	12	6	13	19	5	4	5	19	50	6	2		141	
Общий итог	34	6	43	62	100	32	28	31	92	41	18	32	519	

Рис. 6.67. Распределение преступлений по типам и причинам закрытия

## 6.7. Решение задач путем подбора параметра

Возможности табличного процессора MS Excel по осуществлению анализа данных дополняют методы, которым разработчики MS Excel дали название «Что — если?». Они предлагают несколько возможных способов решения и прогнозирования той



или иной задачи. Каждый из этих методов имеет определенное назначение, а их использование зависит от характера решаемой задачи. С их помощью можно ответить на вопросы:

- Какой результат принесет повышение разницы между себестоимостью и отпускной ценой?
- Какие изменения претерпят экономические показатели при росте объема реализации продукции на 10 %?

Методы «Что — если?» позволяют найти ответ на все вопросы, подобные этим. При обычной установке пакета программ MS Office пользователь получает возможность использовать некоторые инструменты анализа, принадлежащие к этой группе:

- подбор параметра позволяет найти значение, которое требуется ввести в формулу, описывающую зависимость функционально связанных между собой величин, для получения нужного результата;
- таблицы подстановок дают возможность вычисления, просмотра и сравнения результатов, полученных при различных вариантах значений аргумента (одного или двух) на одном листе;
- поиск решения осуществляет вычисление оптимального значения целевой функции.

Рассмотрим более подробно сущность этих методов.

В практической деятельности специалистам приходится решать как прямые, так и обратные задачи. Прямые задачи отвечают на вопрос, к какому результату приведут действия, предпринятые в определенных условиях? Например, вычислить значения функции  $f(x)$  для  $N$  значений аргумента  $x$ , принадлежащих отрезку  $[A, B]$ .

Обратные задачи позволяют получить ответ на вопрос, что надо сделать для того, чтобы получить нужный результат? К числу таких задач относятся задачи нахождения корней уравнения, решение систем уравнений и т. п.

**Подбор параметра** позволяет решить задачу проще. Это средство применимо только в одной конкретной ситуации: когда известен требуемый ответ, формула для вычисления этого ответа и существует только одно входное значение для получения этого ответа. В диалоговом окне **Подбор параметра** показаны три необходимых компонента: **Установить в ячейке**, **Значение** и **Изменяя значение ячейки**. Таким образом, в этом диалоговом окне необходимо указать только одну установленную и одну изме-

няемую ячейки. Причем диапазон ячеек в этом случае использовать нельзя.

Работа этого средства основана на оценке наибольшего числа в изменяющейся ячейке и сравнении с ответом в фиксированной ячейке. После чего этот ответ сравнивается с содержимым поля **Значение**. Затем происходит оценка наименьшего числа и соответствующее сравнение. Поочередно выбирая большие и малые числа, средство **Подбор параметра** пытается подобрать значение в изменяющейся ячейке, необходимое для получения ответа, заданного в фиксированной ячейке. Иногда это средство не может найти ответ — неверны, например, входные данные или ответа вообще не существует. В этом случае появляется сообщение об ошибке. По умолчанию средство **Подбор параметра** производит до 100 итераций (попыток найти ответ). Поиск прекращается, когда найден ответ с точностью до 0,001. Если ответ не найден по той причине, что число, предложенное средством **Подбор параметра**, отличалось от ответа больше, чем позволяет точность, определенная по умолчанию, а при более внимательном рассмотрении установлено, что такое число существует, то можно изменить число итераций или точность вычислений с помощью команды **Сервис/Параметры**. В появившемся диалоговом окне **Параметры** выбрать вкладку **Вычисления**, а затем увеличить число итераций (опция **Предельное число итераций**) или уменьшить относительную погрешность (опция **Относительная погрешность**). Данный инструмент может быть использован при обязательном выполнении двух условий:

- в установленной ячейке должно быть число;
- значения, находящиеся в установленной и изменяемой ячейках, должны быть связаны между собой функциональной зависимостью.

Рассмотрим способы использования этого инструмента для решения конкретных задач.

### 6.7.1. Вычисление корней уравнения

Известно, что уравнение описывает зависимость одной переменной (функции) от второй, независимой, называемой аргументом. Корнем уравнения называется такое значение аргумента, при котором функция равняется 0. На графике функции кор-

ням уравнения соответствуют точки пересечения функции с осью абсцисс.

Для нахождения всех корней уравнения путем подбора параметра целесообразно выполнить следующие действия:

- создать таблицу значений функции  $f(x)$  для  $N$  значений аргумента  $x$  на отрезке, содержащем корни уравнения;
- выбрать в этой таблице ячейку, содержащую значение, наиболее близкое к точке перемены знака (от положительных значений к отрицательным или наоборот);
- выбрать команду **Сервис/Подбор параметра** и в появившемся диалоговом окне ввести в поле **Установить в ячейке** адрес выбранной ячейки, в поле **Значение** — интересующий нас результат (в данном случае это значение равно 0), а в поле **Изменяя значение ячейки** — адрес ячейки, где записано значение аргумента, соответствующее выбранной зависимой ячейке.

Предположим, что нам необходимо найти корни квадратного уравнения  $y = 2,34x^2 - 3,12x + 1$ .

Таблица значений и график этой функции в диапазоне, содержащем оба корня, представлены на рис. 6.68.

Диалоговое окно со сведениями, введенными в его поля, а также результат подбора параметра представлены на рис. 6.69.

Завершается процесс нажатием кнопки **ОК** в окне **Результат подбора параметра**, после чего в таблицу значений функции будет внесено значение корня уравнения. Для нахождения остальных

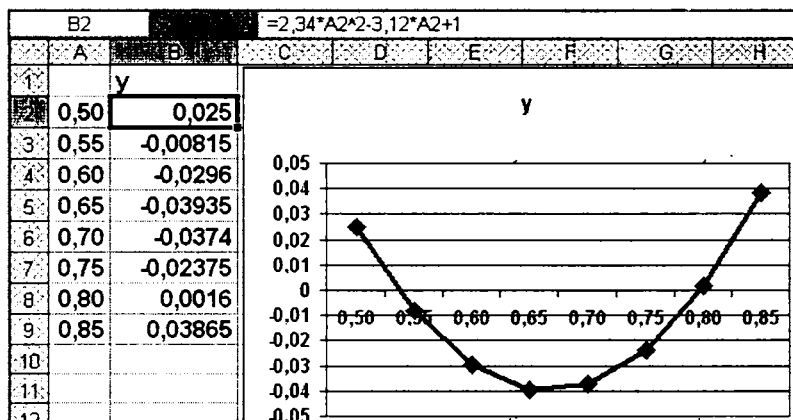


Рис. 6.68. Таблица значений и график функции

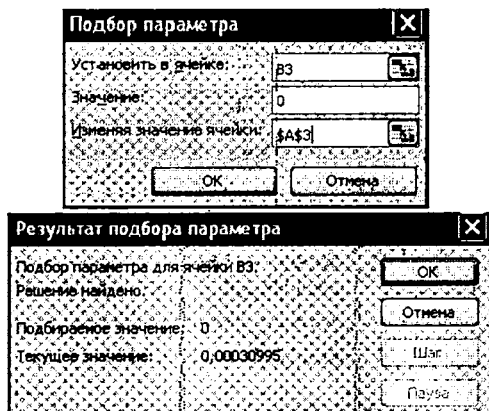


Рис. 6.69. Диалоговое окно Подбор параметра

ных корней необходимо повторить эти действия во всех ячейках, вблизи которых функция меняет свой знак.

### 6.7.2. Решение систем уравнений

Сущность решения системы уравнений состоит в нахождении координат точки, общей для функций, составляющих данную систему. Представляется целесообразным для нахождения решения системы уравнений путем подбора параметра использовать ту же последовательность действий, что и в предыдущем случае. Однако, учитывая, что в искомой точке значения функций равны между собой, при создании таблицы функций оказывается полезным ввести в нее еще одну вспомогательную функцию, которая представляет собой разность функций, составляющих систему. В этом случае решение системы уравнений будет являться корнем уравнения вспомогательной функции.

Предположим, что нам необходимо найти точку равновесия спроса и предложения на рынке в случае, если эти характеристики описываются системой уравнений:

$$\begin{cases} y = \ln(x); \\ z = -2x + 4, \end{cases}$$

где  $x$  — стоимость товара;  $y$  — функция, описывающая предложение;  $z$  — функция, описывающая спрос.

Таблица значений и график функций спроса, предложения и вспомогательной представлены на рис. 6.70.

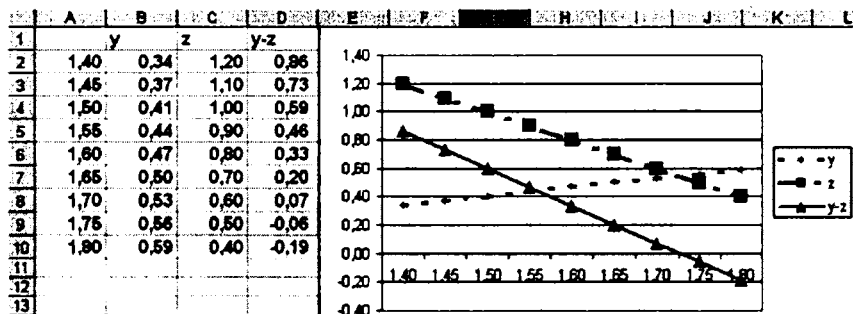
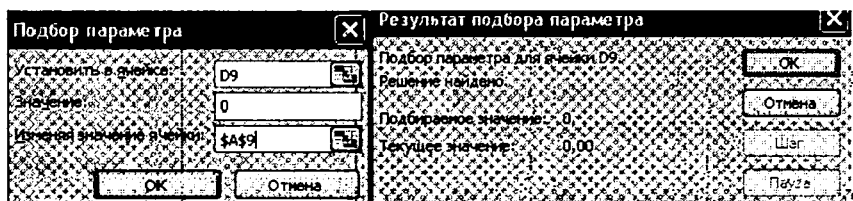


Рис. 6.70. Таблица функций и их графики

Рис. 6.71. Решение задачи нахождения точки равновесия  
спрос — предложение

Диалоговое окно со сведениями, введенными в его поля, а также результат подбора параметра представлены на рис. 6.71.

### 6.7.3. Решение финансовых задач

Применение инструмента **Подбор параметра** может оказаться весьма полезным для финансового анализа. Рассмотрим ситуацию, когда предприятию необходимо получить кредит для закупки оборудования. Чтобы оценить условия получения кредита, составим таблицу платежей (рис. 6.72). С этой целью введем в нее стоимость единицы оборудования, на приобретение которого предназначен кредит (ячейка E3) и количество этого оборудования (E4). Произведение этих величин даст значение требуемой суммы (E5 и B3). Исходя из этого значения, с помощью функции периодического платежа (ПЛТ) можно решить прямую задачу оценки условий кредита. При таком оформлении расчетов величин ПЛТ можно рассматривать как функцию аргументов, записанных в ячейках E3, E4, B4 и B5.

	A	B	C	D	E
1:	Расчет платежей				
2:					
3:	Сумма кредита	\$35 000,00	Цена за ед.	\$3 500,00	
4:	Ставка	12%	Кол-во, шт	10	
5:	Срок, мес.	24	Сумма	\$35 000,00	
6:	Периодич. платеж	-\$1 647,57			

Рис. 6.72. Таблица расчета платежей по кредиту

Если размер периодического платежа не устраивает получателя кредита, а кредитное учреждение отказывается снижать ставку, то перед его клиентом возникает проблема выбора варианта последующих действий. Действительно, для уменьшения платежей можно:

1) снизить размер кредита за счет уменьшения количества закупаемого оборудования (вариант не требует согласования с кредитором);

2) продлить срок выплаты кредиторской задолженности;

3) добиться снижения кредитной ставки.

Очевидно, что получить конкретные показатели, необходимые для обоснования любого из этих вариантов, можно с помощью подбора параметра. Рассмотрим вариант, когда получатель кредита сможет выплачивать по кредиту ежемесячно не более 1000 долл., тогда ему необходимо уменьшить количество закупаемого оборудования, чтобы снизить размер кредита. Диалоговое окно со сведениями, введенными в его поля, а также результат подбора параметра для первого варианта решения представлены на рис. 6.73.

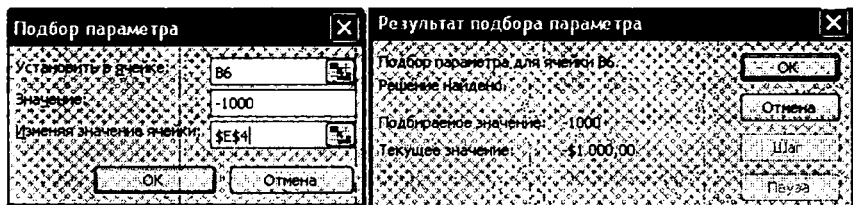


Рис. 6.73. Решение задачи оценки условий получения кредита

Этот анализ позволяет сделать вывод, что при наличии возможности выплачивать ежемесячно \$1000 на условиях, предлагаемых кредитором, можно взять кредит 21 000 долл., что позволит приобрести шесть единиц оборудования.

## 6.8. Использование таблиц подстановки для анализа данных

Таблицы подстановки являются частью блока задач, который иногда называют инструментами анализа «Что — если?». Они представляют собой диапазон ячеек, показывающий, как изменение определенных значений в формулах влияет на результаты этих формул. Таблицы предоставляют способ быстрого вычисления нескольких версий в рамках одной операции, а также способ просмотра и сравнения результатов всех различных вариантов на одном листе. Данный инструмент позволяет проанализировать зависимость конечного результата от двух параметров одновременно.

### 6.8.1. Таблицы подстановки с одной переменной

Таблицы подстановки с одной переменной используются для анализа влияния одного из аргументов на значение функции. Например, известно, что чем выше процентная ставка, предлагаемая банком, тем больше риск клиента потерять свои сбережения. Оценить размер возможного выигрыша при повышении степени риска возможно с помощью данного инструмента.

Для использования таблиц подстановки прежде всего необходимо выполнить соответствующие вычисления. Предположим, что необходимо оценить возможный выигрыш от вклада средств в течение года при ежемесячном платеже в 1000 руб. С этой целью определяем размер возможных накоплений, задаваясь усредненным значением процентной ставки. Определяющее значение для использования таблиц подстановки имеет оформление этих расчетов. Важно, чтобы каждый из аргументов формулы располагался в отдельной ячейке. Пример правильного оформления расчета приведен на рис. 6.74. Здесь в ячейках **A2:A6** размещены имена аргументов, используемых для вычисления суммы накоплений, а в диапазоне ячеек **B2:B6** — их значения.

Затем необходимо создать макет таблицы подстановки. Он представляет собой диапазон ячеек **D3:D7**, в который введены возможные значения процентной ставки, и ячейку, расположенную на одну строку выше и на один столбец правее этого диапа-

	A	B	C	D	E
1					
2	Ставка (годовых)	12%			12 682,50р.
3	Срок (месяцев)	12			
4	Периодичность платежа (раз в год)	12			
5	Периодический платеж	-1 000,00р.			
6	Сумма накоплений	12 682,50р.			

Рис. 6.74. Оформление расчетов и макет таблицы подстановки

зона (E2). В эту ячейку необходимо ввести формулу, содержащую ссылку на итоговую функцию. В нашем случае эта формула принимает вид: =B6.

Далее для создания таблицы подстановки необходимо выделить диапазон ячеек D2:E7, содержащий макет, и выбрать команду **Данные/Таблицы подстановки...**. Затем следует ввести в появившееся диалоговое окно необходимые сведения (рис. 6.75). Поскольку результаты расчета должны располагаться в тех же строках, где и возможные значения процентной ставки, то нужная ссылка должна быть помещена в поле **Подставлять значения по строкам в:**

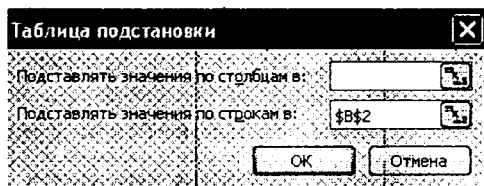


Рис. 6.75. Диалоговое окно Таблица подстановки

После нажатия кнопки **OK** программа выдаст результаты расчета (рис. 6.76).

	A	B	C	D	E
1					
2	Ставка (годовых)	12%			12 682,50р.
3	Срок (месяцев)	12	8,00%		12 449,93р.
4	Периодичность платежа (раз в год)	12	10,00%		12 565,57р.
5	Периодический платеж	-1 000,00р.	14,00%		12 800,75р.
6	Сумма накоплений	12 682,50р.	16,00%		12 920,31р.
7			18,00%		13 041,21р.

Рис. 6.76. Заполненная таблица подстановки



### 6.8.2. Таблицы подстановки с двумя переменными

Таблица подстановки с двумя переменными предоставляет пользователю возможность оценить влияние двух аргументов на значение, принимаемое функцией. Например, оценить возможности по накоплению средств в зависимости от срока вклада и величины процентной ставки.

Как и в предыдущем случае, сначала необходимо выполнить первоначальные расчеты. Для упрощения возьмем в качестве примера уже использованный выше расчет суммы накоплений. Макет таблицы подстановки с двумя переменными включает еще один диапазон, в который вводятся значения второй переменной (рис. 6.77). При этом ячейка с формулой, содержащей ссылку на итоговую функцию, должна располагаться в левом верхнем углу таблицы.

Очевидно, что в этом случае для заполнения таблицы необходимо ввести данные в оба поля диалогового окна **Таблица подстановки** (рис. 6.78).

	срок вклада (мес)					
12 682,50р.	3	6	9	12	15	18
8,00%						
10,00%						
14,00%						
16,00%						
18,00%						

Рис. 6.77. Макет таблицы подстановки с двумя переменными

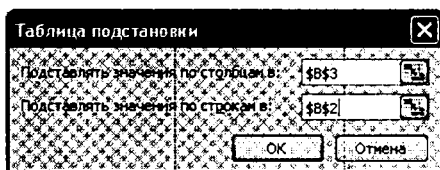


Рис. 6.78. Диалоговое окно **Таблица подстановки** с двумя переменными

	срок вклада (мес)					
12 682,50р.	3	6	9	12	15	18
8,00%	3 020,04р.	6 100,89р.	9 243,77р.	12 449,93р.	15 720,63р.	19 057,19р.
10,00%	3 025,07р.	6 126,40р.	9 305,91р.	12 565,57р.	15 907,40р.	19 333,48р.
14,00%	3 035,14р.	6 177,75р.	9 431,64р.	12 800,75р.	16 289,15р.	19 901,09р.
16,00%	3 040,18р.	6 203,59р.	9 495,24р.	12 920,31р.	16 484,22р.	20 192,60р.
18,00%	3 045,22р.	6 229,55р.	9 559,33р.	13 041,21р.	16 682,14р.	20 489,38р.

Рис. 6.79. Таблица подстановки с двумя переменными

В результате этих действий мы получим таблицу с двумя входами — срок вклада и процентная ставка, данные которой можно использовать для принятия обоснованного решения (рис. 6.79).

## 6.9. Решение оптимизационных задач

### 6.9.1. Сущность задач оптимизации

Табличный процессор MS Excel предоставляет пользователю еще один тип задач, получивших название оптимизационных. Решение этих задач позволяет обосновать выбор такого варианта действий, который по тем или иным соображениям предпочтительнее других. В теории принятия решений всякое мероприятие (совокупность действий), имеющее единый замысел и направленное к достижению определенной цели, называется операцией [3]. Результат операции зависит от выбора некоторых параметров, определяющих условия ее осуществления. Выбор этих параметров называется решением. Для сравнения решений нужно иметь какой-то количественный показатель. В теории принятия решений он называется показателем эффективности, в математике — целевой функцией. Этот показатель выбирается таким образом, чтобы он отражал целевую направленность действий. Более предпочтительным считается то решение, которое в наибольшей степени способствует достижению цели предстоящих (планируемых) действий. В практике наиболее часто имеют место случаи, когда целевая функция должна получить максимальное или, наоборот, минимальное значение. Например, желательно, чтобы прибыль от операции достигала максимума, а затраты — минимума.

Решения, приводящие к получению результата, который по тем или иным показателям предпочтительнее других, называются оптимальными. Все параметры, от которых зависит результат операции, можно разделить на две группы:

- заданные, заранее известные параметры — условия осуществления операции;
- параметры, определяемые при принятии решения — элементы решения.

Очевидно, что условия осуществления операции накладывают ограничения на решение.

Поэтому в общем виде задача оптимизации может быть сформулирована следующим образом: при заданных условиях найти такое решение, которое позволяет получить экстремальное значение показателя эффективности.

В состав табличного процессора Excel включены несколько рабочих листов, демонстрирующих работу средства **Поиск решения**. Они содержатся в файле `C:\Msoffice\EXCEL\EXAMPLES\SOLVER\SOLVSAMP.XLS` и показывают способы решения следующих задач.

**Структура продукции.** Разработка плана максимизации дохода предприятия при ограниченных ресурсах сырья и техники для производства продукции.

**График работы.** Минимизация затрат на заработную плату служащих за счет оптимизации графика работы (из расчета 40 часов в неделю работы каждого).

**Задача транспортировки.** Минимизация стоимости перевозок продукции от производителя к потребителю при заданных сроках доставки.

**Максимизация прибыли.** Максимизация прибыли при вкладе денежных средств в депозитный сертификат.

**Портфель ценных бумаг.** Максимизация прибыли при данном уровне риска за счет оптимального состава ценных бумаг.

Использование метода **Поиск решения** основано на том, что в рабочем листе MS Excel создается ячейка назначения. В нее помещается формула, содержащая целевую функцию. Эта ячейка называется также целевой ячейкой. Целевая функция в ячейке назначения записывается таким образом, что ее значение зависит от значений, находящихся в изменяемых ячейках. В изменяемые ячейки должны быть записаны элементы решения, которые также называются переменными решения. Средство **Поиск решения** в процессе вычисления изменяет значения в этой группе ячеек, что приводит к изменению значения в ячейке назначения. Эти изменения осуществляются до тех пор, пока значение в целевой ячейке не станет соответствовать заданному или пока программа не определит, что ответа не существует. Кроме того, необходимо записать в виде формул условия осуществления операции, которые называются ограничениями. Ограничения — это зависимости, определяющие диапазон значе-

ний данных, находящихся в целевой ячейке и/или изменяемых ячейках.

Ниже на конкретных примерах подробно рассматривается методика использования инструмента **Поиск решения**.

### 6.9.2. Задача о планировании производства

Предприятие выпускает продукцию трех видов: телевизоры (Т), стереосистемы (С) и акустические системы (А), используя общий склад комплектующих. На изготовление изделий идет пять видов комплектующих: шасси (ш), кинескопы (к), динамики (д), блоки питания (б) и электронные платы (э). Количество комплектующих, необходимое для изготовления каждого вида продукции, приведено в табл. 6.19.

Таблица 6.19. Необходимые объемы комплектующих по видам продукции

Комплектующие	Количество комплектующих на единицу продукции		
	Т	С	А
1. Шасси (ш)	1	1	0
2. Кинескопы (к)	1	0	0
3. Динамики (д)	2	2	1
4. Блоки питания (б)	1	1	0
5. Электронные платы (э)	2	1	1

Запас комплектующих на складе составляет:  $N_{шс} = 450$ ,  $N_{кс} = 250$ ,  $N_{дс} = 800$ ,  $N_{бс} = 450$ ,  $N_{эс} = 600$ .

При реализации одно изделие приносит предприятию прибыль: Т — 75, С — 50, А — 35 денежных единиц.

Требуется спланировать производство, чтобы обеспечить максимальную прибыль при условии ограниченности запаса комплектующих.

**Математическая модель.** Обозначим через  $x_t$ ,  $x_c$ ,  $x_a$  количество единиц продукции Т, С и А, производство которых планируется.

Целевая функция — прибыль, приносимая производством ( $x_t$ ,  $x_c$ ,  $x_a$ ), будет равна

$$L = 75x_t + 50x_c + 35x_a. \quad (6.14)$$

**Ограничения.** Обязательность обеспечения ассортимента записывается в виде трех ограничений-неравенств:

$$x_T \geq 1, x_C \geq 1, x_A \geq 1. \quad (6.15)$$

Кроме того, ограничения накладываются запасами сырья: количество израсходованных комплектующих не должно превышать их числа на складе. Это условие можно записать:

$$\begin{cases} N_{\text{ши}} = 1x_T + 1x_C \leq 450; \\ N_{\text{ки}} = 1x_T \leq 250; \\ N_{\text{ди}} = 2x_T + 2x_C + 1x_A \leq 800; \\ N_{\text{би}} = 1x_T + 1x_C \leq 450; \\ N_{\text{эи}} = 2x_T + 1x_C + 1x_A \leq 600. \end{cases} \quad (6.16)$$

Это линейные неравенства. Таким образом, необходимо найти такие неотрицательные значения переменных  $x_1, x_2, x_3$ , чтобы они удовлетворяли ограничениям — неравенствам (6.14)—(6.16), максимизируя линейную функцию этих переменных:

$$L = \sum_{i=1}^3 c_i x_i \Rightarrow \max.$$

С точки зрения теории принятия решений в данном примере операция состоит в производстве определенного количества продукции:

- цель операции — получение максимальной прибыли;
- переменные  $x_1, x_2, x_3$  представляют собой элементы решения;
- ограничения описывают условия осуществления операции.

В математике такие задачи решаются методами линейного программирования, а в табличном процессоре MS Excel — с помощью инструмента **Поиск решения**. Использование этого метода предполагает выполнение некоторых действий, которые будут описаны далее.

1. Ввод исходных данных и формул для ограничений и целевой функции. При решении таких задач с помощью табличного процессора существенное значение приобретает рациональное размещение исходных данных. Это означает, что данные на рабочем листе должны располагаться не произвольным образом, а в определенном порядке. Один из вариантов такой организации расположения исходных данных для решения нашей задачи представлен на рис. 6.80.

	A	B	C	D	E	F	
1				T	C	A	
2		Количество $x_i \rightarrow$					
3	Компл-щие	Склад	Использ.				
4	Шасси	450		1	1	0	
5	Кинескоп	250		1	0	0	
6	Динамик	800		2	2	1	
7	Блок пит.	450		1	1	0	
8	Элек. плата	600		2	1	1	
9	Прибыль от реализации одного изделия:						
10				75р.	50р.	35р.	
11			Всего	160р.			
12	Прибыль от реализации продукции						

Рис. 6.80. Исходные данные для решения задачи планирования производства

При таком расположении данных в таблице различные ее части имеют вполне определенное предназначение. Так, в диапазоне D3:F3 размещаются данные о количестве произведенной продукции ( $x_t$  в ячейке D3,  $x_c$  — в E3,  $x_a$  — F3);

Очевидно, что решать эту задачу необходимо с учетом ограничений, которые обусловлены, с одной стороны, наличием комплектующих на складе, а с другой — необходимостью обеспечения ассортимента выпускаемой продукции. Чтобы учесть первую группу ограничений, в диапазон C4:C8 следует ввести данные об израсходованных комплектующих (табл. 6.20).

Таблица 6.20. Форма записи данных

Ячейка	Формула	
	Математическая запись	Запись в терминах Excel
C4	$N_{\text{ши}} = 1x_t + 1x_c$	=D\$2*D4+\$E\$2*E4+\$F\$2*F4
C5	$N_{\text{ки}} = 1x_t$	=D\$2*D5+\$E\$2*E5+\$F\$2*F5
C6	$N_{\text{ди}} = 2x_t + 2x_c + 1x_a$	=D\$2*D6+\$E\$2*E6+\$F\$2*F6
C7	$N_{\text{би}} = 1x_t + 1x_c$	=D\$2*D7+\$E\$2*E7+\$F\$2*F7
C8	$N_{\text{зи}} = 2x_t + 1x_c + 1x_a$	=D\$2*D8+\$E\$2*E8+\$F\$2*F8

Вторая группа ограничений может быть записана выражением D3:F3>=1.

В ячейку **E12** следует записать целевую функцию (6.14), которая в терминах Excel примет вид:

$$=D10*D2+E10*E2+F10*F2$$

2. Применение инструмента **Поиск решения**. После выполнения этих действий необходимо ввести исходные числа, обозначающие количество произведенной продукции, в ячейки **D3:F3**. Например, по две единицы каждого вида. Затем следует выделить курсором целевую ячейку **E12** и выбрать команду **Сервис/Поиск решения**. В результате ее выполнения появится диалоговое окно, в поля которого **Изменяя ячейки** и **Ограничения** пользователь должен ввести соответствующие ссылки (рис. 6.81).

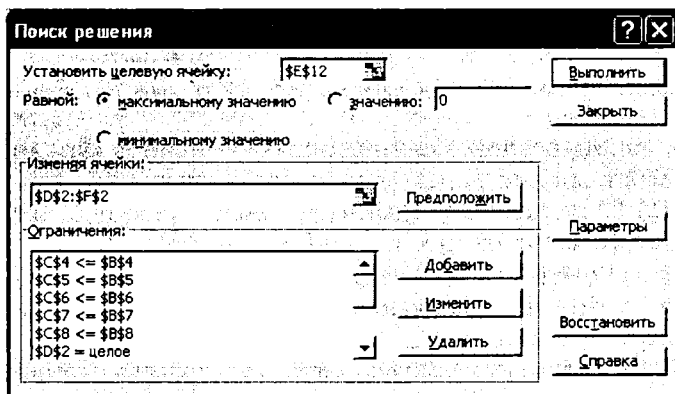


Рис. 6.81. Диалоговое окно **Поиск решения** для задачи планирования производства

После нажатия кнопки **Выполнить** программа решает задачу. Результаты решения будут отображаться в ячейках **E12** (прибыль) и **D2:F2** (элементы решения — оптимальное при данных ограничениях количество единиц произведенной продукции).

### 6.9.3. Транспортная задача

Целью решения транспортной задачи является минимизация стоимости перевозок продукции от производителя к потребителю при заданных сроках доставки.

Предположим, что имеется три предприятия-изготовителя  $A_1$ ,  $A_2$  и  $A_3$ , расположенных в разных географических пунктах,

например в Астрахани, Барнауле, Воронеже. На складах готовой продукции этих предприятий сосредоточены запасы в количестве  $a_a, a_b, a_v$  единиц. Потребители этой продукции  $B_1, B_2, B_3, B_4$  и  $B_5$  расположены в различных регионах, например в Ейске, Ижевске, Курске, Москве и Новгороде. Потребители подали заявки на поставку  $b_e, b_n, b_k, b_m$  и  $b_n$  единиц этой продукции. Сумма всех заявок равна сумме всех запасов:

$$\sum_{i=1}^3 a_i = \sum_{j=1}^5 b_j. \quad (6.17)$$

Известна стоимость перевозки единицы груза от каждого пункта отправления  $A_i$  до каждого пункта назначения  $B_j$ , составляющая  $c_{ij}$  денежных единиц. Эти данные сведены в табл. 6.21.

Таблица 6.21. Исходные данные для решения транспортной задачи

Пункт отправления	Пункт назначения				
	1. Ейск	2. Ижевск	3. Курск	4. Москва	5. Новгород
1. Астрахань	$c_{11} = 10$	$c_{12} = 8$	$c_{13} = 6$	$c_{14} = 5$	$c_{15} = 4$
2. Барнаул	$c_{21} = 6$	$c_{22} = 5$	$c_{23} = 4$	$c_{24} = 3$	$c_{25} = 6$
3. Воронеж	$c_{31} = 3$	$c_{32} = 4$	$c_{33} = 5$	$c_{34} = 5$	$c_{35} = 9$

Требуется составить такой план перевозок, определяющий пункты отправления, назначения и количество перевозимого груза, чтобы удовлетворить все заявки при минимальной суммарной стоимости перевозки всех грузов.

В этой модели представлена задача доставки товаров с трех заводов на пять региональных складов. Товары могут доставляться с любого завода на любой склад, однако, очевидно, что стоимость доставки на большее расстояние будет большей. Требуется определить объемы перевозок между каждым заводом и складом в соответствии с потребностями складов и производственными возможностями заводов, при которых транспортные расходы минимальны

**Математическая модель.** Обозначим  $x_{ij}$  количество единиц продукции, отправляемой из  $i$ -го пункта отправления в  $j$ -й пункт назначения. Эти переменные можно записать в виде матрицы (табл. 6.22).



Таблица 6.22. Матрица перевозимой продукции

Пункт отправления	Пункт назначения				
	1. Ейск	2. Ижевск	3. Курск	4. Москва	5. Новгород
1. Астрахань	$x_{11}$	$x_{12}$	$x_{13}$	$x_{14}$	$x_{15}$
2. Барнаул	$x_{21}$	$x_{22}$	$x_{23}$	$x_{24}$	$x_{25}$
3. Воронеж	$x_{31}$	$x_{32}$	$x_{33}$	$x_{34}$	$x_{35}$

Совокупность чисел ( $x_{ij}$ ) принято называть планом перевозок, а сами величины  $x_{ij}$  — перевозками. Эти переменные должны удовлетворять следующим условиям.

1. Суммарное количество продукции, направляемой из каждого пункта отправления во все пункты назначения, не должно превышать запас продукции в данном пункте. Это условие выражается с помощью условий — неравенств:

$$\begin{cases} x_{11} + x_{12} + x_{13} + x_{14} + x_{15} \leq a_1; \\ x_{21} + x_{22} + x_{23} + x_{24} + x_{25} \leq a_2; \\ x_{31} + x_{32} + x_{33} + x_{34} + x_{35} \leq a_3. \end{cases} \quad (6.18)$$

2. Суммарное количество продукции, доставляемой в каждый пункт назначения, не должно быть меньше указанного в заявке:

$$\begin{cases} x_{11} + x_{21} + x_{31} \geq b_1; \\ x_{12} + x_{22} + x_{32} \geq b_2; \\ x_{13} + x_{23} + x_{33} \geq b_3; \\ x_{14} + x_{24} + x_{34} \geq b_4; \\ x_{15} + x_{25} + x_{35} \geq b_5. \end{cases} \quad (6.19)$$

3. Суммарная стоимость всех перевозок должна быть минимальной:

$$L = \sum_{i=1}^5 \sum_{j=1}^5 c_{ij} x_{ij} \Rightarrow \min, \quad (6.20)$$

где знак двойной суммы означает, что суммирование осуществляется по всем комбинациям индексов  $i$  и  $j$ , т. е. по всем парам пункт отправления — пункт назначения.

Таким образом, решение транспортной задачи состоит в нахождении таких неотрицательных значений переменных  $x_{ij}$ , чтобы они удовлетворяли ограничениям — неравенствам (6.18) и (6.19), минимизируя при этом линейную функцию этих переменных (6.20).

С точки зрения теории принятия решений в данном примере операция состоит в перевозке продукции из пунктов отправления в пункты назначения:

- цель операции — обеспечение минимальной стоимости перевозок;
- переменные  $x_{ij}$  представляют собой элементы решения;
- ограничения описывают условия осуществления операции.

После разработки математической модели появляется возможность создания таблицы исходных данных (рис. 6.82).

	A	B	C	D	E	F	G
1.	Число перевозок от производителя А к потребителю В:						
2. Заводы:	Всего	Ейск	Ижевск	Курск	Москва	Новгород	
3. Астрахань							
4. Барнаул							
5. Воронеж							
6. Итого:							
7.	Потребности складов -->	180	80	200	160	220	
8. Заводы:	Поставки	Затраты на перевозку от завода i к складу j:					
9. Астрахань	310	10	8	6	5	4	
10. Барнаул	260	6	5	4	3	6	
11. Воронеж	280	3	4	5	5	9	
12. Перевозка:							

Рис. 6.82. Запись исходных данных для решения транспортной задачи

1. Ввод исходных данных и формул для ограничений и целевой функции. Как и в предыдущем случае, все части таблицы исходных данных имеют свое предназначение:

- диапазон C9:G11 содержит матрицу стоимости перевозок;
- в ячейки диапазона C7:G7 записаны значения заявок на поставку продукции;
- значения в ячейках диапазона B9:B11 означают наличие продукции в пунктах отправления;
- ячейки диапазона C3:G5 предназначены для размещения элементов решения (матрица  $x_{ij}$ );

- в ячейки диапазона **B3:B5** должны быть введены уравнения ограничений (6.18), а в ячейки **C6:G6** — уравнения ограничений (6.19);
- в ячейки диапазона **C12:G12** записываются формулы для вычисления суммарной стоимости перевозок из всех пунктов отправления в соответствующий пункт назначения;
- ячейка **B12** содержит целевую функцию.

При таком размещении исходных данных математическую модель транспортной задачи можно записать в терминах MS Excel так, как это представлено в табл. 6.23.

Таблица 6.23. Формулы для решения транспортной задачи

Ячейка	Формула	
	Математическая запись	Запись в терминах Excel
B3	$x_{11} + x_{12} + x_{13} + x_{14} + x_{15}$	=СУММ(C3:G3)
B4	$x_{21} + x_{22} + x_{23} + x_{24} + x_{25}$	=СУММ(C4:G4)
B5	$x_{31} + x_{32} + x_{33} + x_{34} + x_{35}$	=СУММ(C5:G5)
C6	$x_{11} + x_{21} + x_{31}$	=СУММ(C3:C5)
D6	$x_{12} + x_{22} + x_{32}$	=СУММ(D3:D5)
E6	$x_{13} + x_{23} + x_{33}$	=СУММ(E3:E5)
F6	$x_{14} + x_{24} + x_{34}$	=СУММ(F3:F5)
G6	$x_{15} + x_{25} + x_{35}$	=СУММ(G3:G5)
C12	$\sum_{j=1}^3 c_{nj}$	=C3*C9+C4*C10+C5*C11
D12	$\sum_{j=1}^3 c_{kj}$	=D3*D9+D4*D10+D5*D11
E12	$\sum_{j=1}^3 c_{mj}$	=E3*E9+E4*E10+E5*E11
F12	$\sum_{j=1}^3 c_{sj}$	=F3*F9+F4*F10+F5*F11
G12	$\sum_{j=1}^3 c_{nj}$	=G3*G9+G4*G10+G5*G11
B12	$\sum_{i=1}^5 \sum_{j=1}^3 c_{ij} x_{ij}$	a=СУММ(C12:G12)

Очевидно, что в этом случае ограничения, выражающие требование о том, что суммарное количество продукции, направляемой из каждого пункта отправления во все пункты назначения, не должно превышать запас продукции в данном пункте, запишутся следующим образом:

$$B3 \leq B9; B4 \leq B10; B5 \leq B11;$$

Вторая группа ограничений, выражающих требование о том, что суммарное количество продукции, доставляемой в каждый пункт назначения, не должно быть меньше указанного в заявке, примет вид:

$$C6 \geq C7; D6 \geq D7; E6 \geq E7; F6 \geq F7; G6 \geq G7;$$

2. Применение инструмента **Поиск решения**. После выполнения этих подготовительных действий необходимо ввести произвольные значения, соответствующие элементам решения в ячейки диапазона **C3:G5**, например 1. Затем с помощью команды **Сервис/\*Поиск решения** необходимо вызвать диалоговое окно **Поиск решения** и заполнить все его поля. Предварительно следует выделить с помощью курсора целевую ячейку **B12** (рис. 6.83). Очевидно, что в число ограничений следует включить требование о том, чтобы элементы решения были целыми, неотрицательными числами.

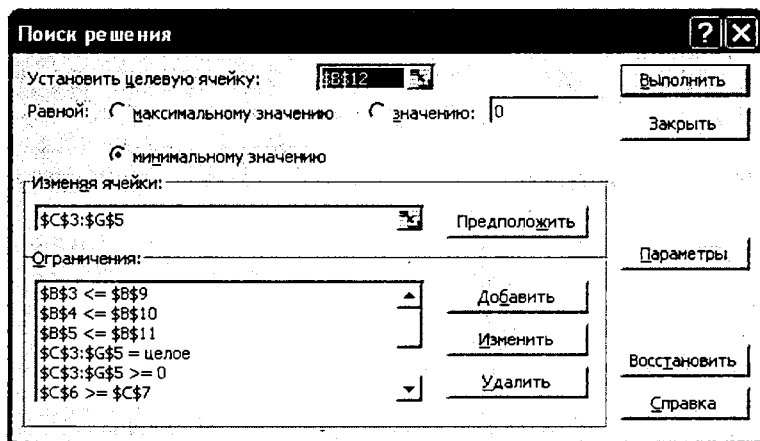


Рис. 6.83. Диалоговое окно **Поиск решения** для транспортной задачи

**Контрольные вопросы**

1. Рабочая книга MS Excel и ее состав.
2. Типы и форматы данных в табличном процессоре MS Excel.
3. Формулы и их синтаксис.
4. Абсолютные и относительные ссылки и их использование для вычислений.
5. Диаграммы как средство анализа данных. Объекты диаграмм: ряды, категории, маркеры данных, легенда.
6. Списки и их возможности по выполнению анализа данных.
7. Консолидация данных в табличном процессоре MS Excel.
8. Сводные таблицы и их возможности по выполнению анализа данных.
9. Средство «подбор параметра» и его возможности по выполнению анализа данных.
10. Средство «таблица подстановки» и его возможности по выполнению анализа данных.

## Глава 7

# ОСНОВЫ ФУНКЦИОНИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

---

---

### 7.1. Базы данных и системы управления ими

База данных (БД) — поименованная совокупность сведений о конкретных объектах реального мира в какой-либо предметной области. БД представляет собой структуру, предназначенную для хранения информации. Предметная область — часть реального мира, которая изучается для организации управления и последующей его автоматизации.

Основными этапами информационной технологии обработки данных с использованием БД и их характеристиками можно считать:

- **сбор данных.** По мере функционирования организации каждое ее действие сопровождается соответствующими записями данных. Совокупность этих данных заносится в БД, где и должна сохраняться. Очевидно, что структура БД должна быть создана заранее;
- **обработка данных.** Для получения информации, отражающей деятельность организации, из данных, которые хранятся в БД, используются следующие типовые операции:
  - классификация или группировка. Первичные данные обычно имеют вид кодов, состоящих из одного или нескольких символов. Эти коды, выражающие определенные признаки объектов, используются для идентификации и группировки записей;
  - сортировка, с помощью которой упорядочивается последовательность записей;

— вычисления, включающие арифметические и логические операции. Эти операции, выполняемые над данными, дают возможность получать новые данные. Этот вид обработки включает укрупнение или агрегирование данных, служащее для уменьшения количества данных и реализуемое в форме расчетов итоговых или средних значений;

- **хранение данных.** Многие данные, необходимые для обеспечения деятельности организации, должны сохраняться для последующего использования. Для этого создаются базы данных;
- **вывод информации.** Для деятельности организации необходимы документы, предназначенные для использования руководителями и работниками, а также для внешних партнеров. При этом такие документы могут создаваться как по запросу или в связи с определенным действием данной организации, так и периодически в конце каждого месяца, квартала или года, что нужно для анализа деятельности и принятия целесообразных решений.

Базы данных могут различаться технологией обработки данных, способом доступа к данным, используемыми моделями данных и другими признаками.

**Централизованные и распределенные базы данных.** По технологии обработки данных базы данных подразделяются на централизованные и распределенные. Централизованная база данных хранится в памяти одной вычислительной системы. Если эта вычислительная система является компонентом сети ЭВМ, возможен распределенный доступ к такой базе. Такой способ использования баз данных часто применяют в локальных сетях ПК.

Распределенная база данных состоит из нескольких частей, хранимых в различных ЭВМ вычислительной сети. Работа с такой базой осуществляется с помощью системы управления распределенной базой данных (СУРБД). Одиночная реляционная база данных может содержать набор таблиц. Каждая из них представляет один из материальных объектов, с которыми имеет дело организация. Однако чаще всего создают несколько отдельных баз данных, каждая из которых связана с одной из организационных функций, обычно продажами, производством, финансами и кадровыми ресурсами. Каждая база данных принадлежит определенному отделу, управляющему ею. Другие отделы имеют доступ только к определенным полям выбранных ими

таблиц. Например, отделу продаж для системы ввода заказов нужно ассоциировать продавцов с обслуженными ими заказами, но этому отделу необязательно иметь доступ к личной информации о продавцах, которой ведаёт отдел кадров.

**Базы данных с локальным и удаленным доступом.** По способу доступа к данным базы данных разделяются на базы данных с локальным доступом и с удаленным (сетевым) доступом. Системы централизованных баз данных с сетевым доступом предполагают различные архитектуры подобных систем: файл — сервер и клиент — сервер.

**Файл — сервер.** Архитектура систем БД с сетевым доступом предполагает выделение одной из машин сети в качестве центральной (сервер файлов). На такой машине хранится совместно используемая централизованная БД. Все другие машины сети выполняют функции рабочих станций, с помощью которых поддерживается доступ пользовательской системы к централизованной базе данных. Файлы базы данных в соответствии с пользовательскими запросами передаются на рабочие станции, где в основном и производится обработка. При большой интенсивности доступа к одним и тем же данным производительность информационной системы падает. Пользователи могут создавать также на рабочих станциях локальные БД, которые используются ими монополично. Таким образом, при реализации данной архитектуры файлы базы данных хранятся на файл-сервере, а обрабатываются рабочими станциями (рис. 7.1).

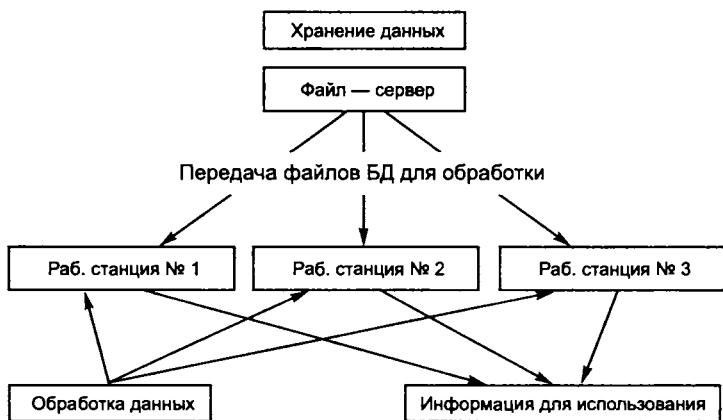


Рис. 7.1. Архитектура файл — сервер



**Клиент — сервер.** В этой концепции подразумевается, что помимо хранения централизованной базы данных центральная машина (сервер базы данных) должна обеспечивать выполнение основного объема обработки данных. Поиск и извлечение необходимых данных осуществляется также сервером по запросу клиента (рабочей станции) на эти данные. В результате сервер выдает клиенту только извлеченные данные. Следовательно, в этом случае файлы не только хранятся, но и обрабатываются на сервере, а рабочая станция управляется клиентской частью программы, позволяющей формировать запросы и получать результаты их выполнения (рис. 7.2).



Рис. 7.2. Архитектура клиент — сервер

**Модели данных.** По своей сущности база данных представляет собой информационную модель объекта. Эту модель образуют элементы данных, связанные соответствующим образом. В этом определении термин «модель» надо понимать как заменитель реального объекта. Свойства модели определяются целью, ради которой она создается. Модели бывают материальными (схема, макет, карта, глобус и т. п.) и информационными. Информационная модель отражает знания человека об объекте моделирования. Модель, реализованная на ЭВМ, называется компьютерной информационной моделью.

Первые базы данных были построены на сетевых и иерархических моделях, которые хорошо подходили для вычислительных машин 50-х. Для задания структуры данных в иерархических базах данных использовались родительско-дочерние отношения, и диаграммы, отображающие такую структуру, напоминали организационные графики для бизнеса или перевернутое дерево, корни которого находятся на самой вершине ие-

рархии. Иерархическая структура представляет совокупность элементов, связанных между собой по определенным правилам. Объекты, связанные иерархическими отношениями, образуют ориентированный граф (перевернутое дерево), вид которого представлен на рис. 7.3.

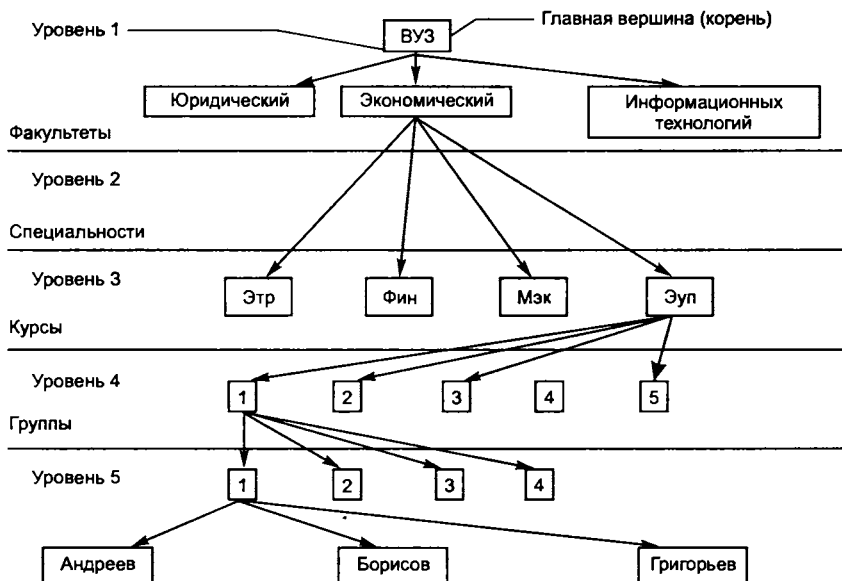


Рис. 7.3. Иерархическая структура данных

К основным понятиям иерархической структуры относятся: уровень, элемент (узел), связь. *Узел* — это совокупность атрибутов данных, описывающих некоторый объект. На схеме иерархического дерева узлы представляются вершинами графа. Каждый узел на более низком уровне связан только с одним узлом, находящимся на более высоком уровне. Иерархическое дерево имеет только одну вершину (корень дерева), не подчиненную никакой другой вершине и находящуюся на самом верхнем (первом) уровне. Зависимые (подчиненные) узлы находятся на втором, третьем и т. д. уровнях. Количество деревьев в базе данных определяется числом корневых записей.

К каждой записи базы данных существует только один (иерархический) путь от корневой записи. Для иерархической БД определен полный порядок обхода — сверху вниз, слева направо.

во. Примерами типичных операторов манипулирования иерархически организованными данными могут быть следующие:

- найти указанное дерево БД (например, факультет);
- перейти от одного дерева к другому;
- перейти от одной записи к другой внутри дерева (например, от отдела — к первому сотруднику);
- перейти от одной записи к другой в порядке обхода иерархии;
- вставить новую запись в указанную позицию;
- удалить текущую запись.

Сетевые базы данных позволили ослабить правила иерархической структуры данных и задать дополнительные отношения между элементами данных. Иерархические и сетевые базы данных обычно являются автономными и их нелегко связать с другими внешними базами данных по глобальной сети (WAN — Wide Area Network).

Доктор Кодд (Codd) в 1970 г. опубликовал в июньском выпуске журнала ассоциации по вычислительной технике (Association for Computing Machinery — ACM) статью под названием «A Relational Model of Data for Large Shared Databanks» («Реляционная модель организации данных для больших, совместно используемых банков данных»). В ней доктор Кодд определил отношение как имеющий название набор кортежей (записей или строк), который имеет атрибуты (поля или столбцы). Один из атрибутов должен содержать уникальное значение, которое позволило бы опознать каждый кортеж. Обычным термином для отношения является таблица, которую любой пользователь может легко себе представить как электронную таблицу. Реляционные базы данных позволяют решить серьезные проблемы, которые встречались в ранних типах баз данных. В иерархических и сетевых базах данных наборы данных и точные связи между каждым двумя наборами определяются как «родитель — потомок» или «старший — подчиненный», соответственно. Для того чтобы извлекать информацию из этих баз данных, программисту необходимо знать структуру всей базы. Использование реляционных баз данных позволяет существенно облегчить процесс извлечения форматированной информации, а также процесс добавления и изменения данных, удалив сложное навигационное программирование.

Реляционная модель ориентирована на организацию данных в виде двухмерных таблиц-отношений. Отношение моделирует

реальный объект и имеет простую графическую интерпретацию: таблица, столбцы которой соответствуют вхождению доменов (реквизитов) в отношение, а строки — набору значений, содержащихся в соответствующих доменах. Каждая реляционная таблица (отношение) представляет собой двухмерный массив и обладает следующими свойствами:

- каждый элемент таблицы — один элемент данных;
- все столбцы в таблице однородные, т. е. все элементы в столбце имеют одинаковый тип (числовой, символьный и т. д.) и длину;
- каждый столбец имеет уникальное имя;
- одинаковые строки в таблице отсутствуют;
- порядок следования строк и столбцов может быть произвольным.

Отношения представлены в виде таблиц, строки которых соответствуют кортежам или записям, а столбцы — атрибутам отношений, доменам, полям (рис. 7.4).

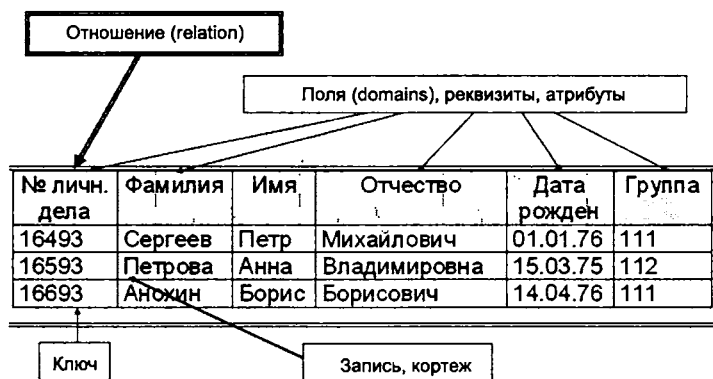


Рис. 7.4. Пример отношения

Поле, каждое значение которого однозначно определяет соответствующую запись, называется простым ключом (ключевым полем). Если записи однозначно определяются значениями нескольких полей, то такая таблица базы данных имеет составной ключ. Чтобы связать две реляционные таблицы, необходимо ключ первой таблицы ввести в состав ключа второй таблицы (возможно совпадение ключей); в противном случае нужно ввести в структуру первой таблицы внешний ключ — ключ второй таблицы.

## 7.2. Состав и классификация информационных систем

### 7.2.1. Состав информационных систем

Базы данных являются важным компонентом информационных систем (ИС), используемых в различных областях деятельности для автоматизации процессов сбора, хранения и поиска информации. Информационная система представляет собой совокупность баз данных, программно-аппаратных средств и персонала, обеспечивающую хранение, обработку и выдачу информации для решения прикладных задач. Основное назначение ИС — информационное обеспечение деятельности предприятия, а основная функция — удовлетворение информационных потребностей пользователей. Структура ИС представлена на рис. 7.5.

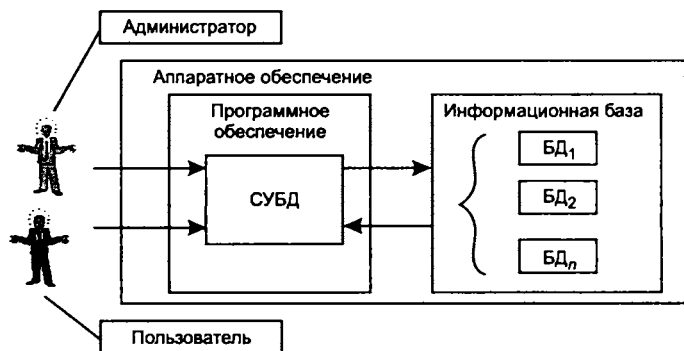


Рис. 7.5. Состав информационной системы

Системы управления базами данных (СУБД) — это компьютерные программы, назначение которых — дать возможность пользователям управлять различными базами данных. СУБД обычно помогают пользователю выполнять следующие работы:

- создание базы данных и занесение в нее информации;
- поиск, изменение и удаление информации;
- проверка наличия ошибок в базе данных;
- анализ информации, создание различных документов на основе информации из базы;
- защита информации, проверка прав доступа пользователей к информации в базе;

- резервное сохранение информации и максимальное ее восстановление в случае сбоев программ и аппаратуры компьютера.

В настоящее время наиболее широкое распространение получили системы управления табличными (реляционными) базами данных. Как правило, эти СУБД достаточно универсальны, т. е. могут использоваться для решения широкого круга задач информационного моделирования в различных сферах человеческой деятельности. Наиболее мощными, надежными и дорогими являются профессиональные СУБД, ориентированные на использование в крупных и средних предприятиях. Примерами таких СУБД могут служить Oracle, Sybase, Microsoft SQL Server. Одной из самых распространенных настольных СУБД, т. е. недорогих систем, предназначенных для использования дома или в небольшом офисе, является Microsoft Access, входящая в состав пакета приложений MS Office.

Администратор базы данных — сотрудник предприятия, отвечающий за систему безопасности БД, включающую контроль доступа пользователей к СУБД, в том числе за резервирование и восстановление. Основные функции группы администратора БД:

- первоначальная загрузка и ведение БД;
- защита данных, обеспечение восстановления БД;
- разработка организационных средств архивирования и принципов восстановления БД;
- разработка дополнительных программных средств и технологических процессов восстановления БД после сбоев;
- анализ обращений пользователей БД, сбор статистики по характеру запросов, по времени их выполнения, по требуемым выходным документам;
- работа с конечными пользователями;
- сбор информации об изменении предметной области;
- сбор информации об оценке работы БД;
- обучение и консультирование пользователей;
- разработка необходимой методической и учебной документации по работе конечных пользователей;
- подготовка и поддержание системных средств, в том числе анализ существующих на рынке программных средств и анализ возможности и необходимости их использования в рамках БД;
- разработка требуемых организационных и программно-технических мероприятий по развитию БД.

### 7.2.2. Классификация информационных систем

Информационные системы могут отличаться степенью автоматизации, характером использования информации, сферой использования и т. п. Рассмотрим особенности ИС, различающихся этими признаками.

**В зависимости от степени автоматизации** информационных процессов в системе управления фирмой информационные системы определяются как ручные, автоматические, автоматизированные.

- ручные ИС характеризуются отсутствием современных технических средств переработки информации и выполнении всех операций человеком. Например, о деятельности менеджера в фирме, где отсутствуют компьютеры, можно говорить, что он работает с ручной ИС;
- автоматические ИС выполняют все операции по переработке информации без участия человека;
- автоматизированные ИС предполагают участие в процессе обработки информации и человека, и технических средств, причем главная роль отводится компьютеру. В современном толковании в термин «информационная система» вкладывается обязательно понятие автоматизируемой системы. Автоматизированные ИС, учитывая их широкое использование в организации процессов управления, имеют различные модификации и могут быть классифицированы, например, по характеру использования информации и по сфере применения.

**Классификация ИС по характеру использования информации:**

- информационно-поисковые системы производят ввод, систематизацию, хранение, выдачу информации по запросу пользователя без сложных преобразований данных. Информационно-поисковые системы широко используются в библиотеках, в кассах продажи железнодорожных и авиабилетов;
- информационно-решающие системы осуществляют все операции переработки информации по определенному алгоритму. Эти системы могут отличаться степенью воздействия выработанной ими результирующей информации на процесс принятия решений и выделить два класса: управляющие и советующие. Управляющие ИС вырабатывают информацию, на основании которой человек принимает

решение. Эти системы обычно используются для решения задач расчетного характера и обработки больших объемов данных. Примером могут служить система оперативного планирования выпуска продукции, система бухгалтерского учета;

- советующие ИС вырабатывают информацию, которая принимается человеком к сведению и не превращается немедленно в серию конкретных действий. Эти системы обладают более высокой степенью интеллекта, так как для них характерна обработка знаний, а не данных.

**По сфере применения принято выделять:**

- ИС организационного управления, предназначенные для автоматизации функций управленческого персонала. Учитывая наиболее широкое применение и разнообразие этого класса систем, часто любые информационные системы понимают именно в данном толковании. К этому классу относятся информационные системы управления как промышленными фирмами, так и непромышленными объектами: гостиницами, банками, торговыми фирмами и др. Основными функциями подобных систем являются: оперативный контроль и регулирование, оперативный учет и анализ, перспективное и оперативное планирование, бухгалтерский учет, управление сбытом и снабжением и другие экономические и организационные задачи;
- ИС управления технологическими процессами (ТП) служат для автоматизации функций производственного персонала. Они широко используются при организации поточных линий, изготовлении микросхем, на сборке, для поддержания технологического процесса в металлургической и машиностроительной промышленности;
- ИС автоматизированного проектирования (САПР) предназначены для автоматизации функций инженеров-проектировщиков, конструкторов, архитекторов, дизайнеров при создании новой техники или технологии. Основными функциями подобных систем являются: инженерные расчеты, создание графической документации (чертежей, схем, планов), создание проектной документации, моделирование проектируемых объектов;
- интегрированные (корпоративные) ИС используются для автоматизации всех функций фирмы и охватывают весь цикл работ от проектирования до сбыта продукции. Созда-



ние таких систем весьма затруднительно, поскольку требует системного подхода с позиций главной цели, например получения прибыли, завоевания рынка сбыта и т. д. Такой подход может привести к существенным изменениям в самой структуре фирмы, на что может решиться не каждый управляющий.

### **Классификация ИС по степени структурирования хранящихся данных:**

- документальные ИС. Классические модели и методы теории БД изначально ориентировались на организацию хранения и обработки детально структурированных данных. Чаще всего эти данные представляли собой числовые значения, описывающие те или иные характеристики информационных объектов. Однако на практике оказалось, что чаще информация представлена не в виде структурированных массивов данных, а в виде текстовых документов. Вследствие этого документальные БД (иногда их еще называют полнотекстовыми) сразу выделялись в особый тип баз данных. Исторически сложилось так, что за системами, ориентированными на работу с текстовыми документами, укоренился термин информационно-поисковые системы (ИПС). Хотя, если быть точнее, их следует называть документальными ИПС (ДИПС);
- фактографические ИС. В этих ИПС регистрируются факты — конкретные значения данных (атрибутов) об объектах реального мира. Основная идея таких систем заключается в том, что все сведения об объектах (фамилии людей и названия предметов, числа, даты) сообщаются компьютеру в каком-то заранее обусловленном формате (например, дата — в виде комбинации ДД.ММ.ГГГГ). Информация, с которой работает фактографическая ИС, имеет четкую структуру, позволяющую машине отличать один элемент данных от другого, например, фамилию от должности человека, дату рождения от роста и т. п. Поэтому фактографическая система способна давать однозначные ответы на поставленные вопросы, например: «Сколько велосипедов марки А-18 продал магазин “Спорт” в июне 1997 г.?», «Кто из работников фирмы с датой рождения не ранее 1 января 1970 г. имеет водительские права?», «Какие культурно-исторические памятники Москвы включены в список ЮНЕСКО?» и т. д.

### 7.2.3. Особенности поиска в документальных информационных системах

Документальные информационные системы обслуживают принципиально иной класс задач, которые не предполагают однозначного ответа на поставленный вопрос. Базу данных таких систем образует совокупность неструктурированных текстовых документов (статьи, книги, рефераты, тексты законов) и графических объектов, снабженная тем или иным формализованным аппаратом поиска. Основной функцией любой ДИПС является информационное обеспечение потребителей на основе выдачи ответов на их запросы. Осуществление выдачи системой требуемых данных реализуется главной операцией ДИПС — информационным поиском, который представляет собой процедуру отыскания документов, содержащих ответ на заданные потребителем вопросы. В отличие от ФИПС, который в ответ на запрос потребителя осуществляет выдачу конкретных сведений (фактов), ДИПС в результате проведения информационного поиска представляет пользователю список документов или объектов, в какой-то мере удовлетворяющих сформулированным в запросе условиям. Например: выдать список всех статей, в которых встречается слово «энтропия». Принципиальной особенностью документальной системы является ее способность, с одной стороны, выдавать ненужные пользователю документы (например, где «энтропия» употреблена в ином смысле, чем предполагалось), а с другой — не выдавать нужные (например, если автор употребил какой-то синоним или ошибся в написании). Поэтому весьма важным свойством документальных систем является способность по контексту определять смысл того или иного термина. Например, различать смысл слова «коса»: сельскохозяйственный инструмент, отмель или прическа.

Информационный поиск в системе проводится на основе поступившего от потребителя запроса на отыскание необходимой ему информации. Потребность человека в определенной информации в процессе его практической деятельности носит название информационной потребности. Под действием получаемой информации информационная потребность людей постоянно изменяется и трансформируется. Вследствие этого ее невозможно однозначно выразить и описать. Значение информационной потребности потребителя в определенные моменты времени,

выраженное на естественном языке (ЕЯ), и представляет собой информационный запрос, с которым пользователь обращается к системе. Запрос может быть неправильно сформулирован потребителем, а потому не в полной мере отвечать его истинной информационной потребности в момент обращения к системе. Кроме того, необходимо учитывать, что ни одна информационная система не в состоянии адекватно реагировать на запрос, сформулированный на естественном языке. Поэтому его необходимо формализовать. Процесс формализации в данном случае сводится к выявлению набора понятий и терминов, характеризующих информационную потребность, и определению отношений между ними. Такие понятия и термины получили название ключевых слов. Следовательно запрос в документальной информационно-поисковой системе представляет собой совокупность ключевых слов с зафиксированными отношениями между ними.

На следующем этапе информационного поиска пользователь вводит запрос с помощью интерфейса информационно-поисковой системы. Система осуществляет выборку документов, которые по внесенным в систему критериям соответствуют запросу пользователя. При этом она «просматривает» все множество документов, составляющих информационно-поисковое пространство, и формирует результат. Список документов, отобранных информационно-поисковой системой, обычно называют выдачей.

Выданные документы по содержанию можно разделить на две группы — соответствующие информационной потребности пользователя и несоответствующие ей, но с точки зрения информационно-поисковой системы соответствующие запросу. С документами второй группы обычно связывают такое понятие, как информационный шум.

Рассмотрим пример. Допустим, требуется собрать сведения о городе Москве. Самый простой способ это сделать — поискать документы, в тексте которых встречается имя «Москва». Если его включить в запрос, то в результате поиска ДИПС отобразит список документов, в которых оно встречается. Среди них будут и те документы, в которых слово «Москва» будет употребляться не в качестве названия города, а, например, гостиницы, предприятия, реки и т. п. Они, естественно, будут относиться к шуму. В то же самое время в этот список не попадут документы, содержащие сведения о столице РФ, самом крупном городе России и т. п.

Таким образом, при проведении информационного поиска в системе фактически рассматривается не информационная по-

требность пользователя, а только информационный запрос, в ответ на который и выдаются те или иные документы системы. Следовательно, реакцию системы необходимо рассматривать не только по отношению к информационной потребности, но по отношению к его информационному запросу. Релевантность представляет собой соответствие содержания документа информационному запросу в том виде, в каком он сформулирован, а документы, содержание которых отвечает запросу потребителя, носят название релевантных. В результате может случиться так, что часть документов, отвечающих запросу, т. е. релевантных ему, остается не выданной потребителю. В то же время во множестве выданных ему документов присутствуют и такие, которые не отвечают запросу, т. е. не являются релевантными. Следовательно, практически любой реальной ДИПС присущи два основных типа ошибок:

- ошибки 1-го рода (или пропуск цели): невыдача потребителю фактически релевантных его запросу документов;
- ошибки 2-го рода (или ложная тревога, иначе шум): выдача потребителю нерелевантных документов, которые не отвечают поставленному запросу.

Наличие ошибок 1- и 2-го рода в реальной системе обуславливает разбиение всего массива документов системы по отношению к запросу на четыре подмассива: *A* — массив выданных релевантных документов; *B* — массив выданных нерелевантных документов; *C* — массив невыданных релевантных документов; *D* — массив невыданных нерелевантных документов (рис. 7.6).

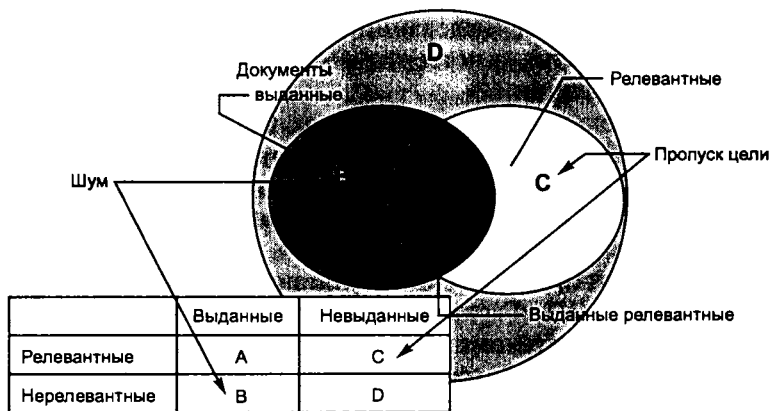


Рис. 7.6. Подмножества документов информационно-поискового пространства

Понимание назначения и устройства ИС позволяет перейти к рассмотрению справочно-правовых систем, которые занимают особое место среди информационных систем.

### 7.3. Справочно-правовые системы

#### 7.3.1. Краткая характеристика российских СПС

Справочно-правовые системы (СПС) — разновидность документальных информационно-поисковых систем, предназначенная для правового обеспечения деятельности специалистов. Необходимость создания и использования таких систем обусловливается наличием огромного количества нормативных и иных правовых документов, регламентирующих политическую, экономическую и социальную сферы. Например, на пополнение СПС «КонсультантПлюс» ежемесячно поступает до 8000 документов [2]. При этом нормативно-правовое пространство Российской Федерации представляет собой многоуровневую систему документов. Так, первый, верхний, уровень этой системы образуют международные акты. Вторые составляют законы, обладающие высшей юридической силой. К числу таких можно отнести Конституцию (Основной Закон) Российской Федерации, Гражданский и Семейный кодексы, федеральные законы. Третий уровень этой системы формируют подзаконные акты федерального уровня (указы Президента, постановления и распоряжения Правительства Российской Федерации, приказы и инструкции министерств и ведомств), а четвертый — правовые акты субъектов Российской Федерации.

Эти обстоятельства породили рынок СПС, где работает достаточно много компаний — разработчиков систем и очень большое число сервисных фирм, осуществляющих поставку и текущее обслуживание СПС. Так, можно назвать следующие продукты и разработавшие их компании:

- КонсультантПлюс (АО «Консультант Плюс»);
- ГАРАНТ (НПП «Гарант-Сервис»);
- «Кодекс» (Центр компьютерных разработок);
- ЮСИС (фирма «Интралекс»);
- «Референт» (ЗАО «Референт-Сервис»);
- «Юридический Мир» (издательство «Дело и право»);

- «Ваше право» и «Юрисконсульт» (фирма «Информационные системы и технологии»);
- «1С:Кодекс», «1С:Гарант», «1С:Эталон» (компания «1С»);
- «Законодательство России» (Ассоциация развития банковских технологий) и некоторые другие.

Кроме того, для обеспечения потребностей в правовой информации государственных ведомств несколько систем аналогичного назначения было создано государственными предприятиями. К их числу относятся следующие СПС:

- «Эталон» (НЦПИ при Министерстве юстиции РФ);
- «Система» (НТЦ «Система» при ФАПСИ).

Степень распространенности продуктов той или иной компании на российском рынке колеблется в широком диапазоне, однако с большой степенью достоверности среди лидеров можно назвать системы КонсультантПлюс и ГАРАНТ. Эти справочные правовые системы используют современные информационные технологии создания и ведения баз данных и обладают существенными достоинствами, к числу которых можно отнести:

- возможность компактно хранить большие объемы информации;
- возможность быстрого поиска нужных документов или их фрагментов в огромных массивах данных;
- возможность с высокой скоростью передавать информацию по телекоммуникациям на любые расстояния.

Сравнительный анализ СПС и, в первую очередь, наиболее распространенных из них, позволяет утверждать, что они отличаются:

- графическими интерфейсами, более или менее удобными для профессионального пользователя;
- глубиной и точностью классификации документов;
- способом обновления баз данных;
- возможностью формирования частных баз данных профессиональным пользователем;
- стоимостью обслуживания.

### **7.3.2. Методы поиска в СПС**

Несмотря на большое количество СПС и их различия, общим для всех них является то, что основной информационной единицей баз данных, входящих в их состав, является документ.

Любой нормативный акт представлен текстом этого документа и вспомогательной, дополнительной, служебной информацией, часто называемой реквизитами документа: номер, дата принятия, вид документа и пр. Как правило, различают две группы реквизитов:

- поисковые, используемые при организации доступа к документам;
- справочные, содержащие сведения, необходимые для работы с документом.

Кроме того, во всех СПС используются три основных метода поиска:

- по реквизитам документа;
- по специализированным классификаторам;
- полнотекстовый.

Каждый метод поиска имеет свои достоинства и недостатки, которые обуславливают преимущества и ограничения при их использовании. Поэтому наилучший результат обеспечивается сочетанием различных методов поиска. Рассмотрим подробнее основные методы доступа к документам, составляющим информационно-поисковое пространство справочно-правовых систем.

#### 7.3.2.1. Поиск по реквизитам документа

Это наиболее простой и удобный метод поиска. Данный метод доступа используется, если известен больший или меньший набор реквизитов, таких, как: тип документа; орган, его издавший; дата принятия; номер документа и т. п. Очевидно, что он применим только в тех ситуациях, когда точно известны реквизиты конкретного документа. С целью облегчения процесса создания запроса для поиска по реквизитам в СПС имеется шаблон с полями, значения которых заполняются пользователем (рис. 7.7).

Часто невозможно точное указание реквизитов, например из-за незнания их. При этом не исключается возможность доступа к документу. В этом случае неоднозначное указание реквизитов или пропуск при создании запроса одного или нескольких из них приведет к увеличению списка документов в выдаче.

Важно понимать, что чем полнее и точнее заданы реквизиты, тем короче полученный список документов, из которых пользователю приходится делать выбор.

The screenshot shows a search interface with the following elements:

- Title:** Поиск по реквизитам
- Search Criteria:**
  - Document number (документ №): [input field]
  - Year (год): [input field]
  - Month (месяц): [input field]
  - Day (день): [input field]
  - Document type (тип документа): [input field]
  - Document status (статус документа): [input field]
  - Document date (дата документа): [input field]
  - Document number (документ №): [input field]
  - Year (год): [input field]
  - Month (месяц): [input field]
  - Day (день): [input field]
- Buttons:**
  - Найти (Find)
  - Сбросить (Reset)
  - Все документы (All documents)

Рис. 7.7. Пример бланка запроса для поиска по реквизитам (СПС Гарант)

### 7.3.2.2. Поиск с использованием тематического классификатора

Как правило, основная проблема при поиске заключается в том, что в большинстве реальных ситуаций чаще всего требуется найти документы, точные реквизиты которых заранее неизвестны. Другими словами, нужно найти документы, зная только общий смысл проблемы.

Для организации такого поиска при создании баз данных в состав реквизитов документа включаются данные, позволяющие отнести документ к той или иной рубрике тематического классификатора. Это означает, что при обработке документа перед его включением в состав базы данных эксперт должен поставить в соответствие элементу информационного массива (документу в целом или его части) определенный элемент классификатора, т. е. таким образом указывается определенная рубрика. После такой обработки пользователь может выбрать интересующую его



рубрику и получить всю подборку документов или их частей, которые были предварительно отобраны экспертом.

Чаще всего в основе таких классификаторов лежит предмет правового регулирования, «тема» документа. При этом классификатор представляет систему юридических понятий (рубрик и ключевых слов), отражающих содержание документов, составляющих информационный массив.

Как правило, классификаторы имеют иерархическую структуру от общего к частному. Число уровней тематического классификатора может быть различным. От глубины иерархии зависит точность выбора списка искомых документов (рис. 7.8).

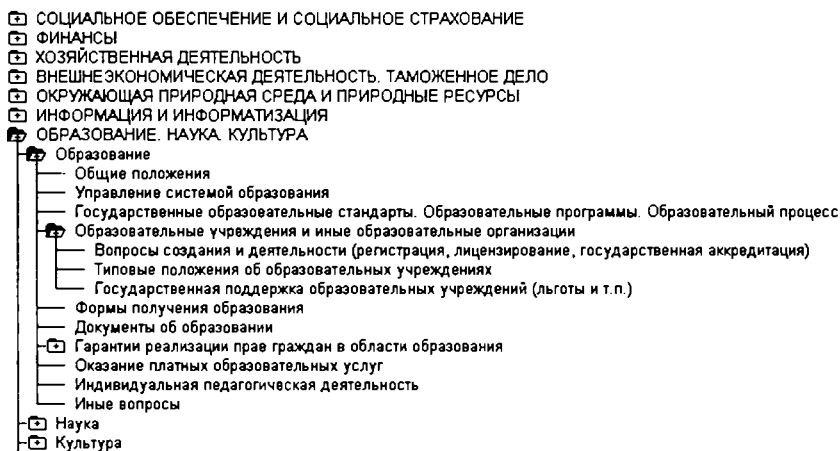


Рис. 7.8. Фрагмент тематического классификатора СПС «КонсультантПлюс»

Основная проблема при использовании специализированных классификаторов для поиска документов состоит в том, что предварительная обработка документов осуществляется «вручную», т. е. документы рубрицируются специалистами компаний — разработчиков СПС. Разумеется, такая работа не может полностью исключать ошибок или субъективного подхода при рубрикации.

### 7.3.2.3. Поиск по содержанию документа (контекстный поиск)

Этот вид поиска основан на автоматической обработке текста и используется, когда затруднено отнесение документа к тому или иному тематическому разделу и неизвестны его рекви-

зиты. При таком поиске осуществляется автоматический перебор всего массива документов и выбор всех тех документов, где это слово встречается.

На основе полнотекстового поиска могут быть реализованы различные способы формирования поисковых запросов. Простейшим запросом является следующий: найти все документы, содержащие одно слово, например **БУХГАЛТЕР**. Более сложным вариантом запроса является запрос на поиск всех документов, содержащих два слова, например **ГЛАВНЫЙ** и **БУХГАЛТЕР**. Если эти два слова соединены логическим условием **И**, то будут отобраны только те документы, в которых есть одновременно оба слова. Если указаны два слова, соединенные логическим условием **ИЛИ**, то будут отобраны документы, в которых есть хотя бы одно из этих слов. Логические выражения могут быть еще более сложными.

Основное преимущество полнотекстового поиска состоит в том, что он не зависит от субъективных мнений или ошибок специалистов-разработчиков и всегда по любому слову дает возможность получить полный и абсолютно точный список документов, где встречается искомое слово. Ничего не будет забыто и пропущено.

Естественно, что у данного метода поиска имеются и свои недостатки. Во-первых, полученный список будет содержать «шум», т. е. документы, где искомое слово используется не в нужном контексте и т. п. Во-вторых, не будут найдены такие документы, в которых встречаются синонимы данного термина.

Как правило, в СПС представлены два способа такого поиска:

- доступ по ключевым понятиям с использованием словаря терминов;
- поиск документов, содержащих произвольно заданные понятия.

Первый способ основывается на выборе ключевого для поиска термина из представленного в системе словаря. Обычно используются многоуровневые, иерархические словари, в которых ключевое понятие каждого уровня раскрывается с помощью уточняющих понятий. Использование словаря не только убыстряет доступ к документам, избавляя от набора с клавиатуры ключевых слов, но и повышает надежность доступа, так как зачастую обычная грамматическая ошибка при наборе ключевых слов делает невозможным доступ к документам.

Второй способ контекстного поиска обычно используется, если ключевые слова не удалось подобрать в словаре терминов. В качестве ключевых понятий в этом случае могут быть заданы слово, словосочетание (слова разделяются пробелом), основа слова.

#### 7.3.2.4. Дополнительные сервисные возможности современных СПС

В современных СПС реализован еще один метод поиска документов — поиск по ситуации (в СПС «КонсультантПлюс» — этот инструмент получил название «Правовой навигатор»). Этот метод целесообразно использовать в тех случаях, когда необходимо найти основные документы по какой-либо правовой проблеме. В этом случае для создания запроса СПС предоставляет пользователю алфавитно-предметный указатель, состоящий из ключевых понятий. Этот указатель размещается в правой части диалогового окна. Близкие по своей сущности ключевые понятия объединены в группы. Перечень групп отображается в левой части окна (рис. 7.9, 7.10). Такая двухуровневая структура облегчает процесс создания запроса.

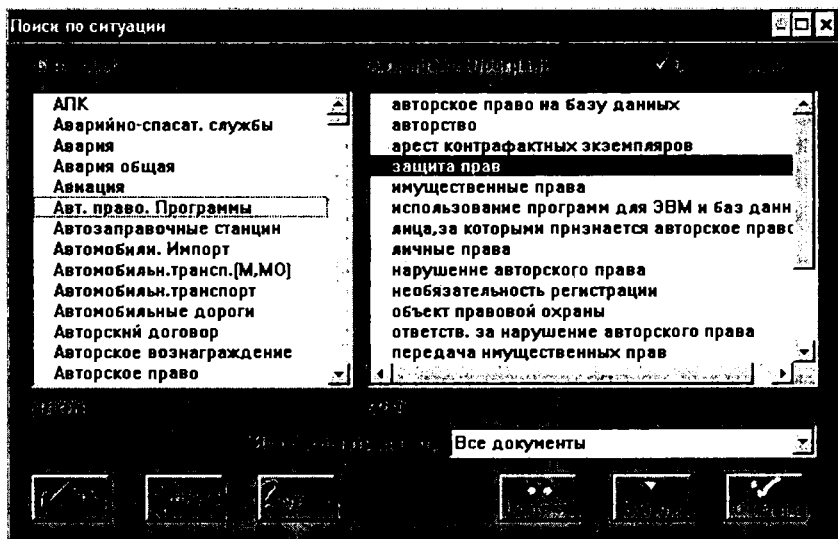


Рис. 7.9. Окно для формирования запроса при поиске по ситуации в СПС «Гарант»

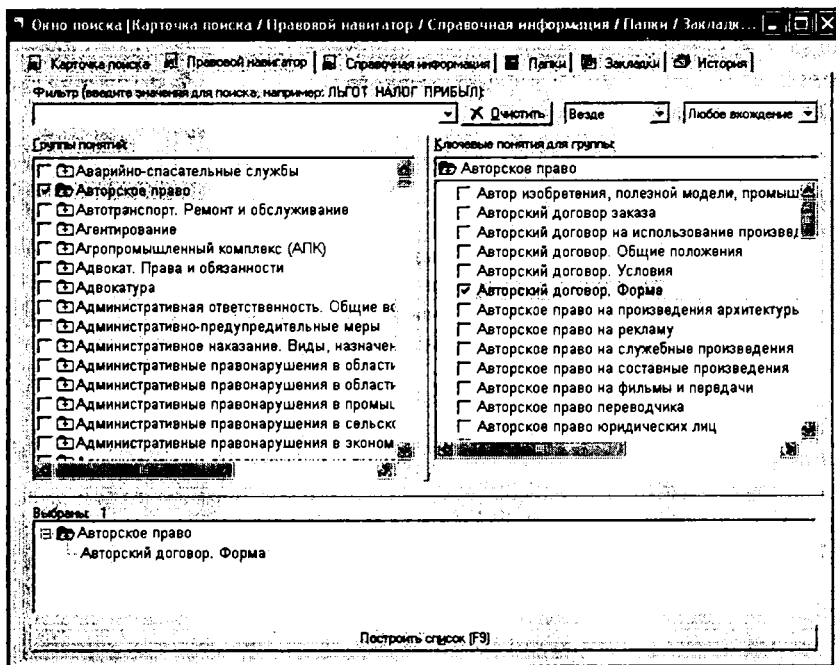


Рис. 7.10. Окно правового навигатора СПС «КонсультантПлюс»

Программные технологии высокого уровня, которые используются в СПС, включают некоторые очень полезные сервисные инструменты, кроме непосредственного поиска документов. К их числу можно отнести:

- возможность создавать собственные постоянные подборки документов по какой-либо проблеме (так называемые папки документов). При этом поиск возможен как по всей базе, так и по конкретным папкам. Пользователи, работающие на различных компьютерах, могут обмениваться такими папками. Это позволяет организовать коллективную работу нескольких специалистов над общей проблемой;
- возможность ставить закладки в тексте, что удобно при работе с большими документами;
- наличие гипертекстовых связей между документами, позволяющих нажатием одной клавиши переходить из одного документа в другой;
- экспорт документов в текстовый редактор Microsoft Word.

Наличие всех этих возможностей делает современные СПС весьма полезным инструментом, необходимым для квалифицированного решения профессиональных задач практически во всех сферах деятельности.

### **Контрольные вопросы**

1. Назначение, состав и классификация информационных систем.
2. Особенности поиска в документальных информационных системах.
3. Назначение и возможности справочно-правовых систем.
4. Методы поиска в справочно-правовых системах.

## Глава 8

# РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ И СИСТЕМЫ УПРАВЛЕНИЯ ИМИ

---

---

### 8.1. Основы построения реляционных баз данных

Понятие реляционный (англ. *relation* — отношение) связано с разработками известного американского специалиста в области систем баз данных Е. Кодда.

Эти модели характеризуются простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных. Реляционная модель ориентирована на организацию данных в виде двумерных таблиц. Каждая реляционная таблица представляет собой двумерный массив и обладает следующими свойствами:

- каждый элемент таблицы — один элемент данных;
- все столбцы в таблице однородные, т. е. все элементы в столбце имеют одинаковый тип (числовой, символьный и т. д.) и длину;
- каждый столбец имеет уникальное имя;
- одинаковые строки в таблице отсутствуют;
- порядок следования строк и столбцов может быть произвольным.

Отношения представлены в виде таблиц, строки которых соответствуют кортежам или записям, а столбцы — атрибутам отношений, доменам, полям.

При проектировании базы данных стремятся к достижению эффективности, что предполагает минимальное дублирование данных, удобство их обработки и обновления. Для удовлетворения этих требований необходимо определить состав отношений,

включаемых в БД, и какие атрибуты должны входить в эти отношения. Теория реляционных баз данных обладает мощным инструментом, который способен помочь разработчику оптимальным образом спроектировать структуру отношений базы данных. Этот инструмент — метод нормализации отношений. Нормализация отношений — пошаговый процесс разложения (декомпозиции) исходных отношений БД на более простые. Каждая ступень этого процесса приводит схему отношений БД в последовательные «нормальные формы». Каждая следующая нормальная форма обладает «лучшими» свойствами, чем предыдущая. В теории реляционных баз данных принято выделять следующую последовательность нормальных форм: первая (1NF); вторая (2NF); третья (3NF); нормальная форма Бойса-Кодда (BCNF); четвертая (4NF); пятая (5NF). Любой нормальной форме соответствует некоторый набор ограничений. Отношение находится в определенной нормальной форме, если оно удовлетворяет набору ограничений этой формы. Переводя структуру отношений БД в формы более высокого порядка, мы добиваемся удаления из таблиц избыточной информации.

Отношение находится в первой нормальной форме (1NF), если значения атрибутов атомарны, т. е. в каждом столбце располагается только одно значение и все неключевые атрибуты функционально зависят от ключа. Отношение находится во второй (2NF) нормальной форме, если выполняются ограничения первой нормальной (1NF) формы и каждый неключевой атрибут функционально полно зависит от первичного ключа (первичный ключ может быть составным). Отношение находится в третьей нормальной форме (3NF), если выполняются ограничения второй нормальной формы (2NF) и если все неключевые атрибуты отношения взаимно независимы и полностью зависят от первичного ключа.

Рассмотрим пример приведения отношения к третьей нормальной форме. Пусть небольшой фирме, продающей комплектующие для компьютеров, требуется сохранять данные о заказах. Эти данные включают: 1) дату заказа; 2) номер заказа; 3) артикул (уникальный номер единицы товара); 4) наименование товара; 5) цену заказанного товара (табл. 8.1).

Эта таблица уже находится в 1NF, так как все ее атрибуты атомарны. В СУБД дата — неделимый тип данных, поэтому, хотя дата заказа и состоит из числа и текста, это — атомарный атрибут. Понятно, что в одном заказе может оказаться несколько

Таблица 8.1. Данные о заказах

Дата	Номер заказа	Артикул	Наименование	Цена
1 фев	1	1672	Процессор AMD ATHLON 64-3000	175
1 фев	1	5301	ASUS P4C800 GOLD	145
1 фев	1	1611	512 MB DDR DIMM PC2700, 333 MHz	75
1 фев	1	1158	256 MB GEFORCE FX-5600	114
1 фев	2	1672	Процессор AMD ATHLON 64-3000	175
1 фев	2	1611	512 MB DDR DIMM PC2700, 333 MHz	75
2 фев	1	1158	256 MB GEFORCE FX-5600	114
2 фев	1	3417	INTEL PENTIUM4-2880/800MHz	189
2 фев	1	1611	512 MB DDR DIMM PC2700, 333 MHz	75
2 фев	2	2660	128 MB GIGABYTE ATI RADEON	138

одинаковых наименований товара. Например, можно в одном заказе затребовать два одинаковых процессора, поэтому составной атрибут **Дата-НомерЗаказа-Артикул** не может быть первичным ключом. Поэтому наше отношение надо привести в 2NF. Чтобы выполнить требования второй нормальной формы, надо добавить к таблице атрибут, который однозначно бы идентифицировал каждую единицу товара, входящую в заказ. Назовем такой атрибут **Учетный №**. Следовательно, таблица, сохраняющая данные о заказах в 2NF, может принять вид, приведенный в табл. 8.2.

Таблица 8.2. Вторая нормальная форма отношения Данные о заказах

Учетный №	Дата	Номер заказа	Артикул	Наименование	Цена
10001	1 фев	1	1672	Процессор AMD ATHLON 64-3000	175
10002	1 фев	1	5301	ASUS P4C800 GOLD	145
10003	1 фев	1	1611	512 MB DDR DIMM PC2700, 333 MHz	75
10004	1 фев	1	1158	256 MB GEFORCE FX-5600	114
10005	1 фев	2	1672	Процессор AMD ATHLON 64-3000	175
10006	1 фев	2	1611	512 MB DDR DIMM PC2700, 333 MHz	75
10007	2 фев	1	1158	256 MB GEFORCE FX-5600	114
10008	2 фев	1	3417	INTEL PENTIUM4-2880/800MHz	189
10009	2 фев	1	1611	512 MB DDR DIMM PC2700, 333 MHz	75
10010	2 фев	2	2660	128 MB GIGABYTE ATI RADEON	138



В этой таблице все атрибуты зависят от атрибута **Учетный №**. Однако в этом отношении существует зависимость атрибутов **Наименование** и **Цена** от атрибута **Артикул**. Требование независимости атрибутов отношения не выполняются. Поэтому данное отношение следует привести в третью нормальную форму (3NF). Для этого предыдущую таблицу требуется разбить на три отношения: **ПРОДАЖИ**, **КОМПЛЕКТУЮЩИЕ**, **ЦЕНЫ** (рис. 8.1).

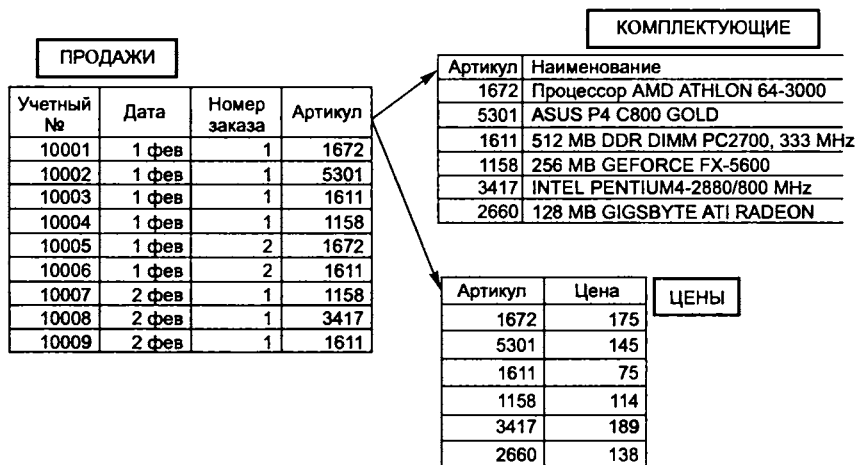


Рис. 8.1. Данные о заказах в третьей нормальной форме

При нормализации отношений достигается две цели. Во-первых, облегчается процесс редактирования таблиц. Пусть в приведенном примере требуется изменить **Наименование** с «512 MB DDR DIMM PC2700,333 МHz» на «512 MB DDR DIMM PC2700, 333 МHz, М-ТЕС». В ненормализованном отношении пришлось бы искать и редактировать все строки, содержащие это наименование, а в нормализованной БД изменяется только одна строка одного отношения. Во-вторых, сокращается объем данных, хранимых в отношении **ПРОДАЖИ**. Подробнее с процессом нормализации и с требованиями нормальных форм старше третьей (3NF) можно ознакомиться в литературе по теории реляционных БД. Данный пример позволяет пояснить некоторые ключевые понятия, необходимые для создания и эффективного использования БД. К их числу относятся:

- **сущность** — реальный или воображаемый объект, информация о котором представляет интерес для пользователя.

Сущность имеет имя, отличающее его от других. Сущность моделируется отношением (таблицей). Иногда для обозначения понятия сущность может использоваться термин «информационный объект»;

- **отношение** — сгруппированные в таблицу логически связанные данные, описывающие сущность (информационный объект);
- **атрибутом** сущности является любая характеристика (реквизит) объекта, отображаемая полями отношения;
- **поле** — один из реквизитов информационного объекта, характеризующий одно из его свойств;
- **запись** — одна из реализаций информационного объекта, т. е. данные, расположенные в одной из строк таблицы;
- **связь** — это взаимная зависимость, устанавливаемая между двумя сущностями.

Поле, каждое значение которого однозначно определяет соответствующую запись, называется простым ключом или ключевым полем. Если записи однозначно определяются значениями нескольких полей, то такая таблица базы данных имеет составной ключ. Чтобы связать две реляционные таблицы, необходимо ключ первой таблицы ввести в состав ключа второй таблицы (возможно совпадение ключей); в противном случае нужно ввести в структуру первой таблицы внешний ключ — ключ второй таблицы.

В нашем примере сущность **ПРОДАЖИ** характеризуется атрибутами **Учетный номер**, **Дата**, **Номер заказа** и **Артикул**. Атрибутами сущности **КОМПЛЕКТУЮЩИЕ** являются **Артикул** и **Название**, а сущности **ЦЕНЫ** — **Артикул** и **Цена**. Очевидно, что эти сущности связаны между собой через атрибут **Артикул**. Связь между сущностями (таблицами) устанавливает отношения между совпадающими значениями в ключевых полях, обычно между полями, имеющими одинаковые имена в обеих таблицах. В большинстве случаев с *ключевым полем* одной таблицы, являющимся уникальным идентификатором каждой записи, связывается *внешний ключ* другой таблицы. Связи между сущностями (межтабличные связи) бывают трех типов:

- связь с отношением «один-ко-многим» представляет собой соединения между одним значением первичного ключа (это сторона «один») и множеством экземпляров этого же значения в поле внешнего ключа (сторона «многие»). Связь с отношением «один-ко-многим» обычно обозначают цифрой

один и символом бесконечности. Связи «один-ко-многим», которые также называют и связями «многие-к-одному», безусловно, являются наиболее распространенными. В отношении «один-ко-многим» каждой записи в таблице *A* могут соответствовать несколько записей в таблице *B*, но запись в таблице *B* не может иметь более одной соответствующей ей записи в таблице *A*. Отношение «один-ко-многим» создается в том случае, когда только одно из полей является полем первичного ключа или уникального индекса;

- связи с отношением «один-к-одному» соединяют значения первичных ключей в двух таблицах. При отношении «один-к-одному» каждая запись в таблице *A* может иметь не более одной связанной записи в таблице *B* и наоборот. Отношения этого типа используются не очень часто, поскольку большая часть сведений, связанных таким образом, может быть помещена в одну таблицу. Отношение «один-к-одному» может использоваться для разделения таблиц, содержащих много полей, для отделения части таблицы по соображениям безопасности, а также для сохранения сведений, относящихся к подмножеству записей в главной таблице. Отношение «один-к-одному» создается в том случае, когда оба связываемых поля являются ключевыми или имеют уникальные индексы;
- при отношении «многие-ко-многим» одной записи в таблице *A* могут соответствовать несколько записей в таблице *B*, а одной записи в таблице *B* — несколько записей в таблице *A*. Этот тип связи возможен только с помощью третьей (связующей) таблицы, первичный ключ которой состоит из двух полей, которые являются внешними ключами таблиц *A* и *B*. Отношение «многие-ко-многим» фактически является двумя отношениями «один-ко-многим» с третьей таблицей, первичный ключ которой состоит из полей внешнего ключа двух других таблиц.

Графическое отображение сущностей, составляющих базу данных, и связей между ними представляет собой информационно-логическую модель объекта. Иногда для обозначения данного понятия используется термин «инфологическая модель». Таким образом, предметная область, данные о которой необходимо использовать для решения практических задач, может быть представлена с помощью инфологической модели, которая отображает предметную область в виде совокупности информационных

объектов (отношений) и структурных связей. В рассматриваемом примере в качестве предметной области являются заказы комплектующих компьютерного оборудования. Инфологическая модель ее представлена на рис. 8.2.



Рис. 8.2. Инфологическая модель базы данных Заказы комплектующих

Таким образом, можно констатировать, что эта база данных включает в себя три отношения: **Продажи**, **Комплектующие** и **Цены**. Эти отношения связаны между собой через атрибут **Артикул**. Для отношения **Продажи** это связь «многие-к-одному».

Даже этот простой пример показывает, что создание базы данных представляет собой сложную творческую задачу. При этом определяющее значение имеет правильное формулирование цели создания БД, т. е. ее назначение и возможное использование. Работа на этом этапе должна проводиться в тесном взаимодействии с будущими пользователями базы данных. Их задача — сформулировать вопросы, ответы на которые предполагается получать с помощью базы данных. В ходе этого процесса формируется перечень необходимых данных. Затем можно проанализировать распределение данных по направлениям. Этим направлениям должны соответствовать отношения (таблицы) БД, а данным — поля (столбцы) в них. Результатом работы на этом этапе должна стать инфологическая модель БД. После этого следует уточнить в каждой таблице поля с уникальными значениями и связи между таблицами.

После этого следует второй этап — создание БД, включающее:

- создание структуры таблиц базы данных;
- ввод данных в таблицы и их редактирование;
- создание связей между таблицами.

Завершение этих работ позволяет перейти к эксплуатации базы данных, в процессе которой пользователь осуществляет обработку данных, содержащихся в таблицах, и вывод информации из базы данных.

## 8.2. Состав и возможности СУБД MS ACCESS

Программа MS Access представляет собой систему управления реляционными базами данных и входит в пакет MS Office фирмы Microsoft. С помощью этой программы пользователь может решать следующие задачи [1]:

1) *организация данных*. Эта задача предполагает возможность создания таблиц и предоставление пользователю возможности работать с данными, хранящимися в них;

2) *ввод и редактирование данных*. MS Access позволяет работать с данными не только в режиме таблицы, но и позволяет пользователю разработать и использовать удобные формы для ввода, редактирования и просмотра данных. Это удобно при большом объеме таблиц, особенно когда она содержит большое количество полей;

3) *связывание таблиц*. Означает возможность создания запросов, форм и отчетов для одновременного отображения сведений из нескольких таблиц;

4) *отбор данных*. Данная СУБД предоставляет пользователю широкие возможности отбора необходимых данных с помощью запросов или фильтрации;

5) *представление данных*. Эта функция обеспечивает создание отчетов, представляющих всевозможные итоговые данные таблиц в удобном и наглядном виде, которые можно просматривать, печатать или публиковать в Интернет или Intranet. В конце концов, именно отчеты являются целью любой системы учета данных. А поскольку требования к внешнему виду отчетов при использовании СУБД на предприятиях исключительно высоки, данные, содержащиеся в отчетах, должны быть профессионально отформатированы и дополнены графиками и диаграммами.

База данных, создаваемая и обрабатываемая с помощью MS Access представляет собой один файл с расширением **.mdb**. Этот файл может включать в себя объекты нескольких типов: таблицы, запросы, формы, отчеты, страницы, макросы и модули. Пе-

речень этих объектов содержится в левой части рабочего окна программы (рис. 8.3).

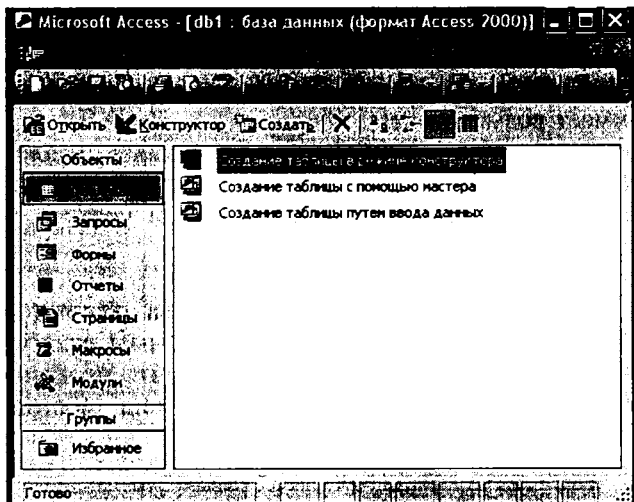


Рис. 8.3. Рабочее окно программы MS Access

СУБД MS Access имеет три основных рабочих режима:

- **начальный режим.** В этом режиме Access функционирует после запуска, до того как будет открыта какая-либо база данных. При этом с базами данных можно работать как с файлами: сжимать, конвертировать, зашифровывать и расшифровывать, а также восстанавливать поврежденные данные. Для выполнения этих операций следует использовать команды **Сервис/Служебные программы** или **Сервис/Защита**. Эти действия, как правило, должны выполняться при закрытой БД, иначе некоторые команды будут недоступными;
- **режим конструктора.** Этот режим предназначен для создания и изменения структуры таблиц и запросов, разработки форм и страниц доступа к данным и форматирования отчетов для печати;
- **режим выполнения или просмотра.** В этом режиме пользователь работает с данными таблиц, форм и отчетов, открытых в отдельных окнах. Режим выполнения или просмотра называется **Режим таблицы** для таблиц и запросов, **Режим формы** — для форм, **Просмотр страницы** — для страниц доступа к данным и **Предварительный просмотр** — для отчетов.

### 8.2.1. Таблицы MS ACCESS

Таблицы являются главным объектом баз данных и составляют их основу. Они предназначены для хранения данных. Таблицы моделируют сущности, входящие в состав той предметной области, которую отображает БД. При работе с таблицами пользователь имеет возможность:

- 1) изменить структуру таблицы — добавлять, удалять, переименовывать столбцы, менять тип и формат данных;
- 2) редактировать таблицы — добавлять и удалять, изменять записи и данные в них;
- 3) сортировать и отбирать записи.

В программе MS Access существуют два режима работы с таблицами: конструктора и таблицы. Основным режимом является режим таблицы, позволяющий выполнять практически все операции, перечисленные выше. Внешний вид таблицы в этом режиме представлен на рис. 8.4.

Учетный №	дата	Номер заказа	Артикул
10001	01.02.2005	1	1672
10002	01.02.2005	1	5301
10003	01.02.2005	1	1611
10004	01.02.2005	1	1158
10005	01.02.2005	2	1672
10006	01.02.2005	2	1611
10007	02.02.2005	1	1158
10008	02.02.2005	1	3417
10009	02.02.2005	1	1611
10010	02.02.2005	2	2660

Рис. 8.4. Отношение Продажи в режиме таблицы

Использование режима конструктора целесообразно при создании таблиц, изменении типа и формата данных. Отношение в этом режиме принимает вид, представленный на рис. 8.5.

Следует помнить, что при этом тип данных характеризует форму представления данных в памяти компьютера и способ их обработки, а формат определяет вид представления данных на внешнем носителе (экране монитора, бумаге,...). СУБД MS Access позволяет использовать в таблицах несколько типов данных, которые приведены в табл. 8.3.

Отсюда следует, что для хранения данных в виде текста или комбинации текста и цифр в Microsoft Access существует два

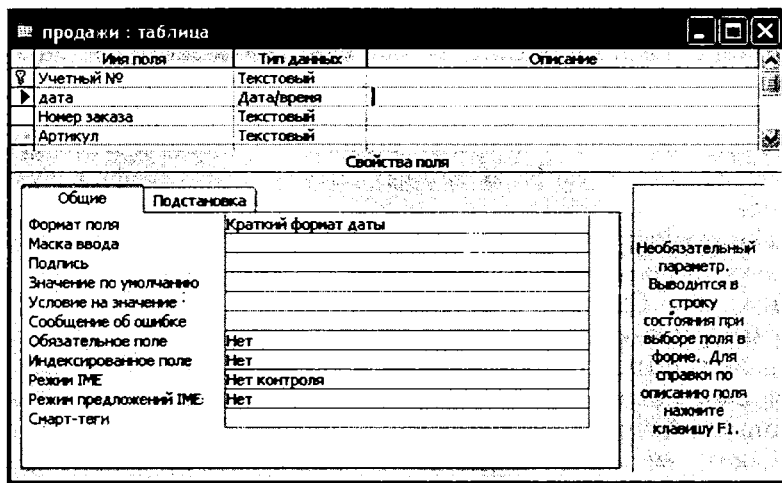


Рис. 8.5. Отношение Продажи в режиме конструктора.

Таблица 8.3. Типы данных, используемые в MS Access

Тип данных	Краткая характеристика типа данных
Текстовый (Text)	Простой, обычный набор, включая числа, буквы и символы. Длина поля не более 255 символов
Поле МЕМО (Memo)	То же, но не ограничена максимальная длина поля, поэтому можно вводить текст любого размера
Числовой (Number)	Простое, обычное число (не деньги и не дата)
Дата/Время (Date/Time)	Дата или время
Денежный (Currency)	Число, имеющее денежный формат
Счетчик (AutoNumber)	Порядковый номер каждой записи
Логический (Yes/No)	Содержит ответ на вопрос «истина/ложь?»
Поле объекта OLE (OLE Object)	Позволяет установить связь с другой базой данных или с другим файлом

типа данных: текстовые и поля **МЕМО**. Текстовый тип данных используется для хранения таких данных, как имена, адреса, а также чисел, не требующих вычислений, например номеров телефонов, инвентарных номеров или почтовых индексов. В текстовом поле может находиться до 255 знаков. По умолчанию устанавливается размер поля 50 знаков. Свойство **Размер поля (FieldSize)** определяет максимальное количество знаков, которые можно ввести в текстовое поле.



Тип данных «Поле МЕМО» используется для хранения более 255 знаков. В поле МЕМО может находиться до 65 536 знаков. Для хранения форматированного текста или длинных документов вместо поля следует использовать поле объекта OLE.

Использование в таблицах полей типа «поле объекта OLE» позволяет хранить в базах данных рисунки, фотографии, аудио-записи и т. п.

Для хранения числовых данных в Microsoft Access может быть использовано несколько типов данных: Числовой, Дата/Время, Денежный и Счетчик.

Числовые поля используются для хранения собственно числовых данных, которые должны использоваться в математических вычислениях, за исключением денежных расчетов, а также вычислений, требующих высокой точности. Тип и размер значений, которые могут находиться в числовом поле, можно изменить в свойстве **Размер поля (FieldSize)**. СУБД MS Access позволяет использовать восемь вариантов этого свойства. Оно определяет внутреннее представление числа в оперативной памяти компьютера, в том числе и количество ячеек оперативной памяти, которое следует выделить для хранения данного числа. Особенности числовых данных, определяемые свойством Размер поля, приведены в табл. 8.4.

Таблица 8.4. Особенности числовых данных в MS Access

Размер поля	Десятичных знаков	Диапазон значений	Байт
Действительное (Decimal)	28	От $-1E-28$ до $1E28-1$	14
Двойное с плавающей точкой (Double)	15	От $-1,79769313486231E308$ до $+1,79769313486231E308$	8
Одинарное с плавающей точкой (Single)	7	От $-3.402823E38$ до $+3.402823E38$	4
Длинное целое (Long Integer)	Нет	От $-2\ 147\ 483\ 648$ до $+2\ 147\ 483\ 647$	4
Целое (Integer)	Нет	От $-32\ 768$ до $+32\ 767$	2
Байт (Byte)	Нет	От 0 до 255	1
Код репликации (Replication ID)	Нет	Не применяется	16
Денежный (Currency)	4	От $-922\ 337\ 203\ 685\ 477,5808$ до $+922\ 337\ 203\ 685\ 477,5807$	8

Данные этой таблицы обозначают, что в поле с размером «Байт» допускается ввод только целых чисел без десятичных знаков от 0 до 255.

Денежный тип поля используют для предотвращения округления во время вычислений. В денежных полях обеспечивается 15 знаков слева от десятичной запятой и 4 знака справа. Денежное поле занимает на диске 8 байт.

Для создания полей, в которые при добавлении записи автоматически вводится уникальное число, в Microsoft Access существует тип данных счетчика. Созданный для записи номер уже не может быть удален или изменен. Наиболее часто используется счетчик последовательно возрастающих чисел. Такой тип счетчика удобно использовать как первичный ключ таблицы.

### 8.2.2. Формы MS ACCESS

Формы представляют собой объекты базы данных, предназначенные для просмотра, ввода и редактирования данных. Формы включены в состав объектов базы данных для обеспечения удобства пользователей. Дело в том, что:

- при большом количестве полей таблицы для просмотра всех данных в одной записи может потребоваться прокрутка;
- отсутствует возможность одновременного обновления данных, размещенных в нескольких таблицах.

При открытии формы Microsoft Access отбирает данные из одной или более таблиц и выводит их на экран с использованием макета, выбранного в мастере форм или созданного пользователем самостоятельно в режиме конструктора. Форма позволяет сосредоточить внимание пользователя на одной записи и отображать поля из нескольких таблиц. Кроме того, форма позволяет отображать рисунки и другие объекты. Форма может содержать кнопки, выполняющие печать, открывающие другие объекты или автоматически выполняющие другие задачи. В СУБД MS Access используются формы трех видов: формы для ввода данных, кнопочные формы и пользовательские диалоговые окна.

Основным видом форм являются формы для ввода данных. Источником данных такой формы являются поля в базовых таблицах и запросах. При этом она может включать не все поля из

каждой таблицы или запроса, на основе которых она создается. Пример формы, созданной для просмотра и редактирования записей отношения **Продажи**, приведен на рис. 8.6.

The image shows a screenshot of a Microsoft Access form window titled "продажи". The form is designed for data entry and editing. It features a grid-like layout with several input fields. The fields and their values are: "Учетный №" (Account No.) with the value "10001", "дата" (date) with the value "01.02.2006", "Номер заказа" (Order No.) with the value "1", and "Артикул" (Article No.) with the value "1672". At the bottom of the form, there is a section labeled "Записи:" (Records:) which includes navigation buttons (back, forward, first, last, search) and a page indicator showing "1" out of "10" records.

Рис. 8.6. Форма для ввода данных **Продажи**

Такая форма позволяет увидеть все поля одной записи таблицы **Продажи** и отредактировать их содержание в случае необходимости. Для того чтобы иметь возможность одновременного обновления данных, размещенных в нескольких связанных отношениях, создаются формы, содержащие соответствующие поля. Такие формы могут быть подчиненными (рис. 8.7) и связанными (рис. 8.8). В первом случае поля всех базовых таблиц размещаются в одном макете формы, во втором — в двух разных, но связанных друг с другом. Правда, в этом случае пользователь имеет возможность открывать только ту из связанных форм, которая нужна ему в данный момент времени. Для создания форм для ввода данных следует в левом окне БД выбрать объект **Формы**, после чего целесообразно использовать команду **Создание формы с помощью мастера** в ее правом окне.

СУБД MS Access предполагает три режима работы с формами. Режимы формы и таблицы используются для просмотра и редактирования данных в таблицах. Режим конструктора позволяет придать форме необходимые свойства, так как в этом случае пользователь получает доступ к макету формы. В макете размещаются и сохраняются все объекты, которые выводятся для просмотра в режиме формы. К числу таких объектов относятся: заголовок, дата и номера страниц, графические элементы и поясняющий текст. Данные в форму поступают из полей в базовом источнике записей, а результаты расчетов — из выражений, которые также располагаются в макете формы.

комплектующие

Артикул: 115  
Наименование: 256 MB GEFORCE FX-5600

продажи

	Цена	Учетный №	дата	Номер заказа
▶	114,006	10004	01.02.2005	1
	114,006	10007	02.02.2005	1
*				

Записей: 1 из 2

Рис. 8.7. Подчиненная форма, включающая поля всех трех отношений БД

комплектующие

продажи

Артикул: 131  
Наименование: 512 MB DDR DIMM PC2700,333 MHz

Записей: 2 из 6 (фильтр)

продажи

	Цена	Учетный №	дата	Номер заказа
▶	75,006	10003	01.02.2005	1
*	75,006	10006	01.02.2005	2
	75,006	10009	02.02.2005	1
*				

Записей: 1 из 3 (фильтр)

Рис. 8.8. Связанные формы Комплектующие и Продажи

Связь между формой и ее источником записей создается с помощью графических объектов, которые называют элементами управления. Наиболее часто для ввода и вывода данных используются элементы управления типа «поле». Макет формы **Комплектующие** и панель элементов управления приведены на рис. 8.9.

Формы и таблицы можно также открывать в режиме сводной таблицы или в режиме диаграммы для анализа данных. В этих режимах пользователь может изменять макет формы для изменения способа представления данных. Существует возможность упорядочивать заголовки строк и столбцов, а также применять



Рис. 8.9. Форма Комплектующие в режиме конструктора

фильтры к полям. При каждом изменении макета сводная форма немедленно выполняет вычисления заново в соответствии с новым расположением данных. В режиме сводной таблицы имеется возможность просматривать исходные данные или сводные данные, упорядочивая поля в областях фильтра, строк, столбцов и данных. В режиме сводной диаграммы можно визуальнo представлять данные, выбрав тип диаграммы и упорядочивая поля в областях фильтра, рядов, категорий и данных.

Кроме рассмотренного типа форм можно создавать кнопочные формы и пользовательские диалоговые окна. Такие разновидности форм могут применяться для организации пользовательского интерфейса. С их помощью можно осуществлять открытие и закрытие других форм, печать отчетов. Для создания кнопочных форм используется команда **Сервис/Служебные программы/Диспетчер кнопочных форм**.

### 8.2.3. Другие объекты MS ACCESS

В состав базы данных могут включаться и другие объекты. Ниже приводится только их краткая характеристика. Такие объекты как отчеты и запросы представляют собой мощное средство анализа данных, поэтому ознакомлению с ними посвящена отдельная глава. Работа же с макросами, модулями и страницами доступа к данным требует специальных знаний. По этим причинам представляется, что в данной главе целесообразно дать только краткую характеристику данных объектов.

**Отчеты** — объекты, предназначенные для вывода информации из базы данных. В СУБД MS Access данные таблиц, запро-

сов и форм могут объединяться в отчете, который можно увидеть на экране монитора или распечатать для дальнейшего использования. Одно из главных достоинств Access заключается в простоте создания полностью отформатированных и готовых для использования отчетов. Создавая отчет, пользователь может управлять размером и внешним видом всех элементов отчета, что позволяет отобразить сведения желаемым образом. Источником записей отчета являются поля в базовых таблицах и запросах. Отчет не должен включать все поля из каждой таблицы или запроса, на основе которых он создается.

**Запросы** служат для выбора записей, обновления таблиц и включения в них новых записей. Чаще всего запросы применяют для выбора конкретных групп записей, удовлетворяющих заданному условию. Кроме того, запросы позволяют комбинировать информацию, содержащуюся в различных таблицах, обеспечивая связанным элементам данных таблиц унифицированный вид.

**Макросы** являются средством автоматизации выполнения повторяющихся операций и представляют собой последовательность макрокоманд. В Access 97 и более ранних версиях Access макросы были главным средством автоматизации. В Access 200x макросы оставлены главным образом для совместимости с базами данных, созданными в ранних версиях Access. Используйте язык программирования Visual Basic for Applications (VBA) для автоматизации действий в базах данных Access 2002.

**Модули** — это наборы функций и процедур, написанных на языке программирования VBA. С их помощью выполняют вычисления настолько сложные, что простого выражения для их описания недостаточно. Процедуры VBA связываются с определенными событиями и выполняются при их наступлении. Примером подобного события может быть щелчок мыши на определенной кнопке формы.

**Страница доступа к данным** — объект, созданный на основе Web-технологий, представляет собой Web-страницу, связанную с данными базы, для опубликования их в Intranet и Интернет. Эти объекты позволяют осуществлять ввод, редактирование данных и взаимодействие с ними не только в пределах базы данных Microsoft Access, но и в Интернете или в интрасети вне базы данных. С их помощью можно рассылать данные по электронной почте. Правда, для работы за пределами БД необходимо наличие браузера Microsoft Internet Explorer 5.01 с пакетом обновления SP2 или более поздней версии.

### 8.3. Анализ данных в MS ACCESS

#### 8.3.1. Общая характеристика средств анализа СУБД MS ACCESS

СУБД MS Access обладает мощными средствами анализа данных. Так, данные в таблицах можно сортировать или фильтровать. Эти процедуры осуществляются так же, как в списках MS Excel. Для сортировки необходимо открыть таблицу и выбрать команду **Записи/Сортировка**. В отличие от табличного процессора MS Access может осуществлять сортировку записей по одному признаку, который выбирается пользователем путем установки курсора в нужный столбец таблицы, или несколькими признакам. В последнем случае выделяются поля, содержащие признаки, по которым должна осуществляться сортировка. Однако при этом следует учитывать, что сортировка производится только по признакам, записанным в смежных столбцах. Она осуществляется поочередно в каждом столбце слева направо. Это означает, что для такой сортировки необходимо сначала изменить макет таблицы таким образом, чтобы соответствующие столбцы располагались рядом. При этом слева должны располагаться признаки, значения которых принимает большее количество записей. Пример сортировки записей отношения **Продажи** по признакам **Дата продажи**, **Номер заказа** и **Артикул** приведен на рис. 8.10.

Для фильтрации данных в таблицах БД можно использовать два типа фильтров: фильтр «по выделенному» или расширенный фильтр. С этой целью используется команда **Записи/Фильтр...** (для

продажи : таблица				
	Учетный №	дата	Номер заказа	Артикул
	10004	01.02.2005	1	1158
	10003	01.02.2005	1	1611
	10001	01.02.2005	1	1672
	10002	01.02.2005	1	5301
	10006	01.02.2005	2	1611
	10005	01.02.2005	2	1672
	10007	02.02.2005	1	1158
	10009	02.02.2005	1	1611
	10008	02.02.2005	1	3417
	10010	02.02.2005	2	2660

Рис. 8.10. Пример сортировки записей по трем признакам

того чтобы эта команда стала доступной пользователю, необходимо открыть таблицу). Затем пользователь выбирает тип фильтра.

При применении фильтра «по выделенному» программа оставляет доступными для просмотра только записи, содержащие признак, совпадающий с тем, который выбрал пользователь. Такой фильтр можно установить, выделив в таблице часть поля, одну или несколько смежных ячеек, содержащих данные, которые должны быть в соответствующих полях результирующего набора. Access отобразит записи, совпадающие с выделенным образцом. В табл. 8.5 показано, каким образом результат применения фильтра зависит от объема выделенных данных.

Таблица 8.5. Особенности работы фильтра по выделенному

Объем выделения	Результат применения фильтра
Ничего	Отбираются записи, значение текущего поля которых совпадает со значением текущего поля текущей записи
Поле целиком	Отбираются записи, в которых значение текущего поля совпадает со значением выделенного поля текущей записи
Начало поля	Отбираются записи, в которых начальные символы текущего поля совпадают с выделенными
Конец поля	Отбираются записи, в которых конечные символы текущего поля совпадают с выделенными
Ряд символов внутри поля, но не в начале и не в конце	Отбираются записи, в которых в текущем поле (в любом его месте) присутствуют выделенные символы
Несколько полей одной записи	Отбираются записи, в которых значения всех соответствующих полей совпадают с выделенными
Вертикальный блок значений одного поля	Отбираются записи, в которых соответствующее поле имеет одно из нескольких выделенных значений
Прямоугольный блок с несколькими полями нескольких записей	Отбираются записи, в которых значения соответствующей группы полей совпадают со значениями полей одной из выделенных групп

Особую разновидность фильтра по выделенному представляет собой результат выполнения команды **Записи/Фильтр/Исключить выделенное**. В этом случае отбираются записи, не содержащие выделенных данных.

В случае использования расширенного фильтра СУБД открывает диалоговое окно с макетом фильтра (рис. 8.11).



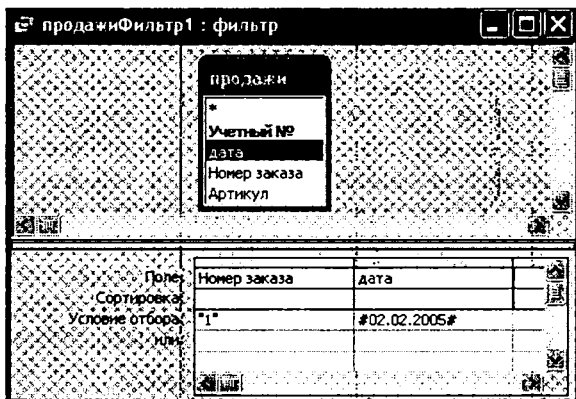


Рис. 8.11. Макет расширенного фильтра к отношению Продажи

В верхней части макета размещается окно с перечнем полей таблицы, а в нижней — бланк для записи условий фильтрации. Чтобы отфильтровать записи таблицы, пользователь должен перетащить с помощью курсора в строку **Поле** бланка названия полей, по которым будет осуществляться отбор записей, а в строку **Условие отбора** ввести соответствующие значения признаков отбора. Размещение этих признаков в одной строке соответствует логической операции **И**, а в разных — **ИЛИ**. Этот вид фильтра позволяет не только отбирать записи, но и указывать состав и очередность полей, выведенных на экран. При осуществлении такой фильтрации можно указать способ сортировки данных. Кроме того, надо иметь в виду, что для отбора записей допускается задавать не только значения полей, но и логические выражения, содержащие эти значения.

Для анализа данных, находящихся в таблицах, MS Access позволяет использовать сводные таблицы и/или диаграммы, создаваемые пользователем на основе этих данных. Для этого необходимо открыть нужную таблицу и выбрать команду **Вид/Сводная таблица** или **Вид/Сводная диаграмма**. В результате откроется окно, содержащее макет соответствующего объекта (таблицы или диаграммы) и список полей. Содержание и последовательность действий по созданию сводной таблицы или диаграммы такие же, как и в табличном процессоре MS Excel. Кроме этих методов, для анализа данных можно использовать отчеты.

Еще одну возможность проанализировать данные дает использование отчетов. Основным предназначением отчетов, как

объектов MS Access, является представление данных для их просмотра как в электронной, так и в печатной форме. Возможность использовать отчеты для анализа данных обусловлена тем, что в них можно не только включать необходимые данные других объектов БД (таблиц, запросов и форм), но и использовать для их обработки формулы и выражения.

Отчеты имеют много общего с формами. Так, источником записей отчета являются поля в базовых таблицах и запросах. Отчет может включать не все поля из каждой таблицы или запроса, на основе которых он создается, а только те, которые необходимы для достижения цели. Основу отчета составляет макет, в котором сохраняются заголовок, дата, номера страниц и выражения для вычисления результирующих данных. Связь между отчетом и его источником данных создается с помощью графических объектов, называемых элементами управления. Элементами управления являются поля, в которых отображаются имена и числа, надписи, в которых отображаются заголовки, а также декоративные линии, графически структурирующие данные и улучшающие внешний вид отчета. Разновидностью отчетов являются наклейки, представляющие собой область данных, которая содержит одно или несколько полей. При необходимости можно добавлять и удалять поля, перемещать и изменять их размеры, а также настроить их параметры, например размер и начертание шрифта.

Существует два режима отображения отчетов. В режиме предварительного просмотра отчет отображается так, как он будет выглядеть при печати. Режим конструктора предоставляет пользователю доступ к макету отчета. При этом можно придать отчету необходимые свойства, а также изменить состав и свойства объектов отчета. В этом режиме можно создавать отчет. Однако обычному пользователю целесообразно для этого использовать **Мастер создания отчетов**.

### **8.3.2. Возможности запросов в MS ACCESS, их состав и виды**

Несмотря на достоинства рассмотренных выше средств анализа данных они не всегда могут в полной мере удовлетворить потребности пользователя. Дело в том, что основное предназначение баз данных состоит не только в хранении данных, но и в своевременном представлении пользователю необходимых ему

сведений. Поэтому важнейшим объектом СУБД MS Access наряду с таблицами являются запросы. Запрос в MS Access представляет собой инструкцию по отбору необходимых пользователю данных. По своему действию запрос имеет много общего с фильтрами. Основное сходство между запросами на выборку и фильтрами заключается в том, что они извлекают подмножество записей из базового объекта (таблицы или запроса), где эти сведения находятся [6]. Однако действие фильтра сводится к тому, что он оставляет доступными для просмотра только те записи, которые отвечают требованиям пользователя, скрывая от него все остальные. При этом новые объекты в состав БД не включаются. Запрос же представляет собой новый объект в составе базы данных. Выбор применения фильтра или запроса зависит от того, как предполагается использовать отобранные записи. Фильтр обычно используют при работе в режиме формы или в режиме таблицы для просмотра или изменения подмножества записей. Запрос следует использовать для выполнения следующих действий:

- просмотр подмножества записей без предварительного открытия конкретной таблицы или формы;
- выбор таблиц, содержащих нужные записи, с возможностью последующего добавления других таблиц;
- отбор полей, выводящихся на экран при отображении результирующего набора записей;
- выполнение вычислений над значениями полей.

Сравнение возможностей фильтров и запросов позволяет сделать выводы о существенных преимуществах последних при работе с данными (табл. 8.6). Возможности запросов по отбору и обработке данных делают их мощным средством анализа данных в СУБД MS Access.

Отличие запросов от других объектов БД MS Access проявляется и в том, что в отличие от таблиц и отчетов для них существует три режима отображения:

- режим SQL. Если можно так выразиться, этот режим позволяет увидеть «истинное лицо» данного объекта БД MS Access. Запрос на отбор данных по заказу № 1, состоявшемуся 2 февраля, можно увидеть на рис. 8.12. Как видно, запрос содержит команды на языке SQL (Structured Query Language — структурированный язык запросов) и указания, в каких таблицах и какие данные нужны пользователю. В таком виде запрос сохраняется в файле БД, если пользователь считает это необходимым;

Таблица 8.6. Сравнительная характеристика средств отбора записей в БД

Возможности объектов БД по отбору с данными	Фильтр	Запрос
Возможность указать поля, которые должны отображаться в результирующем наборе записей	Нет	Да
Возможность просмотра как отдельного объекта в окне базы данных	Нет	Да
Возможность использования с закрытой таблицей, запросом или формой	Нет	Да
Возможность вычисления сумм, средних значений, подсчитывать количество записей и находить другие итоговые значения	Нет	Да
Возможность сортировки записей	Да	Да

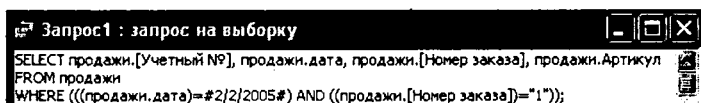


Рис. 8.12. Внешний вид запроса в режиме SQL

- режим таблицы. В этом режиме отображаются данные, отобранные с помощью запроса. В таком режиме запрос существует только в оперативной памяти. В этом режиме запрос представляется на экране монитора в форме таблицы, содержащей данные, отвечающие условиям запроса. Внешний вид запроса в режиме таблицы представлен на рис. 8.13;

Учетный №	дата	Номер заказа	Артикул
10007	02.02.2005	1	1158
10008	02.02.2005	1	3417
10009	02.02.2005	1	1611

Рис. 8.13. Внешний вид запроса в режиме таблицы

- режим конструктора. Этот режим рекомендуется для создания запросов. Как и в предыдущем случае, запрос в этом режиме существует только в оперативной памяти. Внешний вид запроса в этом режиме напоминает макет расширенного фильтра, в чем можно убедиться на примере, представленном на рис. 8.14.

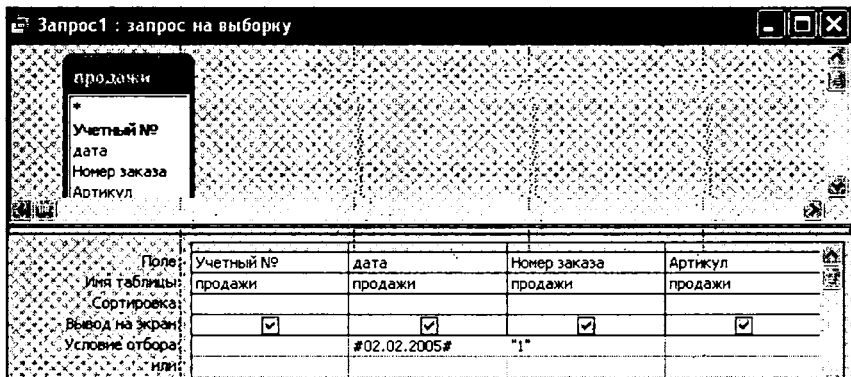


Рис. 8.14. Внешний вид запроса в режиме конструктора

Так же, как и при формировании расширенного фильтра, при создании запросов в строке «Условие отбора» можно записать не только сами значения признаков, но и выражения, содержащие эти значения. Термин «выражение» в СУБД MS Access имеет тот же смысл, что и «формула» в табличном процессоре MS Excel. Таким образом, выражение представляет собой последовательность операндов и операторов, которые создают новое значение из тех данных, которые уже введены в ячейки таблицы. Рассмотрим более подробно компоненты выражений MS Access.

- **Операторы.** В выражениях MS Access используются арифметические операторы и операторы сравнения, уже известные по табличному процессору MS Excel. Наряду с ними в Access нашли применение и некоторые специфичные операторы, например **Between**, **In**, **Is** и **Like**.
- **Литералы.** Этот термин соответствует термину «константа» в MS Excel и служит для обозначения вводимых пользователем значений констант в явном представлении. Они используются в том виде, который имеет на экране, скажем, число 12 345 или строка ABCDE. Литералы чаще всего применяют для создания стандартных значений, а в комбинации с идентификаторами полей — для сравнения значений в полях таблиц и столбцах запросов.
- **Идентификаторы.** Имена объектов Access (например, форм или отчетов) или объектов, входящих в их состав (например, имена полей таблиц), в которых используются числовые или текстовые значения. Данное понятие имеет тот же

смысл, что и понятие «имя» в MS Excel, и означает, что текущее значение идентификатора подставляется в выражение вместо его имени. Например, идентификатор имени поля **Название** отношения **Цены** в выражении возвращает значение (название комплектующих устройств) поля **Название** текущей выбранной записи. В Access имеется пять определенных именованных констант, служащих также идентификаторами: **True**, **False**, **Yes**, **No** и **Null**.

- **Функции.** Назначение и принцип действия этих элементов выражений в полной мере совпадает с тем, что уже стало известным при изучении табличного процессора. Функции возвращают в выражение вместо своего имени вычисляемое ими значение. Большинство функций требует задания аргументов — идентификаторов или значений, заключенных в скобки.

Понимание этих особенностей запросов позволяет глубже изучить особенности их создания и работы. В практике широко используется несколько типов запросов, различающихся своими возможностями. Рассмотрим их более подробно.

Запросы — важнейший инструмент любой системы управления базами данных. Запросы служат для выбора записей, обновления таблиц и включения в них новых записей. Чаще всего запросы применяют для выбора конкретных групп записей, удовлетворяющих заданному условию. Кроме того, запросы позволяют комбинировать информацию, содержащуюся в различных таблицах, обеспечивая связанным элементам данных таблиц унифицированный вид.

Все множество типов запросов можно условно разделить на две группы. Запросы, относящиеся к первой группе, не вызывают изменений в уже существующих к моменту создания запроса объектах БД. Такие запросы называют запросами на выборку. Запросы, принадлежащие ко второй группе, называют запросами на изменение.

#### 8.3.2.1. Запросы на выборку

Запрос на выборку — это такая инструкция по отбору данных, в соответствии с которой программа отбирает данные из одной или более таблиц по заданным условиям, а затем отображает их в нужном порядке. По особенностям действия среди за-

просов на выборку можно выделить простые запросы, групповые запросы и запросы с вычисляемым полем.

**Простые запросы на выборку.** Действие таких запросов ограничивается отбором данных без их обработки. Пример такого запроса был приведен ранее (см. рис. 8.12—8.14). Запрос на выборку может быть создан на основе данных нескольких таблиц и запросов. Предположим, что необходимо получить данные о наименовании и стоимости комплектующих, закупленных по заказу № 2. Этот запрос в режиме конструктора представлен на рис. 8.15. Можно убедиться, что этот запрос содержит поля:

- Номер заказа отношения Продажи;
- Наименование отношения Комплектующие;
- Цена отношения Цены.

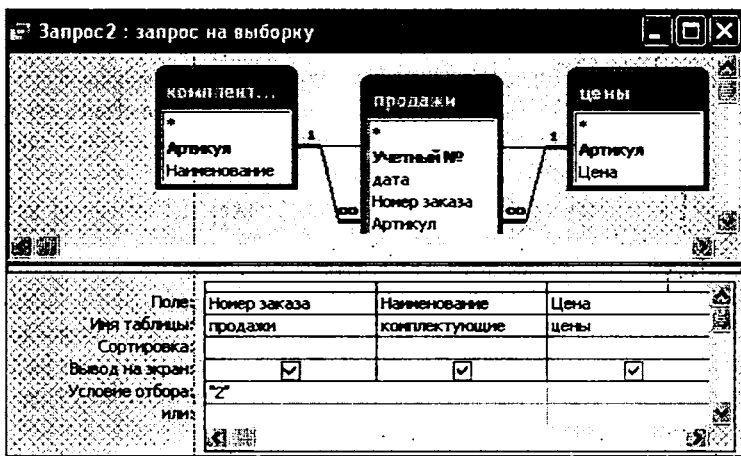


Рис. 8.15. Многотабличный запрос в режиме конструктора

По этому запросу пользователю будут представлены соответствующие данные (рис. 8.16).

Номер заказа	Наименование	Цена
2	512 MB DDR DIMM PC2700,333 MHz	75,00€
2	Процессор AMD ATHLON 64-3000	175,00€
2	128 MB GIGABYTE ATI RADEON	138,00€

Рис. 8.16. Результат выполнения многотабличного запроса

Как уже отмечалось, при создании запросов в строке «Условие отбора» можно записать не только сами значения признаков, но и выражения, содержащие эти значения. При этом СУБД MS Access предоставляет в распоряжение пользователя инструмент, облегчающий процесс создания выражений — **Построитель**. Этот инструмент включается либо кнопкой **Построить** на панели инструментов (обозначена «волшебной палочкой»), либо командой **Построить**, которую можно выбрать из контекстного меню, вызываемого правой клавишей мыши, когда курсор находится в строке «Условие отбора». Рассмотрим пример запроса, который позволит выбрать из нашей БД записи, содержащие сведения о комплектующих, стоимость которых находится в пределах 110—170 евро. В этом случае макет запроса принимает вид, изображенный на рис. 8.17.

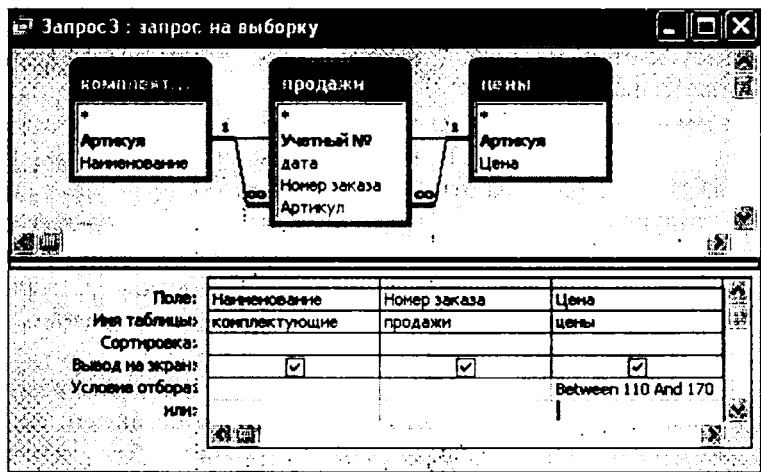


Рис. 8.17. Макет запроса

Условие отбора в этом запросе сформулировано в форме выражения **Between 110 And 170**. Для его записи был использован построитель выражений, представляющий собой диалоговое окно. Это окно, внешний вид которого представлен на рис. 8.18, состоит из трех разделов, располагающихся сверху вниз [1]:

- *поле выражения*. В верхней части окна построителя расположено поле, в котором создается выражение. Ниже находится раздел, предназначенный для создания элементов выражения и их последующей вставки в поле выражения.



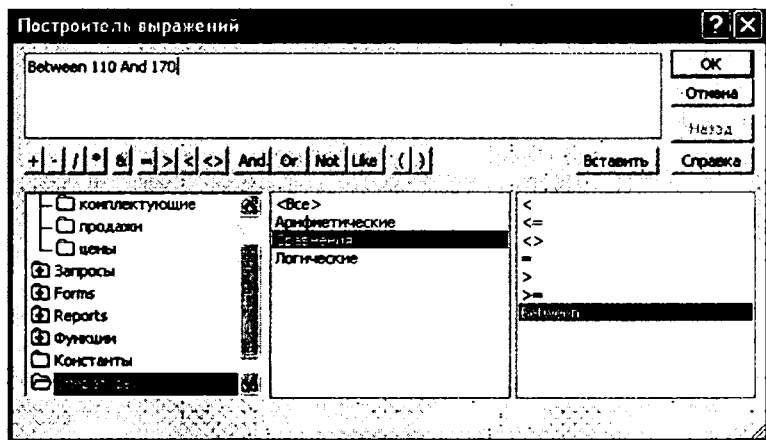


Рис. 8.18. Построитель выражений

Допускается непосредственный ввод части выражения в поле выражения;

- *кнопки операторов.* В средней части окна построителя находятся кнопки с часто используемыми операторами. При нажатии на одну из этих кнопок построитель вставит соответствующий оператор в текущую позицию поля выражения. Чтобы вывести полный список операторов, следует выбрать папку *Операторы* в нижнем левом поле и нужный тип в среднем поле. В правом поле будут выведены все операторы выбранного типа;
- *элементы выражения.* В нижней части окна построителя находятся три поля. В левом поле выводятся папки, содержащие таблицы, запросы, формы, объекты базы данных, встроенные и определенные пользователем функции, константы, операторы и общие выражения. В среднем — задаются определенные элементы или типы элементов для папки, заданной в левом поле. Например, если выбрать в левом поле *Встроенные функции*, то в среднем поле появится список всех типов функций Microsoft Access. В правом поле выводится список значений (если они существуют) для элементов, заданных в левом и среднем полях. Например, если выбрать в левом поле *Встроенные функции* и тип функции в среднем, то в правом поле будет выведен список всех встроенных функций выбранного типа.

Применение этого инструмента не только упрощает процесс создания выражения, но и снижает вероятность синтаксических ошибок при этом. Дело в том, что этот инструмент создает выражение из готовых операндов, которые выбирает пользователь (см. рис. 8.18).

В результате исполнения запроса будут получены необходимые сведения (рис. 8.19).

Наименование	Номер заказа	Цена
256 MB GEFORCE FX-5600	1	114,00€
256 MB GEFORCE FX-5600	1	114,00€
128 MB GIGABYTE ATI RADEON	2	138,00€
ASUS P4C800 GOLD	1	145,00€

Рис. 8.19. Результат выполнения запроса

Очевидно, что эта технология создания выражений может быть использована не только в запросах на выборку.

**Групповые запросы.** Групповые запросы предназначены для вычисления сумм, средних значений, числа значений или выполнения других статистических расчетов. При этом результаты группируются в виде таблицы. Запросы этого типа позволяют получать итоговую информацию для заданного набора записей таблицы. Статистические функции, применяемые для выполнения итоговых расчетов по полям таблиц, перечислены в табл. 8.7.

Таблица 8.7. Статистические функции MS Access

Функция	Описание	Типы полей
Avg ( )	Вычисляет среднее арифметическое значение набора чисел, содержащихся в указанном поле запроса	Все типы, кроме типов Текстовый (Text), Поле МЕМО (Memo) и Поле объекта OLE (OLE Object)
Count ( )	Вычисляет количество записей, не содержащих в данном поле значения Null	Все типы полей
First ( )	Возвращает значение поля из первой записи результирующего набора данных запроса	То же
Last ( )	Возвращает значение поля из последней записи результирующего набора данных запроса	» »

Функция	Описание	Типы полей
Max ( )	Возвращает максимальное значение из набора значений, содержащихся в указанном поле запроса	Все типы, кроме типов Поле MEMO и Поле объекта OLE
Min ( )	Возвращает минимальное значение из набора значений, содержащихся в указанном поле запроса	То же
StDev ( ), StDevP ( )	Возвращают смещенное и несмещенное значение среднеквадратичного отклонения, вычисляемого на основе значений, содержащихся в указанном поле запроса	Все типы, кроме типов Текстовый, Поле MEMO и Поле объекта OLE
Sum ( )	Возвращает сумму значений, содержащихся в заданном поле запроса	То же
Var ( ) , VarP ( )	Возвращает значение смещенной и несмещенной дисперсии, вычисляемой на основе значений, содержащихся в указанном поле запроса	» »

Рассмотрим макет и результат выполнения группового запроса на конкретном примере. Пусть необходимо узнать количество единиц комплектующих и суммарную стоимость комплектующих каждого наименования по двум заказам. В этом случае запрос в режиме конструктора примет вид (рис. 8.20).

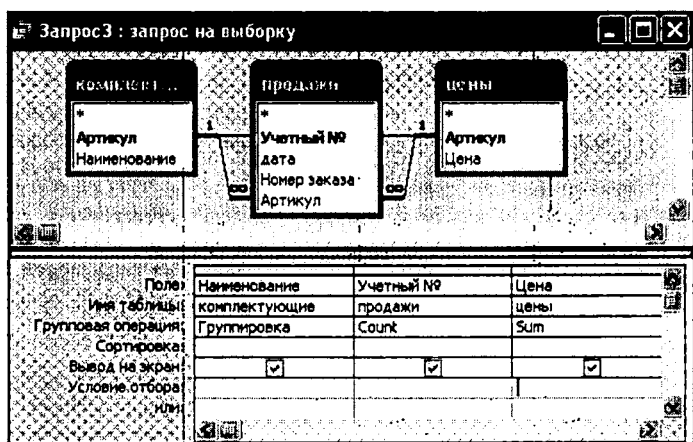


Рис. 8.20. Макет группового запроса

Следует обратить внимание, что в нижней части макета, предназначенной для записи условий отбора данных, появилась новая строка **Групповая операция**. Внести такое изменение в макет запроса можно двумя способами:

- нажать кнопку **Групповые операции** на панели инструментов (обозначена символом  $\Sigma$ ) или
- вызвать контекстное меню с помощью правой клавиши мыши, поместив при этом курсор в нижнюю часть макета запроса, и в этом меню выбрать команду **Групповые операции**.

Задать статистическую функцию для вычисления итоговых значений по каждому из полей запроса можно путем выбора ее из раскрывающегося списка, который появляется после установки курсора в строке **Групповая операция**. Запрос, подготовленный описанным способом, представит необходимые данные (рис. 8.21).

Наименование	Count: Учетный	Sum: Цена
128 MB GIGABYTE ATI RADEON	1	138,00€
256 MB GEFORCE FX-5600	2	228,00€
512 MB DDR DIMM PC2700,333 MHz	3	225,00€
ASUS P4C800 GOLD	1	145,00€
INTEL PENTIUM4-2880/800MHz	1	189,00€
Процессор AMD ATHLON 64-3000	2	350,00€

Рис. 8.21. Результат выполнения группового запроса

**Перекрестные запросы.** Перекрестный запрос также предназначен для использования в расчетах и представления данных в структуре, облегчающей их анализ. Перекрестный запрос, как и групповой, вычисляет сумму, среднее значение, число значений или выполняет другие статистические расчеты, после чего результаты группируются в виде таблицы по двум наборам данных, один из которых определяет заголовки столбцов, а другой — заголовки строк. В частности, этот вид запросов позволяет группировать данные одной таблицы по признакам, обозначенным в другой. Результат выполнения перекрестного запроса напоминает сводную таблицу. Рассмотрим пример создания перекрестного запроса, который группирует записи по номерам заказов, суммируя при этом стоимость комплектующих. Макет такого запроса представлен на рис. 8.22.

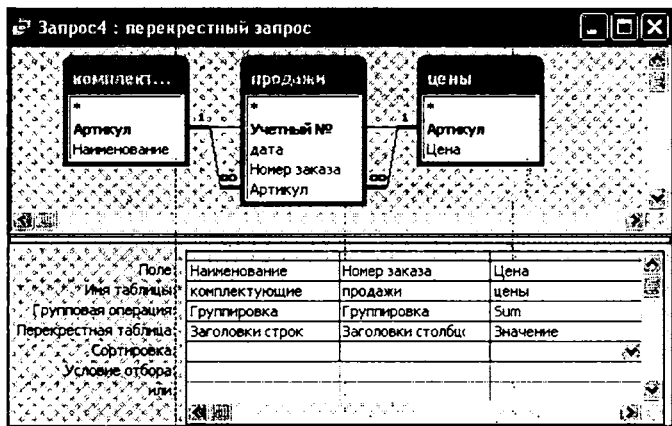


Рис. 8.22. Макет перекрестного запроса

Для получения возможности создания перекрестного запроса необходимо в режиме конструктора запросов выбрать команду **Запрос/Перекрестный**. В результате ее выполнения в бланк запроса вводятся строки **Групповая операция** и **Перекрестная таблица**. Далее необходимо:

1) для поля **Наименование** отношения **Комплектующие**, значения которого должно быть представлено в виде заголовков строк, выбрать значение **Заголовки строк** в строке **Перекрестная таблица** и в строке **Групповая операция** оставить значение **Группировка**;

2) затем для поля **Номер заказа** отношения **Продажи**, значения которого должны быть представлены в виде заголовков столбцов, следует выбрать значение **Заголовки столбцов** строки **Перекрестная таблица**. Следует иметь в виду, что значение **Заголовки столбцов** можно задать только для одного поля. Для этого поля нужно оставить в ячейке строки **Групповая операция** значение **Группировка**;

3) на завершающем этапе создания перекрестного запроса для поля **Цена** отношения **Цены**, значения которого используются при создании перекрестной таблицы, надлежит выбрать значение **Значение** строки **Перекрестная таблица**. Для перекрестных запросов обязательным является требование, чтобы значение **Значение** было установлено только в одном поле. Для этого поля в строке **Групповая операция** выбирается статистическая функция, используемая для заполнения перекрестной таблицы (например, **Sum**, **Avg** или **Count**).

Результат выполнения такого запроса демонстрируется на рис. 8.23.

Наименование	1	2
128 MB GIGABYTE ATI RADEON		138,00€
256 MB GEFORCE FX-5600	228,00€	
512 MB DDR DIMM PC2700,333 MHz	150,00€	75,00€
ASUS P4C800 GOLD	145,00€	
INTEL PENTIUM4-2880/800MHz	189,00€	
Процессор AMD ATHLON 64-3000	175,00€	175,00€

Запись: 6 из 6

Рис. 8.23. Результат выполнения перекрестного запроса

**Запросы с вычисляемым полем.** СУБД MS Access позволяет выполнять в запросе вычисления двух типов.

1. Встроенные вычисления, называемые «итоговыми», для расчета следующих значений по группам записей или по всем записям, отобранным в запросе: сумма, среднее, число значений, минимальное или максимальное значение, стандартное отклонение или дисперсия.

2. Пользовательские вычисления для выполнения расчетов с числовыми и строковыми значениями или значениями дат для каждой записи с использованием данных из одного или нескольких полей. Для ввода таких выражений необходимо создать новое вычисляемое поле непосредственно в бланке запроса.

Первый тип вычислений был рассмотрен ранее. Поэтому далее рассмотрим процесс выполнения пользовательских вычислений в запросе. Для этого необходимо в макет запроса ввести вычисляемое поле. Вычисляемым называется поле, определенное в запросе для вывода результата расчета выражения, а не для сохранения данных. Поэтому результаты вычислений, выводящиеся в таком поле, не запоминаются в базовой таблице. Вычисления в таком поле производятся всякий раз, когда выполняется запрос, следовательно, результаты вычисления всегда представляют текущее содержимое базы данных. Обновить вычисленные результаты вручную невозможно.

Для определения вычисляемого поля можно использовать встроенные функции Microsoft Access или выражения, создаваемые пользователем. Выражения, определяющие вычисляемые поля, создаются с помощью мастера простых запросов или вводятся пользователем в строку **Групповая операция** бланка запро-

са, в которой допускается выбор статистических функций для выполняемых вычислений. В бланке запроса с вычисляемым полем можно также указать условия отбора для изменения результатов вычислений.

Пользовательские вычисления позволяют выполнять расчеты с числовыми и строковыми значениями или значениями дат для каждой записи с использованием данных из одного или нескольких полей. Например, выражение позволяет умножить значения одного поля на итоговое значение, найти разность значений двух полей типа «Дата, время», соединить несколько строковых значений в текстовом поле или создать подчиненные запросы. Примеры выражений приведены в приложении 2.

Для расчетов с использованием формул, определяемых пользователем, требуется создать новое вычисляемое поле прямо в бланке запроса. Вычисляемое поле создается с помощью выражения, которое следует ввести в пустую ячейку Поле в бланке запроса. Рассмотрим пример создания запроса, вычисляющего суммарную стоимость комплектующих одного наименования. В макет запроса включено поле, название которого содержит выражение для вычисления интересующих нас данных:

**Стоимость: [цены]![Цена]\*[Количество\_комплектующих]!  
[Count-Учетный №]**

Особенность синтаксиса выражений MS Access заключается в том, что в качестве оператора присвоения здесь используется не знак равенства, а двоеточие. Макет запроса представлен на рис. 8.24.

Для записи выражения использован построитель (рис. 8.25).

В результате выполнения запроса будут получены необходимые сведения о стоимости закупленных комплектующих, которые можно будет использовать для создания счетов-фактур (рис. 8.26). Таким образом, запросы с вычисляемым полем позволяют получить новые данные на основе тех, которые уже имеются в БД.

Итак, запросы на выборку позволяют получить необходимые пользователю сведения, не внося изменений в таблицы. Однако зачастую в профессиональной деятельности возникает потребность в этом. Эту потребность удовлетворяют запросы на изменение.

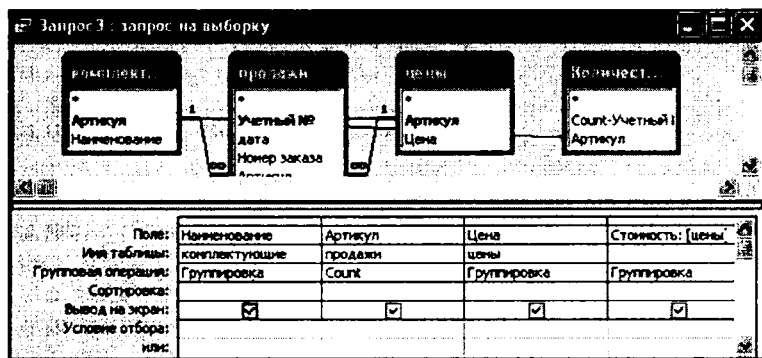
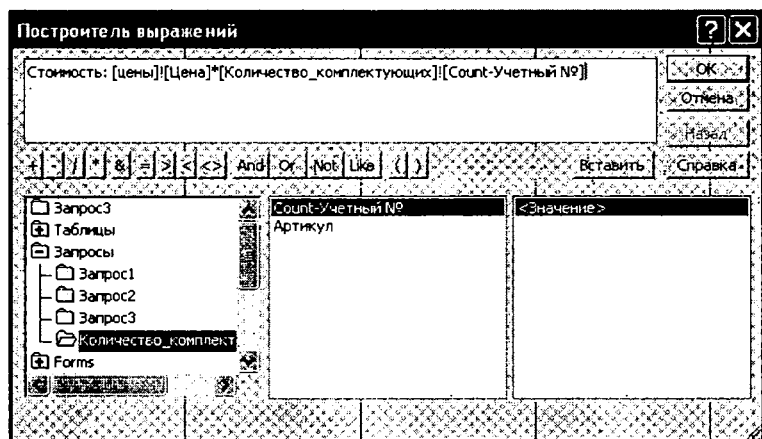


Рис. 8.24. Макет запроса с вычисляемым полем

Рис. 8.25. Диалоговое окно построителя выражений с записанным выражением для получения значений в вычисляемом поле **Стоимость**

Наименование	Count-Артикул	Цена	Стоимость
128 MB GIGABYTE ATI RADEON	1	138,00€	138,00€
256 MB GEFORCE FX-5600	2	114,00€	228,00€
512 MB DDR DIMM PC2700_333 MHz	3	75,00€	225,00€
ASUS P4C800 GOLD	1	145,00€	145,00€
INTEL PENTIUM4-2880/600MHz	1	189,00€	189,00€
Процессор AMD ATHLON 64-3000	2	175,00€	350,00€

Рис. 8.26. Результат выполнения запроса с вычисляемым полем



## 8.3.2.2. Запросы на изменение

Запрос на изменение — это запрос, который за одну операцию вносит изменения в базу данных. Существует четыре типа запросов на изменение: на обновление, на добавление и удаление записей, а также на создание таблиц.

**Запрос на обновление записей.** Запрос на обновление записей предназначен для внесения изменений, общих для группы записей одной или нескольких таблиц. Это дает возможность пользователю изменять данные в существующих таблицах базы данных. Например, на 10 % поднимаются цены на все молочные продукты или на 5 % увеличивается зарплата сотрудников определенной категории. Для создания такого запроса пользователь должен в режиме конструктора запросов выбрать команду **Запрос/Обновление**. В результате ее выполнения в макете запроса появляется новая строка — **Обновление** (рис. 8.27). Далее необходимо выбрать таблицы, содержащие поля для записи условий отбора и требований обновления. Пример на рис. 8.27 демонстрирует запрос, с помощью которого в отношении **Цены** будет записана новая, сниженная на 10 % цена на процессор AMD ATHLON 64-3000.

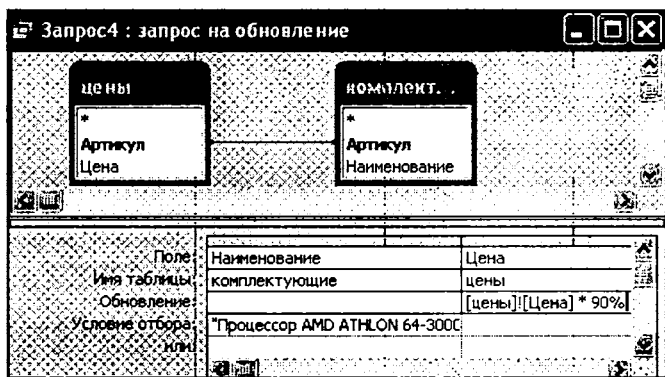


Рис. 8.27. Макет запроса на обновление

**Запрос на добавление и удаление.** Запрос на добавление предназначен для добавления группы записей из одной или нескольких таблиц в другие. Например, появилось несколько новых клиентов, а также база данных, содержащая сведения о них. Чтобы не вводить все данные вручную, их можно добавить в

таблицу «Клиенты». Запрос на добавление также полезен при выполнении следующих действий:

- добавление полей на основе условий отбора. Например, необходимо добавить имена и адреса клиентов с очень крупными заказами;
- добавление записей, если некоторые поля из одной таблицы не существуют в другой. Например, в демонстрационной базе данных «Борей» таблица «Клиенты» содержит 11 полей. Пусть требуется добавить записи из другой таблицы с полями, соответствующими 9 из 11 полям таблицы «Клиенты». Запрос на добавление добавит данные в совпадающие поля и пропустит остальные.

При использовании запросов на добавление следует помнить, что они не могут добавить новые записи в базовые таблицы. Процесс создания такого запроса начинается с выполнения команды **Запрос/Добавление...** в режиме конструктора. Выполнение этой команды вызывает появление диалогового окна **Добавление** (рис. 8.28), в котором необходимо указать имя таблицы назначения. Эта таблица может принадлежать другой базе данных, найти которую можно, пользуясь кнопкой **Обзор...** в нижней части диалогового окна.

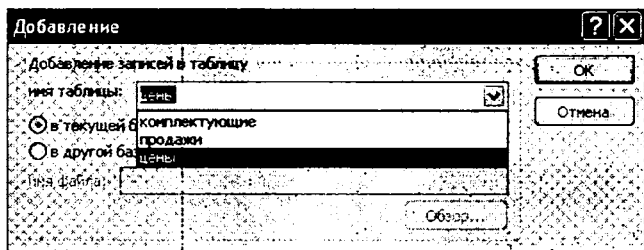


Рис. 8.28. Диалоговое окно **Добавление**

Рассмотрим случай добавления записей о продажах процессора AMD ATHLON 64-3000 в таблицу «Продажи1», созданную для дублирования данных о продажах. Такое дублирование необходимо для обеспечения сохранности данных. Макет запроса представлен на рис. 8.29.

В противоположность запросам на добавление назначение запроса на удаление состоит в удалении группы записей из одной или нескольких таблиц. Например, запрос на удаление, макет которого представлен на рис. 8.30, позволяет удалить записи

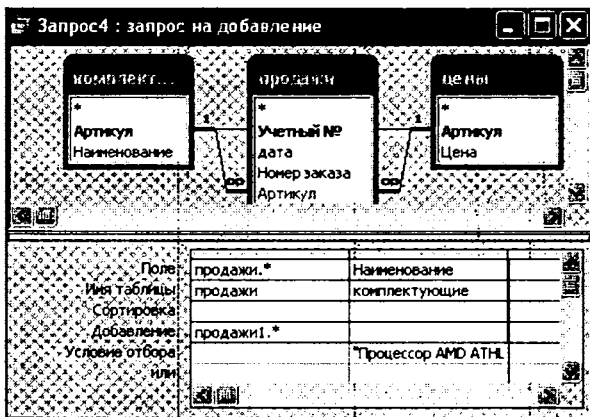


Рис. 8.29. Макет запроса на добавление записей

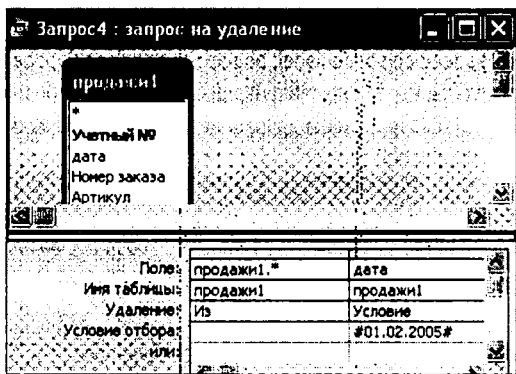


Рис. 8.30. Макет запроса на удаление

о продажах 01.02.2005. С помощью запроса на удаление можно удалять только всю запись, а не отдельные поля внутри нее.

**Запрос на создание таблицы.** Запрос на создание таблицы предназначен для создания новых таблиц на основе всех или части данных из одной или нескольких таблиц. Такой запрос полезен для выполнения следующих действий:

- создание таблицы для экспорта в другую базу данных Microsoft Access. Например, требуется создать таблицу, содержащую все поля из таблицы **Продажи**, а затем экспортировать эту таблицу в базу данных, используемую бухгалтерией;

- создание отчетов, содержащих данные, соответствующие определенному моменту времени;
- создание резервной копии таблицы;
- создание архивной таблицы, содержащей старые записи. Например, можно создать таблицу, сохраняющую все старые продажи, прежде чем удалить их из текущей таблицы **Продажи**;
- создание новой таблицы осуществляется с помощью команды **Запрос/Создание таблицы...**, которая выводит диалоговое окно (рис. 8.31), куда необходимо ввести название новой таблицы и базы данных, в состав которой войдет создаваемая таблица.

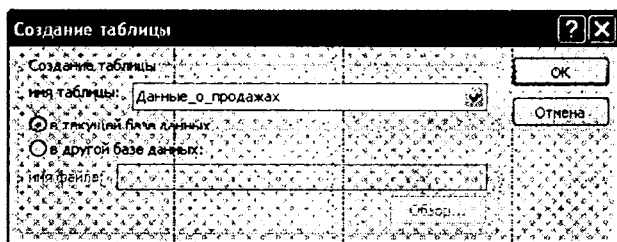


Рис. 8.31. Диалоговое окно для создания запроса

Дальнейшая задача пользователя заключается в заполнении полей макета запроса. Запрос, представленный на рис. 8.32, создает таблицу, содержащую все данные о продажах, содержащиеся в нашей БД.

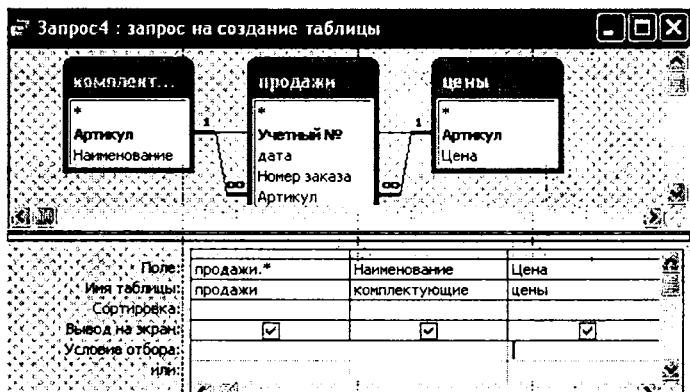


Рис. 8.32. Макет запроса на создание таблицы

Результат выполнения запроса представлен на рис. 8.33.

Учетный №	дата	Номер за	Артикул	Наименование	Цена
10004	01.02.2005	1	1158	256 MB GEFORCE FX-5600	114,00€
10007	02.02.2005	1	1158	256 MB GEFORCE FX-5600	114,00€
10003	01.02.2005	1	1611	512 MB DDR DIMM PC2700,333 MHz	75,00€
10006	01.02.2005	2	1611	512 MB DDR DIMM PC2700,333 MHz	75,00€
10009	02.02.2005	1	1611	512 MB DDR DIMM PC2700,333 MHz	75,00€
10001	01.02.2005	1	1672	Процессор AMD ATHLON 64-3000	175,00€
10005	01.02.2005	2	1672	Процессор AMD ATHLON 64-3000	175,00€
10010	02.02.2005	2	2660	128 MB GIGABYTE ATI RADEON	138,00€
10008	02.02.2005	1	3417	INTEL PENTIUM4-2680/600MHz	189,00€
10002	01.02.2005	1	5301	ASUS P4C800 GOLD	145,00€

Рис. 8.33. Итоговая таблица с данными о продажах

## Контрольные вопросы

1. Основные понятия реляционных баз данных: отношение, атрибут, запись, связь.
2. База данных MS Access и ее основные объекты.
3. Назначение и режимы работы таблиц MS Access.
4. Назначение и режимы работы форм MS Access.
5. Назначение и режимы работы запросов MS Access.
6. Виды запросов MS Access и их возможности для анализа данных.
7. Назначение и режимы работы отчетов MS Access.

## Глава 9

# ОСНОВЫ ПОСТРОЕНИЯ КОМПЬЮТЕРНЫХ СЕТЕЙ

---

---

В процессе решения практических задач с использованием компьютера периодически возникает потребность переноса данных или программ на другой компьютер, например для распечатки файла. Решением этой проблемы может быть использование дискет. Однако это решение не является идеальным. Таким образом, можно считать, что интенсивное использование компьютерной техники и информационных технологий ставит задачу совместного использования ресурсов. При этом под ресурсами будем понимать оборудование, программы и данные. Решение этой задачи привело к созданию компьютерных сетей. В свою очередь, это явилось новым этапом процесса информатизации (рис. 9.1).

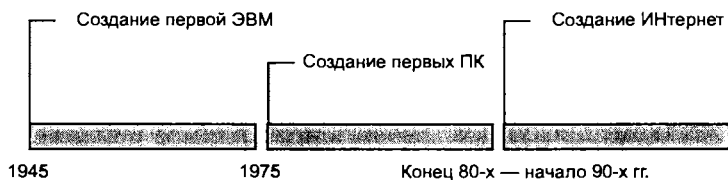


Рис. 9.1. Этапы процесса информатизации

На каждом этапе к уровню компьютерной грамотности специалистов предъявлялись определенные требования. Так, на первом этапе основным требованием было умение программировать, на втором — умение использовать прикладные программы. На современном этапе информатизации пользователь должен уметь не только работать с прикладными программами, но и использовать ресурсы ИНТЕРНЕТ. Это обстоятельство обуславливает необходимость изучения основ функционирования компьютерных сетей.

## 9.1. Принципы функционирования компьютерных сетей

### 9.1.1. Назначение компьютерных сетей

Очевидно, что для совместного использования компьютеры должны быть соединены между собой линиями связи. При физическом соединении двух или более компьютеров образуется компьютерная сеть. По своему назначению сети могут быть информационными, вычислительными и комбинированными — информационно-вычислительными.

Можно считать, что назначение всех видов компьютерных сетей определяется двумя функциями:

- обеспечение совместного использования аппаратных и программных ресурсов сети;
- обеспечение совместного доступа к ресурсам данных.

Другими словами, основные цели, для достижения которых компьютеры объединяются в сети, заключаются в обмене данными между пользователями и совместном использовании ресурсов вычислительной системы, а именно: дисковой памяти, программ, принтеров, данных. Такое объединение позволяет:

- перенести файлы с одного компьютера на другой, а не переходить от одного компьютера к другому с дискетой;
- сэкономить рабочее пространство на дисках. К примеру, если текстовым процессором MS Word пользуются одновременно несколько человек, то нет смысла ставить его на каждый компьютер. Лучше выбрать один (наименее загруженный и имеющий достаточно малое время обращения к жесткому диску) и установить необходимое приложение на него. После этого любой пользователь, подключенный к сети, сможет применить этот процессор;
- повысить безопасность информации. Так, если информация является достаточно важной и ее потеря недопустима, то можно поставить на одну из машин дополнительное устройство внешней памяти (стример, ZIP или JAZZ) и организовать периодическое копирование на это устройство тех областей дискового пространства в сети, которые содержат важную информацию;
- решить проблему использования принтеров. Даже если существует необходимость распечатать документы в одно и то же время, нет никакой необходимости оснащать каждый

компьютер сети отдельным принтером. В офисе целесообразно иметь два принтера: струйный (более дешевый) для черновой печати и лазерный — для печати высококачественных документов. Любой пользователь на свое усмотрение выбирает принтер и посылает документ на печать. Если в данный момент принтер занят печатью другого документа, то новый попадает в очередь печати и будет распечатан, как только до него дойдет очередь.

Однако главное преимущество сети заключается в том, что совместное использование данных позволяет создавать такие системы, с которыми либо невозможно работать на автономных ПК, либо такая работа будет крайне неэффективной. Например, одна из основных задач производственной или коммерческой фирмы — управление своим складским хозяйством. Информационная база такой системы хранит перечень товаров, которые производит или продает фирма: классификаторы продукции, массивы приходных и расходных накладных, справочники покупателей, валют и т. п. У такой системы как минимум три коллективных пользователя: отдел продаж, бухгалтерия и руководство фирмы. Определенные сотрудники вводят и распечатывают накладные, другие корректируют справочные данные, третьи готовят сводные финансовые отчеты для бухгалтерии, а руководителям необходимы выборочные данные по какому-то виду товаров.

При решении этой задачи с помощью автономных ПК придется иметь несколько копий базы данных и после обновления информации обмениваться дискетами с коллегами. Склад в постоянном движении, его состояние, фиксируемое базой данных, меняется динамично и непрерывно. Товары поступают и выдаются (меняется файл накладных), изменяется остаток, поступают товары новых видов и, наоборот, исчезают старые наименования (т. е. модифицируются справочники), меняются цены и т. п. Очевидно, что в таких условиях поддерживать целостность и точность нескольких копий БД просто невозможно.

Таким образом, все рассмотренные аргументы иллюстрирует тезис о том, что все абоненты сети могут совместно использовать оборудование, программы и данные сетевой системы, т. е. ее ресурсы. Отсюда следует, что компьютерная сеть — это сложный комплекс взаимосвязанных и согласованно функционирующих программных и аппаратных компонентов: компьютеров, коммуникационного оборудования, операционных систем и сетевых приложений.



### 9.1.2. Абоненты компьютерных сетей и формы их взаимодействия

Объединение в один комплекс средств вычислительной техники, аппаратуры связи и каналов передачи данных требует использования специальной терминологии:

- 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. В качестве станции часто используются сетевые адаптеры (карты) и модемы;
- абонентская система — совокупность абонента и станции. Как правило, абонентскую систему образуют компьютер с установленной в него сетевой картой или модемом;
- коммуникационная среда — линии связи или пространство, в котором распространяются электрические сигналы, и аппаратура передачи данных. Эта среда обеспечивает взаимодействие абонентов.

На базе коммуникационной среды строится коммуникационная сеть, которая обеспечивает передачу информации между абонентскими системами. Такой подход позволяет рассматривать любую компьютерную сеть как совокупность абонентских систем и коммуникационной сети. Обобщенная структура компьютерной сети приведена на рис. 9.2.



Рис. 9.2. Обобщенная структура компьютерной сети

Можно предположить, что характер взаимодействия пользователей в такой сети можно представить в виде графической модели, представленной на рис. 9.3. В соответствии с этой моделью

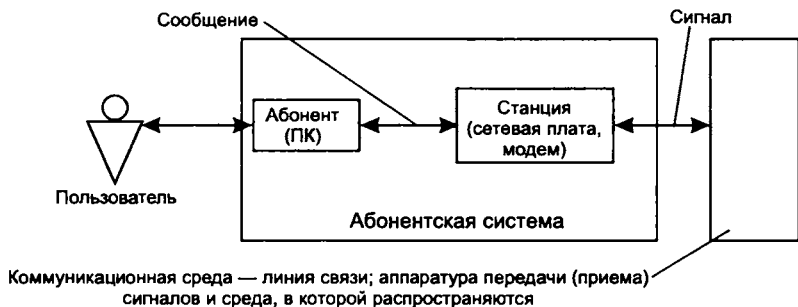


Рис. 9.3. Схема взаимодействия элементов компьютерной сети

компьютер, управляемый пользователем, является либо источником сообщений — передатчиком, либо их приемником. В данном случае под *сообщением* подразумеваются цифровые данные, предназначенные для распространения (передачи или приема) в коммуникационной среде. Станция преобразует сообщение в сигнал. *Сигнал* является переносчиком сообщения и представляет собой физический процесс или явление, который способен обеспечить перенос сообщения в коммуникационной среде.

Очевидно, что содержание сообщения могут составлять не только данные, но и команды, представляющие собой закодированные инструкции по выполнению тех или иных задач на компьютере. Запрограммированную последовательность действий для решения задачи принято называть процессом. Процесс, обеспечивающий решение прикладной задачи, называется прикладным. Отсюда следует, что взаимодействие абонентских ЭВМ в сети можно рассматривать как взаимодействие прикладных процессов конечных пользователей через коммуникационную сеть. Следовательно, коммуникационная сеть обеспечивает физическое соединение между абонентскими ЭВМ — передачу сообщений по каналам связи. Для того чтобы процессы могли взаимодействовать, между ними должна существовать и логическая связь (процессы должны быть инициированы, файлы данных открыты).

В случае, если компьютер используется только для ввода и вывода команд и данных, его называют терминалом.

Анализ работы вычислительных сетей позволяет установить следующие формы взаимодействия между абонентскими ЭВМ [10]:

- терминал — удаленный процесс;
- терминал — доступ к удаленному файлу;

- терминал — доступ к удаленной базе данных;
- терминал — терминал;
- электронная почта.

При реализации этих форм взаимодействия происходит следующее:

- *терминал — удаленный процесс.* В этом случае происходит обращение с терминала одной из абонентских ЭВМ к процессу, находящемуся на другой абонентской ЭВМ сети. При этом устанавливается логическая связь с процессом и проводится сеанс работы с ним. Можно запустить удаленный процесс, получить результаты обработки данных этим процессом. Другими словами, при этом реализуется обработка данных на удаленном компьютере;
- *терминал — доступ к удаленному файлу.* В этом случае можно открыть удаленный файл, модифицировать его или произвести транспортировку этого файла на любое внешнее устройство абонентской ЭВМ для дальнейшей работы с ним в локальном режиме;
- *терминал — доступ к удаленной базе данных.* Эта форма взаимодействия аналогична предыдущей. Только в этом случае производится работа с базой данных в ее полном объеме в соответствии с правами доступа, которыми обладает данный пользователь вычислительной сети;
- *терминал — терминал.* При этой форме взаимодействия осуществляется обмен сообщениями между абонентами сети в диалоговом режиме. Сообщения могут посылаются как отдельным абонентам, так и группам абонентов сети. Длина сообщения не должна превышать некоторой установленной для данной сети величины (обычно — строка на экране терминала);
- *электронная почта* в последнее время стала очень распространенной. В данном случае каждый абонент имеет на своей ЭВМ «почтовый ящик». Это специальный файл, в который записываются все поступающие в его адрес сообщения. Конечный пользователь может проверять в начале работы свой «почтовый ящик», выводить сообщения на печать и передавать сообщения в адрес других абонентов вычислительной сети.

Структурная схема, иллюстрирующая основные формы взаимодействия между абонентскими ЭВМ в сети, приведена на рис. 9.4.



Рис. 9.4. Формы взаимодействия абонентов сети

### 9.1.3. Многоуровневое представление средств сетевого взаимодействия

Организация взаимодействия между устройствами в сети является сложной задачей. Как известно, для решения сложных задач используется универсальный прием — декомпозиция, т. е. разбиение одной сложной задачи на несколько более простых задач-модулей. Процедура декомпозиции включает в себя четкое определение функций каждого модуля, решающего отдельную задачу, и интерфейсов между ними. В результате достигается логическое упрощение задачи, а кроме того, появляется возможность модификации отдельных модулей без изменения остальной части системы.

При декомпозиции часто используют многоуровневый подход. Он заключается в следующем. Все множество модулей разбиваются на уровни. Уровни образуют иерархию, т. е. имеются вышележащие и нижележащие уровни. Множество модулей, составляющих каждый уровень, сформировано таким образом, что для выполнения своих задач они обращаются с запросами только к модулям непосредственно примыкающего нижележащего уровня. С другой стороны, результаты работы всех модулей, принадлежащих некоторому уровню, могут быть переданы только модулям соседнего вышележащего уровня. Такая иерархическая декомпозиция задачи предполагает четкое определение функции каждого уровня и интерфейсов между уровнями. Интерфейс определяет набор функций, которые нижележащий уровень предоставляет вышележащему. В результате иерархической декомпозиции достигается относительная независимость уровней, а значит, и возможность их легкой замены.

Средства сетевого взаимодействия, конечно, тоже могут быть представлены в виде иерархически организованного множества модулей. При этом модули нижнего уровня могут, например, решать все вопросы, связанные с надежной передачей электрических сигналов между двумя соседними узлами. Модули более высокого уровня организуют транспортировку сообщений в пределах всей сети, пользуясь для этого средствами упомянутого нижележащего уровня. А на верхнем уровне работают модули, предоставляющие пользователям доступ к различным службам — файловой, печати и т. п. Конечно, это только один из множества возможных вариантов деления общей задачи организации сетевого взаимодействия на частные подзадачи.

Многоуровневый подход к описанию и реализации функций системы применяется не только в отношении сетевых средств. Такая модель функционирования используется, например, в локальных файловых системах, когда поступивший запрос на доступ к файлу последовательно обрабатывается несколькими программными уровнями (рис. 9.5). Запрос вначале анализируется верхним уровнем, на котором осуществляются последовательный разбор составного символьного имени файла и определение уникального идентификатора файла. Следующий уровень находит по уникальному имени все основные характеристики файла: адрес, атрибуты доступа и т. п. Затем на более низком уровне осуществляется проверка прав доступа к этому файлу, а далее,



Рис. 9.5. Многоуровневая модель файловой системы

после расчета координат области файла, содержащей требуемые данные, выполняется физический обмен с внешним устройством с помощью драйвера диска.

Многоуровневое представление средств сетевого взаимодействия имеет свою специфику, связанную с тем, что в процессе обмена сообщениями участвуют две машины, т. е. в данном случае необходимо организовать согласованную работу двух «иерархий». При передаче сообщений оба участника сетевого обмена должны принять множество соглашений. Например, они должны согласовать уровни и форму электрических сигналов, способ определения длины сообщений, договориться о методах контроля достоверности и т. п. Другими словами, соглашения должны быть приняты для всех уровней, начиная от самого низкого — уровня передачи битов — до самого высокого, реализующего сервис для пользователей сети.

На рис. 9.6 показана модель взаимодействия двух узлов. С каждой стороны средства взаимодействия представлены четырьмя уровнями. Процедура взаимодействия этих двух узлов может быть описана в виде набора правил взаимодействия каждой пары соответствующих уровней обеих участвующих сторон. Формализованные правила, определяющие последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на одном уровне, но в разных узлах, называются протоколом.

Модули, реализующие протоколы соседних уровней и находящиеся в одном узле, также взаимодействуют друг с другом в соответствии с четко определенными правилами и с помощью стандартизованных форматов сообщений. Эти правила принято называть интерфейсом. Интерфейс определяет набор сервисов,

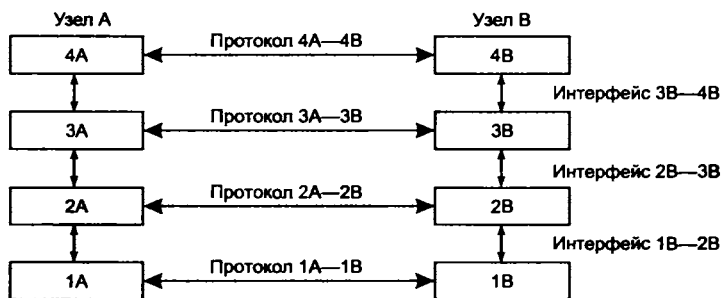


Рис. 9.6. Схема взаимодействия двух узлов

предоставляемый данным уровнем соседнему уровню. В сущности, протокол и интерфейс выражают одно и то же понятие, но традиционно в сетях за ними закрепили разные области действия: протоколы определяют правила взаимодействия модулей одного уровня в разных узлах, а интерфейсы — модулей соседних уровней в одном узле.

Средства каждого уровня должны обрабатывать, во-первых, свой собственный протокол, а во-вторых, интерфейсы с соседними уровнями.

Чтобы еще раз пояснить понятия «протокол» и «интерфейс», рассмотрим пример взаимодействия двух предприятий, связанных между собой деловым сотрудничеством (рис. 9.7). Между предприятиями существуют договоренности и соглашения, например регулярные поставки продукции одного предприятия другому. В этом случае между ними устанавливается порядок взаимодействия, который можно считать «протоколом уровня директоров». Директора посылают свои сообщения и заявки через своих секретарей. Порядок взаимодействия директора и секретаря соответствует понятию межуровневого интерфейса «директор — секретарь».

Выбор способа передачи документов между предприятиями — это уровень компетенции секретарей, они могут решать этот вопрос, не уведомляя об этом своих начальников, так как их протокол взаимодействия связан только с передачей сообще-

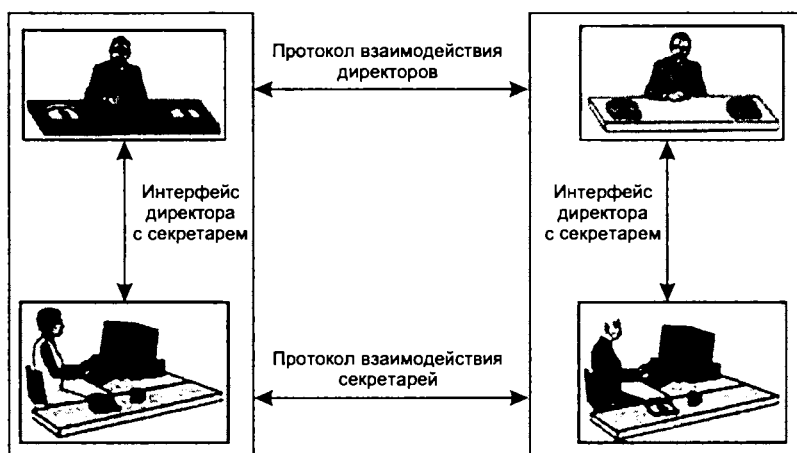


Рис. 9.7. Пример взаимодействия предприятий

ний, поступающих сверху, и не касается содержания этих сообщений. Итак, в данном случае мы имеем дело с двумя уровнями — уровнем директоров и уровнем секретарей, и каждый из них имеет собственный протокол, который может быть изменен независимо от протокола другого уровня. Эта независимость протоколов друг от друга и делает привлекательным многоуровневый подход.

#### 9.1.4. Модель взаимодействия открытых систем

Очевидно, что основной задачей, решаемой при создании компьютерных сетей, является обеспечение совместимости оборудования по электрическим и механическим характеристикам и обеспечение совместимости информационного обеспечения (программ и данных) по системе кодирования и формату данных. По своему характеру — это задача стандартизации. Для ее решения Международный институт стандартов ISO (*International Standards Organization*) разработал соответствующие технические предложения, которые послужили основой для создания так называемой модели *OSI* (*модель взаимодействия открытых систем — Model of Open System Interconnections*).

При рассмотрении этой модели следует иметь в виду, что:

- *открытой* называется система, взаимодействующая с другими системами в соответствии с принятыми стандартами;
- *архитектура* вычислительной сети — это описание ее общей модели.

Модель *OSI* в настоящее время является наиболее распространенной сетевой моделью. Она описывает общие функции систем, предполагая, что системы в сети взаимодействуют на семи уровнях: прикладном, представительном, сеансовом, транспортном, сетевом, канальном и физическом. Каждый уровень имеет дело с одним определенным аспектом взаимодействия сетевых устройств. Самый верхний уровень — прикладной. На этом уровне пользователь взаимодействует с вычислительной системой. Самый нижний уровень — физический, обеспечивающий обмен сигналами между устройствами. Обмен данными в открытых системах происходит путем их перемещения с верхнего уровня компьютера-отправителя на нижний, затем транспортировки и, наконец, обратным воспроизведением на компьютере-получателе в результате перемещения с нижнего



уровня на верхний (рис. 9.8). Преимущество этой модели заключается в том, что при изменении одного из уровней нет необходимости вносить изменения в другие, т. е. нет жесткой зависимости всех уровней от одного. Взаимодействуют только два соседних уровня.

При передаче информации от прикладного процесса в сеть происходит ее обработка уровнями модели взаимодействия открытых систем. Смысл этой обработки заключается в том, что каждый уровень добавляет к информации процесса свой заголовок — служебную информацию, которая необходима для адреса-

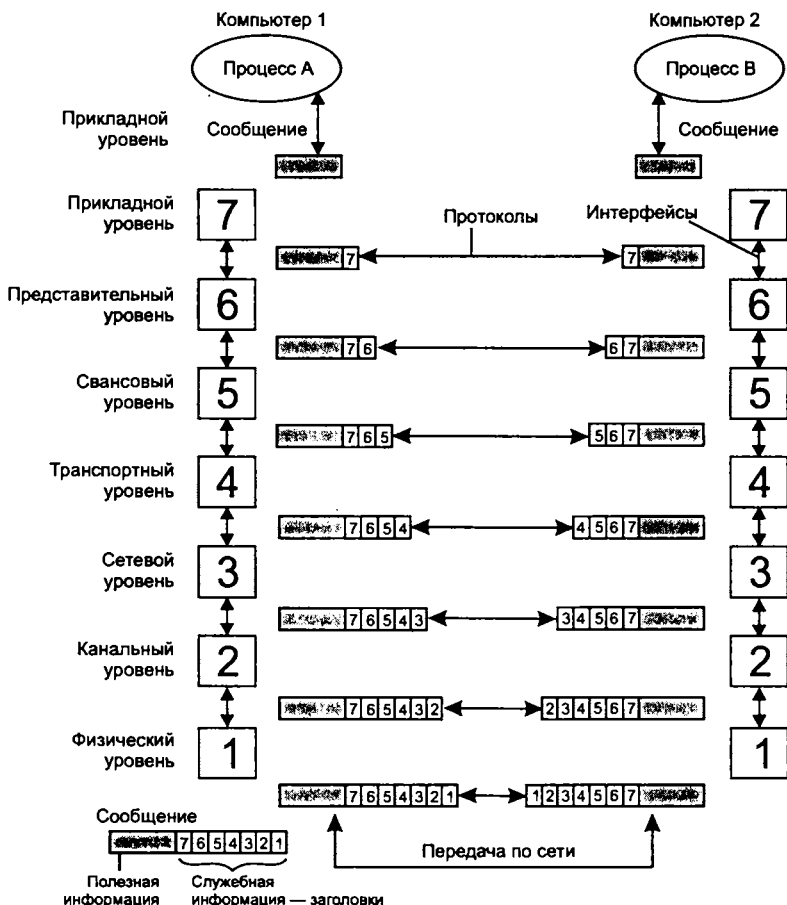


Рис. 9.8. Схема взаимодействия компьютеров в сети в соответствии с моделью OSI

ции сообщений и для некоторых контрольных функций. Обработанным таким образом сообщением, через коммуникационную сеть поступает на абонентские ЭВМ вычислительной сети. Каждая абонентская ЭВМ, принявшая сообщение, дешифрирует адреса и определяет, предназначено ли ей данное сообщение.

Предназначение различных уровней модели OSI кратко может быть описано следующим образом.

**7-й уровень** — *прикладной* — обеспечивает поддержку прикладных процессов конечных пользователей. Этот уровень определяет круг прикладных задач, реализуемых в данной вычислительной сети. Он также содержит все необходимые элементы сервиса для прикладных программ пользователя. На прикладной уровень могут быть вынесены некоторые задачи сетевой операционной системы.

**6-й уровень** — *представительный* — определяет представление данных в кодах и форматах, принятых в каждой конкретной системе. В некоторых системах этот уровень может быть объединен с прикладным.

**5-й уровень** — *сеансовый* — реализует установление и поддержку сеанса связи между двумя абонентами через коммуникационную сеть. Он позволяет производить обмен данными в режиме, определенном прикладной программой, или предоставляет возможность выбора режима обмена. Сеансовый уровень поддерживает и завершает сеанс связи.

**4-й уровень** — *транспортный* — обеспечивает интерфейс между процессами и сетью. Он устанавливает логические каналы между процессами и обеспечивает передачу по этим каналам информационных пакетов, которыми обмениваются процессы. Логические каналы, устанавливаемые транспортным уровнем, называются *транспортными каналами*.

**3-й уровень** — *сетевой* — определяет интерфейс *оконечного оборудования данных* пользователя с сетью коммутации пакетов. Он также отвечает за маршрутизацию пакетов в коммуникационной сети и за связь между сетями — реализует межсетевое взаимодействие.

**2-й уровень** — *канальный* — реализует процесс передачи информации по информационному каналу. Информационный канал — логический канал, он устанавливается между двумя ЭВМ, соединенными физическим каналом. Канальный уровень обеспечивает управление потоком данных в виде кадров, в которые упаковываются информационные пакеты, обнаруживает ошибки

передачи и реализует алгоритм восстановления информации в случае обнаружения сбоев или потерь данных.

**1-й уровень — физический** — выполняет все необходимые процедуры в канале связи. Его основная задача — управление аппаратурой передачи данных и подключенным к ней каналом связи.

Очевидно, что объединение трех верхних уровней представляет собой *процесс* или *прикладной процесс*. Эти уровни определяют функциональные особенности вычислительной сети прикладной системы.

Необходимая совместимость на каждом из семи возможных уровней архитектуры компьютерной сети обеспечивается соответствующими протоколами. Они определяют характер аппаратного взаимодействия компонентов сети (*аппаратные протоколы*) и характер взаимодействия программ и данных (*программные протоколы*). Эти стандарты устанавливают правила выполнения действий, необходимых для обмена сообщениями, и последовательность их выполнения. Очевидно, что для реализации протоколов необходимы соответствующие программы.

В случае, если два компьютера соединены между собой прямым соединением, то на низшем (физическом) уровне протокол их взаимодействия определяют конкретные устройства физического порта (параллельного или последовательного) и механические компоненты (разъемы, кабель и т. п.). На более высоком уровне взаимодействие между компьютерами определяют программные средства, управляющие передачей данных через порты. Для стандартных портов они находятся в базовой системе ввода-вывода (*BIOS*). На самом высоком уровне протокол взаимодействия обеспечивают приложения операционной системы. Например, для MS Windows это стандартная программа **Прямое кабельное соединение**, имеющаяся в составе операционной системы (**Пуск/Программы/Стандартные/Связь/Прямое кабельное соединение**).

В более сложных случаях, когда для соединения компьютеров используется коммуникационная среда, необходимо использовать специальные устройства, согласующие параметры сигнала компьютера с параметрами сигнала, передаваемого по каналам связи. Такие устройства называются сетевыми адаптерами.

Перенос информации между абонентами вычислительной сети обеспечивает коммуникационная среда — линии связи или пространство, в котором распространяются электрические сигналы, и аппаратура передачи данных.

### 9.1.5. Краткая характеристика линий связи

Один из вариантов классификации линий связи, используемых в компьютерных сетях, представлен на рис. 9.9.

В настоящее время три типа кабелей используются для создания локальных компьютерных сетей: витая пара проводов, коаксиальный кабель, оптоволоконный кабель.

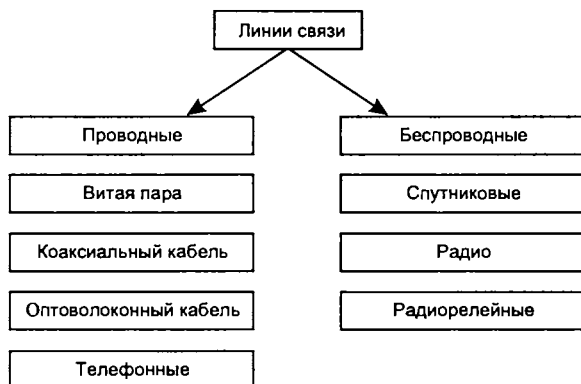


Рис. 9.9. Виды линий связи

Витая пара состоит из двух изолированных проводов, свитых между собой (рис. 9.10). Скручивание проводов уменьшает влияние внешних электромагнитных полей на передаваемые сигналы. Самый простой вариант витой пары — телефонный кабель. Витые пары имеют различные характеристики, определяемые размерами, изоляцией и шагом скручивания. Дешевизна этого вида передающей среды делает ее достаточно популярной для ЛВС.

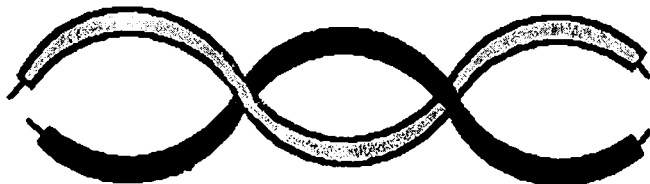


Рис. 9.10. Устройство витой пары

Основной недостаток витой пары — плохая помехозащищенность и низкая скорость передачи информации — 0,25—1 Мбит/с.

Технологические усовершенствования позволяют повысить скорость передачи и помехозащищенность (экранированная витая пара), но при этом возрастает стоимость этого типа передающей среды. Кроме того, при распространении в этой среде сигнал быстро затухает. Поэтому витая пара применима в таких сетях, где взаимное удаление компьютеров не превышает 80—100 метров.

Коаксиальный кабель (рис. 9.11) по сравнению с витой парой обладает более высокой механической прочностью, помехозащищенностью и обеспечивает скорость передачи информации до 10—50 Мбит/с. Для промышленного использования выпускают два типа коаксиальных кабелей: толстый и тонкий. Толстый кабель более прочен и передает сигналы нужной амплитуды на большее расстояние, чем тонкий. В то же время тонкий кабель значительно дешевле. Коаксиальный кабель так же, как и витая пара, является одним из популярных типов передающей среды для ЛВС.

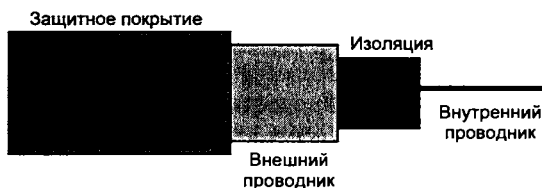


Рис. 9.11. Схема устройства коаксиального кабеля

Затухание сигнала при прохождении через коаксиальный кабель значительно меньше, чем при использовании витой пары. Поэтому коаксиальный кабель можно применять для прокладки локальной сети в пределах одного здания (до нескольких сотен метров).

Оптоволоконный кабель — идеальная передающая среда (рис. 9.12). Он не подвержен действию электромагнитных полей и сам практически не имеет излучения. Последнее свойство по-

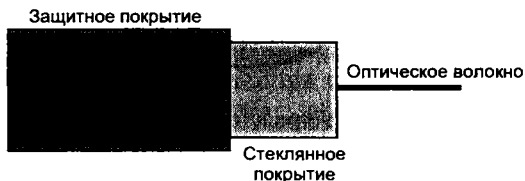


Рис. 9.12. Схема устройства оптоволоконного кабеля

звоняет использовать его в сетях, требующих повышенной секретности информации.

Скорость передачи информации по оптоволоконному кабелю — более 50 Мбит/с. Кроме того, он обладает наименьшим затуханием, что позволяет использовать его для прокладки сетей протяженностью до нескольких километров. По сравнению с предыдущими типами передающей среды он более дорог, менее технологичен в эксплуатации.

Различные каналы связи различаются тремя основными свойствами: *пропускной способностью, помехоустойчивостью, стоимостью.*

По параметру стоимости самыми дорогими являются оптоволоконные линии, самыми дешевыми — телефонные. Однако с уменьшением цены снижается и качество работы линии: уменьшается пропускная способность, сильнее влияют помехи. Практически не подвержены помехам оптоволоконные линии.

*Пропускная способность* — это максимальная скорость передачи информации по каналу. Обычно она выражается в килобитах в секунду (Кбит/с) или в мегабитах в секунду (Мбит/с).

Пропускная способность телефонных линий — десятки, сотни килобит в секунду; пропускная способность оптоволоконных линий и линий радиосвязи измеряется десятками и сотнями мегабит в секунду.

Чаще всего пользователи сети подключаются к узлу через коммутируемые (т. е. переключаемые) телефонные линии. Такое подключение производится с помощью специального устройства, которое называется *модем*. Слово «модем» — сокращенное объединение двух слов: МОдулятор — ДЕМОдулятор. Модем устанавливается как на компьютере пользователя, так и на узловом хост-компьютере. Модем выполняет преобразование физической формы представления информации из компьютерного стандарта в стандарт телефонной связи (дискретный электрический сигнал преобразуется в аналоговый сигнал) и обратное преобразование. Основной характеристикой модема является предельная скорость передачи данных. В разных моделях она колеблется в диапазоне от 1200 до 56 000 бит/с.

Проще всего уяснить порядок взаимодействия удаленных компьютеров с помощью телефонной линии связи, когда две абонентские системы с помощью модема (модулятора-демодулятора) соединяются с АТС, обеспечивающей соединение этих абонентов. В данном случае модем выполняет функции станци-

онного оборудования, преобразуя цифровой сигнал ПК в аналоговый, который распространяется по телефонной линии связи (рис. 9.13).

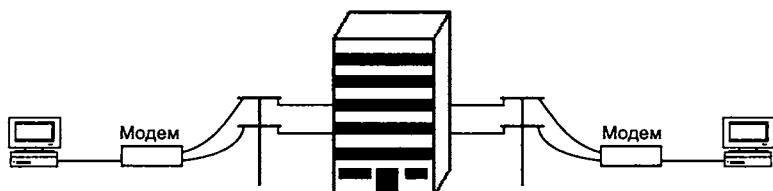


Рис. 9.13. Схема соединения ПК с помощью модема

**Модемная связь.** Наиболее распространенным вариантом сетей сегодня является модемная телефонная сеть на основе стандартной телефонной линии и персонального компьютера. Она позволяет создавать информационно-вычислительные сети практически на неограниченной территории, при этом по указанной сети могут передаваться как данные, так и речевая информация. Основным преимуществом организации информационно-вычислительной сети на основе телефонной линии связи является стандартность и доступность всех ее компонентов, а также охват телефонной сетью, хотя и неравномерный, почти всего земного шара. Количество абонентов такой информационно-вычислительной сети практически неограниченно. Для соединения компьютера с телефонной сетью используется специальная плата, называемая модемом, а также соответствующее программное обеспечение.

Схема взаимодействия компьютеров при использовании линий связи другой природы (радио-, радиорелейной, спутниковой или оптоволоконной) отличается только принципом действия стационарного оборудования и средств коммутации, установленными на узле, к которому подключена данная абонентская система.

**Спутниковая связь.** Линия связи через спутниковый транслятор обладает большой пропускной способностью, перекрывает огромные расстояния, передает информацию с высокой надежностью вследствие низкого уровня помех. Эти достоинства делают спутниковую связь уникальным и эффективным средством передачи информации. Почти весь объем пересылаемой информации спутниковой связи приходится на геостационарные спут-

ники. Но спутниковая связь весьма дорога, так как необходимо иметь наземные станции, антенны, собственно спутник, а также на спутнике необходимо иметь корректирующие двигатели и соответствующие системы управления для удержания его на орбите. В общем балансе связи на спутниковые системы пока приходится примерно 3 % мирового трафика. Но потребности в спутниковых линиях продолжают расти, поскольку при дальности свыше 800 км спутниковые каналы становятся экономически более выгодными по сравнению с другими видами дальней связи.

**Оптоволоконная связь.** Благодаря огромной пропускной способности и помехозащищенности, оптический кабель становится незаменимым в информационно-вычислительных сетях, где требуется передавать большие объемы информации с исключительно высокой надежностью: в местных телевизионных сетях и локальных вычислительных сетях. Быстрые темпы развития производства световодов и соответствующей аппаратуры ведут к снижению стоимости оптических линий связи и к их более широкому распространению.

**Радиосвязь.** К сожалению, радио как беспроводный вид связи несвободно от недостатков. Атмосферные и промышленные помехи, взаимное влияние радиостанций, замирание на коротких волнах, высокая стоимость специальной аппаратуры — все это затрудняет использование радиосвязи в ИВС.

**Радиорелейная связь.** Освоение диапазона ультракоротких волн позволило создать радиорелейные линии. Недостатком радиорелейных линий связи является необходимость установки через определенные промежутки ретрансляционных станций, их обслуживание и т. д.

Использование различных видов связи позволяет создавать сети, существенно различающиеся своей пространственной протяженностью. Поэтому чаще всего деление сети на типы осуществляют по величине территории, которую покрывает сеть. По этому признаку различают локальные и глобальные сети.

К локальным сетям — Local Area Networks (LAN) — относят сети компьютеров, сосредоточенные на небольшой территории (обычно в радиусе не более 1—2 км). В общем случае локальная сеть представляет собой коммуникационную систему, принадлежащую одной организации. Для этих сетей характерна небольшая протяженность линий связи, что позволяет:

- использовать относительно дорогие высококачественные линии связи;



- достигать высоких скоростей обмена данными порядка 100 Мбит/с, применяя простые методы передачи данных;
- получать разнообразные услуги в режиме непосредственного общения (on-line).

Глобальные сети — Wide Area Networks (WAN) — объединяют территориально рассредоточенные компьютеры, которые могут находиться в различных городах и странах. Так как прокладка высококачественных линий связи на большие расстояния обходится очень дорого, в глобальных сетях часто используются уже существующие линии связи, изначально предназначенные совсем для других целей. Например, многие глобальные сети строятся на основе телефонных и телеграфных каналов общего назначения. Из-за низких скоростей таких линий связи в глобальных сетях (десятки килобит в секунду) набор предоставляемых услуг обычно ограничивается передачей файлов, преимущественно не в оперативном, а в фоновом режиме, с использованием электронной почты. Для устойчивой передачи дискретных данных по некачественным линиям связи применяются методы и оборудование, существенно отличающиеся от методов и оборудования, характерных для локальных сетей. Как правило, здесь применяются сложные процедуры контроля и восстановления данных, так как наиболее типичный режим передачи данных по территориальному каналу связи связан со значительными искажениями сигналов.

Практика применения персональных компьютеров в различных отраслях науки, техники и производства показала, что наибольшую эффективность от внедрения вычислительной техники обеспечивают не отдельные автономные ПК, а локальные вычислительные сети. Вследствие этого они приобрели широкое распространение. Поэтому в дальнейшем рассмотрим более подробно принципы построения локальных сетей.

## **9.2. Локальные вычислительные сети**

### **9.2.1. Топология сети и ее виды**

Вычислительные машины, входящие в состав ЛВС, могут располагаться на территории, где создается вычислительная сеть, случайным образом. В то же время, способ обращения к пере-

дающей среде и методы управления сетью зависят от расположения абонентских ЭВМ. Поэтому при создании сети в первую очередь необходимо выбрать способ организации физических связей, т. е. *топологию*. Под топологией вычислительной сети понимается конфигурация графа, вершинам которого соответствуют устройства, непосредственно подключенные к коммуникационной среде сети (компьютеры или другое оборудование, например концентраторы), а ребрам — физические связи между ними. Компьютеры, подключенные к сети, часто называют станциями или узлами сети. При этом следует понимать, что конфигурация физических связей определяется электрическими соединениями компьютеров между собой и может отличаться от конфигурации логических связей между узлами сети. Логические связи представляют собой маршруты передачи данных между узлами сети и образуются путем соответствующей настройки коммуникационного оборудования.

Выбор топологии электрических связей существенно влияет на многие характеристики сети. Например, наличие резервных связей повышает надежность сети и делает возможным балансирование загрузки отдельных каналов. Простота присоединения новых узлов, свойственная некоторым топологиям, делает сеть легко расширяемой. Экономические соображения часто приводят к выбору топологий, для которых характерна минимальная суммарная длина линий связи. Рассмотрим наиболее распространенные виды топологий.

#### 9.2.1.1. Полносвязная топология

Такая топология соответствует сети, в которой каждый компьютер сети связан со всеми остальными (рис. 9.14). Несмотря на логическую простоту, этот вариант оказывается громоздким и неэффективным. Действительно, при таком способе соединения каждый компьютер в сети должен иметь большое количество коммуникационных портов, чтобы обеспечить связь с каждым из остальных компьютеров сети. Кроме того, для каждой пары компьютеров должна быть выделена отдельная проводная линия связи. Эти требования приводят к тому, что данный вид топологии используется очень редко. Его применение целесообразно только в многомашинных комплексах или глобальных сетях при небольшом количестве компьютеров.

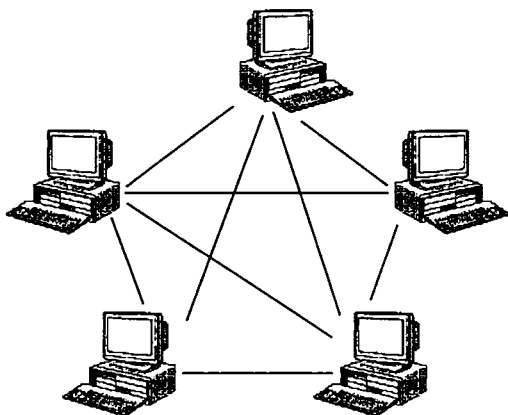


Рис. 9.14. Пример полностью связанной топологии

### 9.2.1.2. Кольцевая топология

В сетях с кольцевой конфигурацией (рис. 9.15) данные передаются по кольцу от одного компьютера к другому, как правило, в одном направлении. Если компьютер распознает данные как «свои», то он копирует их себе во внутренний буфер. В сети с кольцевой топологией необходимо принимать специальные меры, чтобы в случае выхода из строя или отключения какой-либо станции не прервался канал связи между остальными станциями. Кольцо представляет собой очень удобную конфигу-

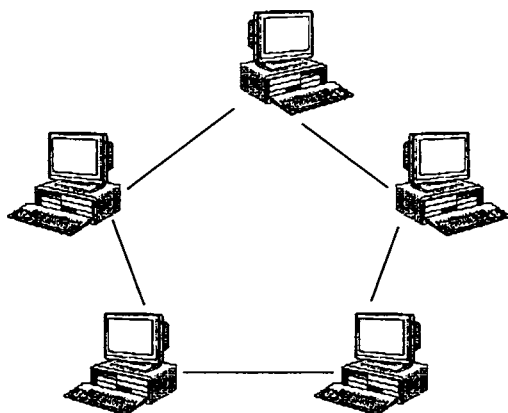


Рис. 9.15. Топология «кольцо»

рацию для организации обратной связи — данные, сделав полный оборот, возвращаются к узлу-источнику. Поэтому этот узел может контролировать процесс доставки данных адресату. Часто это свойство кольца используется для тестирования связности сети и поиска узла, работающего некорректно. Для этого в сеть посылаются специальные тестовые сообщения.

При реализации данной топологии выход одного узла сети соединяется с входом другого. Информация по кольцу передается от узла к узлу. Каждый промежуточный узел между передатчиком и приемником ретранслирует посланное сообщение. Принимающий узел распознает и получает только адресованные ему сообщения.

Кольцевая топология является идеальной для сетей, занимающих сравнительно небольшое пространство. В ней отсутствует центральный узел, что повышает надежность сети. Ретрансляция информации позволяет использовать в качестве передающей среды любые типы кабелей. Последовательная дисциплина обслуживания узлов такой сети снижает ее быстродействие, а выход из строя одного из узлов нарушает целостность кольца и требует принятия специальных мер для сохранения тракта передачи информации.

### 9.2.1.3. Топология «общая шина»

Это один из наиболее простых видов топологий. Общая шина (рис. 9.16) является очень распространенной топологией для локальных сетей. В этом случае компьютеры подключаются к одному коаксиальному кабелю, именуемому магистралью или сегментом. К магистрали подключены все компьютеры сети. Передаваемая информация может распространяться в обе стороны. Применение общей шины снижает стоимость проводки, унифицирует подключение различных модулей, обеспечивает возможность почти мгновенного широковещательного обращения ко

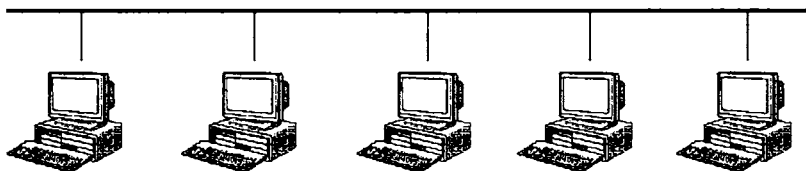


Рис. 9.16. Топология «общая шина»

всем станциям сети. Таким образом, основными преимуществами такой схемы являются дешевизна и простота разводки кабеля по помещениям. Сеть легко наращивать и конфигурировать, а также адаптировать к различным системам. Сеть шинной топологии устойчива к возможным неисправностям отдельных узлов. Однако самый серьезный недостаток общей шины заключается в ее низкой надежности: любой дефект кабеля или какого-нибудь из многочисленных разъемов полностью парализует всю сеть. Другим недостатком общей шины является ее невысокая производительность, так как при таком способе подключения в каждый момент времени только один компьютер может передавать данные в сеть. Поэтому пропускная способность канала связи всегда делится здесь между всеми узлами сети.

В сети с топологией «шина» компьютеры адресуют данные конкретному компьютеру, передавая их по кабелю в виде электрических сигналов. В этом случае сообщение, отправляемое одной рабочей станцией, принимается одновременно всеми остальными, подключенными к общей шине. Но сообщение включает в себя адрес станции назначения и адрес станции отправителя. Та станция, которой предназначено сообщение, принимает его, остальные игнорируют. Промежуточные узлы не транслируют поступающих сообщений. Информация поступает на все узлы, но принимает сообщение только тот, которому оно адресовано. Следует отметить, что они имеют малую протяженность и не позволяют использовать различные типы кабеля в пределах одной сети.

#### **9.2.1.4. Топология «звезда»**

В этом случае каждый компьютер подключается отдельным кабелем к общему устройству, называемому концентратором, который находится в центре сети (рис. 9.17). В функции концентратора входит направление передаваемой компьютером информации одному или всем остальным компьютерам сети, т. е. концентратор ретранслирует, переключает и маршрутизирует информационные потоки в сети. Главное преимущество этой топологии перед общей шиной — высокая надежность. Вывести из строя всю сеть может только неисправность концентратора. Кроме того, концентратор может играть роль фильтра информации, поступающей от узлов в сеть, и при необходимости блокировать запрещенные администратором передачи.

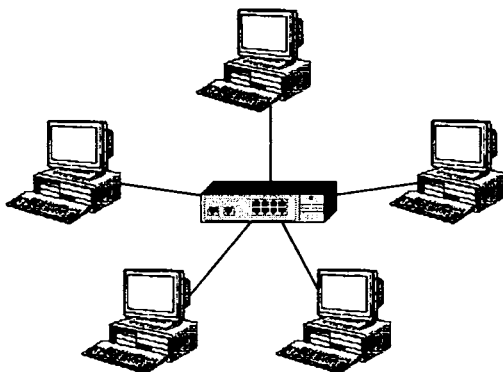


Рис. 9.17. Топология «звезда»

Звездообразная топология значительно упрощает взаимодействие узлов ЛВС друг с другом, позволяет использовать более простые сетевые адаптеры. В то же время работоспособность ЛВС со звездообразной топологией целиком зависит от центрального узла.

К недостаткам топологии типа звезда относится более высокая стоимость сетевого оборудования из-за необходимости приобретения концентратора. Кроме того, возможности по наращиванию количества узлов в сети ограничиваются количеством портов концентратора. Иногда возникает необходимость строить сеть с использованием нескольких концентраторов, иерархически соединенных между собой связями типа звезда [10]. В настоящее время иерархическая звезда является самым распространенным типом топологии связей как в локальных, так и глобальных сетях.

В реальных вычислительных сетях могут использоваться более сложные топологии, представляющие в некоторых случаях сочетания рассмотренных. Выбор той или иной топологии определяется областью применения ЛВС, географическим расположением ее узлов и размерностью сети в целом.

### 9.2.2. Способы обработки данных в сетях

Объединение компьютеров с помощью сетей позволяет использовать распределенную обработку данных. Дело в том, что до того, как были созданы компьютерные сети, использовался

метод централизованной обработки данных. Сущность этого метода иллюстрируется рис. 9.18.

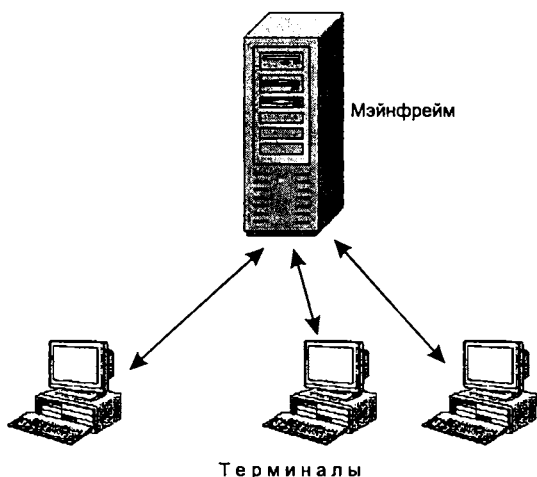


Рис. 9.18. Централизованная обработка данных

В этом случае данные хранятся и обрабатываются центральной ЭВМ, обладающей значительными ресурсами. Иногда для обозначения таких ЭВМ используется термин «мэйнфрейм». Терминалы пользователей позволяют управлять этим процессом и видеть результаты решения задач. Однако этот метод имеет ряд существенных недостатков, основными из которых являются:

- недостаточная надежность процесса обработки данных, так как даже кратковременный выход из строя центральной ЭВМ приводит к роковым последствиям для системы в целом;
- большие затраты времени при диалоговой обработке данных в многопользовательском режиме.

Появление малых ЭВМ, микроЭВМ и, наконец, персональных компьютеров потребовало нового подхода к организации систем обработки данных, который был реализован при распределенной обработке данных. При распределенной обработке данные могут храниться и обрабатываться любым из компьютеров, включенных в сеть (рис. 9.19).

Распределенная обработка данных — обработка данных, выполняемая на независимых, но связанных между собой компьютерах, представляющих распределенную систему. Для реализа-

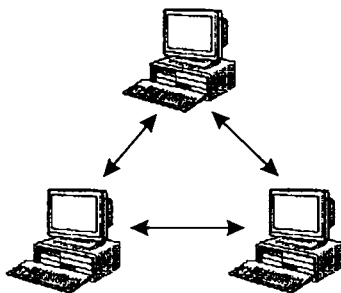


Рис. 9.19. Распределенная обработка данных

ции распределенной обработки данных были созданы многомашинные ассоциации, структура которых разрабатывается по одному из следующих направлений:

- многомашинные вычислительные комплексы (МВК);
- компьютерные (вычислительные) сети.

При этом различные машины, включенные в такую ассоциацию, могут выполнять различные задачи обработки данных. При этом обработка данных может быть распределена между двумя объектами: клиентом и сервером. Клиент — задача, рабочая станция или пользователь компьютерной сети. В процессе обработки данных клиент может сформировать запрос на сервер для выполнения сложных процедур, чтение файла, поиск информации в базе данных и т. д. Сервер, определенный ранее, выполняет запрос, поступивший от клиента. Результаты выполнения запроса передаются клиенту. Сервер обеспечивает хранение данных общего пользования, организует доступ к этим данным и передает данные клиенту. Клиент обрабатывает полученные данные и представляет результаты обработки в виде, удобном для пользователя. В принципе обработка данных может быть выполнена и на сервере. Для подобных систем приняты термины — системы клиент — сервер или архитектура клиент — сервер.

В настоящее время сервером принято называть компьютер, подключенный к сети и обеспечивающий ее пользователей определенными услугами. Серверы могут осуществлять различные функции в зависимости от потребности пользователей сети. Таким образом, сервер является источником ресурсов сети.

Рабочая станция — персональный компьютер, подключенный к сети, через который пользователь получает доступ к ее ресурсам. Рабочая станция сети функционирует как в сетевом, так



и в локальном режиме. Она оснащена собственной операционной системой (MS DOS, Windows и т. д.) и обеспечивает пользователя всеми необходимыми инструментами для решения прикладных задач.

#### **9.2.2.1. Одноранговая сеть**

ЛВС может быть создана таким образом, что в ней не будет единого центра управления взаимодействием рабочих станций и единого устройства для хранения данных. При этом сетевая операционная система должна быть распределена по всем рабочим станциям. Каждая станция сети может обслуживать запросы от других рабочих станций и направлять свои запросы на обслуживание в сеть. Пользователю сети доступны все устройства, подключенные к другим станциям (диски, принтеры). Такие сети называются одноранговыми. В такой сети нет единого центра управления взаимодействием рабочих станций и нет единого устройства для хранения данных. Сетевая операционная система распределена по всем рабочим станциям. Каждая станция сети 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 сети доступны все устройства, подключенные к другим станциям (диски, принтеры).

Достоинства одноранговых сетей: низкая стоимость и высокая надежность. Их недостатки:

- зависимость эффективности работы сети от количества станций;
- сложность управления сетью;
- сложность обеспечения защиты информации;
- трудности обновления и изменения программного обеспечения станций.

Наибольшей популярностью пользуются одноранговые сети на базе сетевых операционных систем LANtastic, NetWare Lite.

#### **9.2.2.2. Сеть с выделенным сервером**

На один из компьютеров сети может быть возложено выполнение таких функций по обслуживанию рабочих станций как хранение данных, предназначенных для использования всеми рабочими станциями, управление взаимодействием между

рабочими станциями и другие сервисные функции. Такой компьютер называется сервером сети, а сеть в этом случае носит название сети с выделенным сервером. На сервере сети устанавливается сетевая операционная система, к нему подключаются все разделяемые внешние устройства — жесткие диски, принтеры и модемы.

Архитектура клиент — сервер может использоваться как в одноранговых локальных вычислительных сетях, так и в сети с выделенным сервером.

Как правило, взаимодействие между рабочими станциями в сети осуществляется через сервер. Логическая организация такой сети может быть представлена топологией «звезда». Роль центрального устройства выполняет сервер. 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).

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.

В зависимости от того, какой ресурс сервера является разделяемым, он называется файл-сервером, факс-сервером, принт-сервером, сервером приложений и т. д.

### 9.2.3. Особенности сетевых операционных систем

Очевидно, что на выделенных серверах желательно устанавливать операционные системы, специально оптимизированные для выполнения тех или иных серверных функций. Поэтому в сетях с выделенными серверами чаще всего используются сетевые операционные системы, в состав которых входит несколько вариантов операционных систем, различающихся возможностя-

ми серверных частей. Например, сетевая операционная система Novell NetWare имеет серверный вариант, оптимизированный для работы в качестве файл-сервера.

В сетевой операционной системе отдельной машины можно выделить несколько частей:

- средства управления локальными ресурсами компьютера: функции распределения оперативной памяти между процессами, планирования и диспетчеризации процессов, управления процессорами в мультипроцессорных машинах, управления периферийными устройствами и другие функции управления ресурсами локальных операционных систем;
- средства предоставления собственных ресурсов и услуг в общее пользование — серверная часть операционной системы (сервер). Эти средства обеспечивают, например, блокировку файлов и записей, что необходимо для их совместного использования; ведение справочников имен сетевых ресурсов; обработку запросов удаленного доступа к собственной файловой системе и базе данных; управление очередями запросов удаленных пользователей к своим периферийным устройствам;
- средства запроса доступа к удаленным ресурсам и услугам и их использования — клиентская часть операционной системы — редиректор. Эта часть выполняет распознавание и перенаправление в сеть запросов к удаленным ресурсам от приложений и пользователей, при этом запрос поступает от приложения в локальной форме, а передается в сеть в другой форме, соответствующей требованиям сервера. Клиентская часть также осуществляет прием ответов от серверов и преобразование их в локальный формат, так что для приложения выполнение локальных и удаленных запросов неразлично;
- коммуникационные средства операционной системы, с помощью которых происходит обмен сообщениями в сети. Эта часть обеспечивает адресацию и буферизацию сообщений, выбор маршрута передачи сообщения по сети, надежность передачи и т. п., т. е. является средством транспортировки сообщений.

В зависимости от функций, возлагаемых на конкретный компьютер, в его операционной системе может отсутствовать либо клиентская, либо серверная части.

Очевидно, что на выделенных серверах желательно устанавливать операционные системы, специально оптимизированные для выполнения тех или иных серверных функций. Поэтому в сетях с выделенными серверами чаще всего используются сетевые операционные системы, в состав которых входит несколько вариантов операционных систем, различающихся возможностями серверных частей. Например, сетевая операционная система Novell NetWare имеет серверный вариант, оптимизированный для работы в качестве файл-сервера, а также варианты оболочек для рабочих станций с различными локальными операционными системами, причем эти оболочки выполняют исключительно функции клиента. Другим примером операционной системы, ориентированной на построение сети с выделенным сервером, является операционная система Windows NT. В отличие от NetWare, оба варианта данной сетевой операционной системы — Windows NT Server (для выделенного сервера) и Windows NT Workstation (для рабочей станции) — могут поддерживать функции и клиента, и сервера. Но серверный вариант Windows NT имеет больше возможностей для предоставления ресурсов своего компьютера другим пользователям сети, так как может выполнять более широкий набор функций, поддерживает большее количество одновременных соединений с клиентами, реализует централизованное управление сетью, имеет более развитые средства защиты.

Выделенный сервер не принято использовать в качестве компьютера для выполнения текущих задач, не связанных с его основным назначением, так как это может уменьшить производительность его работы как сервера. В связи с такими соображениями в ОС Novell NetWare на серверной части возможность выполнения обычных прикладных программ вообще не предусмотрена, т. е. сервер не содержит клиентской части, а на рабочих станциях отсутствуют серверные компоненты. Однако в других сетевых операционных системах функционирование на выделенном сервере клиентской части вполне возможно. Например, под управлением Windows NT Server могут запускаться обычные программы локального пользователя, которые могут потребовать выполнения клиентских функций операционной системы при появлении запросов к ресурсам других компьютеров сети. При этом рабочие станции, на которых установлена операционная система Windows NT Workstation, могут выполнять функции невыделенного сервера.

Для решения проблем ЛВС фирма Microsoft разработала интегрированный пакет Microsoft BackOffice. Концепция BackOffice заключается в следующем: если интегрированный пакет Microsoft Office — это Front Office, или офис, «повернутый лицом» к конечному пользователю, то Microsoft BackOffice — это его обратная, скрытая сторона, невидная пользователю, но обеспечивающая ему комфортные условия работы. При этом пользователь может думать о том, что именно хочет сделать у себя на рабочем месте, в локальной или глобальной сети, а не о том, как именно это сделать.

В пакет Microsoft BackOffice включены несколько программ-серверов:

- Microsoft Windows NT Server 4.0 — операционная система, обеспечивающая работу в сети;
- Microsoft Windows NT Server 4.0 Terminal Server Edition — операционная система, позволяющая работать с Windows-терминалами;
- Microsoft SQL Server 7.0 — масштабируемая СУБД, работающая в среде «клиент — сервер»;
- Microsoft Exchange Server 5.5 — платформа для коллективной работы и обмена сообщениями;
- Microsoft Site Server 3.0 — набор приложений для создания web-узла и управления им;
- Microsoft Proxy Server 2.0 — шлюз между Интернет и локальной сетью;
- Microsoft SNA Server 4.0 — система для взаимодействия с мэйнфреймами.

В совокупности эти программы образуют систему управления сетью масштаба предприятия. Следовательно, можно утверждать, что Microsoft BackOffice — это комплекс программ, который позволяет интегрировать множество приложений, необходимых для организации работы учреждения. На базе Microsoft BackOffice можно организовать и обработку баз данных, и выполнение Интернет/Интранет-приложений, и почтовую систему.

Варианты поставки пакета могут различаться, например, Microsoft BackOffice Enterprise Edition для крупных предприятий и Microsoft BackOffice Small Business Edition для предприятий малого бизнеса.

Основу системы BackOffice составляет Windows NT Server, который может выступать как файл-сервер, сервер приложений, сервер печати, сервер домена, сервер удаленного доступа, сервер

Интернет, сервер обеспечения безопасности данных, сервер резервирования данных, сервер связи и сервер вспомогательных служб.

Особое внимание следует уделить одному из типов серверов — файловому серверу (File Server). В распространенной терминологии для него принято сокращенное название — файл-сервер. Файл-сервер хранит данные пользователей сети и обеспечивает им доступ к этим данным. Как правило, это компьютер с большой емкостью оперативной памяти и жесткими дисками большой емкости. Он работает под управлением специальной операционной системы, которая обеспечивает одновременный доступ пользователей сети к расположенным на нем данным.

Файл-сервер выполняет следующие функции: хранение данных, архивирование данных, синхронизацию изменений данных различными пользователями, передачу данных.

*Windows NT Server как файл-сервер* обеспечивает создание централизованного хранилища большого количества файлов, коллективно используемых в офисе, или файлов, хранение которых на локальном компьютере нецелесообразно. Для организации файл-сервера не требуется специальной подготовки типа монтирования томов. Все файловые ресурсы независимо от того, на каком уровне они расположены (жестком диске или CD-ROM), сразу могут быть предоставлены для совместного использования.

*Windows NT Server как сервер приложений* приспособлен для работы в системах клиент — сервер в качестве высокопроизводительного компьютера, выполняющего запросы клиентских станций. Происходящий переход от больших мэйнфреймов на современные системы на базе персональных компьютеров (ПК) требует именно такого подхода.

Основными приложениями, выполняемыми в интересах клиентских станций, являются системы управления базами данных, системы информационного обмена, системы управления. Именно поэтому в Microsoft BackOffice входят Microsoft SQL Server — сервер баз данных, Microsoft System Management Server — сервер управления системой, Microsoft Exchange — сервер информационного обмена, Microsoft Internet Information Server — сервер для Интернет (сервер Web, FTP и Gopher) и интрасетей и SNA Server — сервер связи с мэйнфреймами.

Кроме приложений корпорации Microsoft, существует более 2000 разработок других фирм: серверы баз данных (Informix,

Oracle, IBM и т. д.), системы управления сетями (HP, DEC), управления производством (SAP), документооборота (Lotus, Saros), финансовые (Platinum) и другие системы для бизнеса.

*Windows NT Server как сервер печати* дает возможность организовать выполнение запросов пользователей сети на печатные работы, предоставляя в совместное пользование неограниченное число принтеров. Они могут быть подключены локально или по сети с помощью протоколов TCP/IP и DLC. Особенность сервера печати — возможность обслуживания рабочих станций с компьютерами, управляемыми MS DOS, Windows for Workgroup, Windows 95, OS/2, Macintosh, Windows NT Workstation.

*Windows NT Server как сервер домена* обеспечивает централизованное управление и защиту в больших сетях с несколькими серверами. Для Windows NT домен — это произвольная группа серверов, использующих единую базу учетных записей пользователей и политику защиты. Как минимум один из этих серверов назначается контроллером домена. На нем хранится центральная база данных учетных записей всего домена. Такая схема повышает эффективность администрирования и удобна для пользователей сети, поскольку для каждого из них создается только одна учетная запись, которая определяет права доступа к любому серверу домена. Благодаря возможности Windows NT Server устанавливать доверительные отношения между доменами, права доступа, определенные одной пользовательской учетной записью, могут распространяться на множество доменов.

*Windows NT Server как сервер удаленного доступа* позволяет пользователю рабочей станции, связанной сетью через сервер удаленного доступа, чувствовать себя работающим непосредственно в сети. Такой пользователь имеет доступ к файлам и данным в сети, может печатать документы, подключаться к хостам через SNA Server и обмениваться с коллегами сообщениями по электронной почте.

Служба удаленного доступа (Remote Access Service — RAS) состоит из двух частей:

- серверной, устанавливаемой на компьютере с Windows NT Server;
- клиентской, устанавливаемой на рабочей станции.

RAS применяется и для связи территориально удаленных филиалов предприятий. Одновременно через протоколы PPP и SLIP поддерживается до 256 сессий удаленного доступа. Набор протоколов Point-to-Point Protocol (PPP) позволяет осуществлять

удаленный доступ в условиях разнородной сети. Возможно также использование средств удаленного доступа сторонних производителей (NetWare Connect, Shiva Landrover и др.).

RAS в Windows NT Server поддерживает любую комбинацию протоколов TCP/IP, IPX/SPX, NetBEUI при удаленном доступе.

*Windows NT Server как сервер обеспечения безопасности данных* предназначен для защиты информации от несанкционированного доступа и контроля за доступом. Гарантируется обеспечение уровня безопасности C2 в соответствии со стандартами США. Данный уровень подразумевает полный контроль доступа к информации на локальном компьютере, осуществляемый как непосредственно с консоли, так и по сети. При этом обеспечивается возможность определения различных прав на доступ и регистрации любых видов доступа (или попыток доступа) к защищенной информации или ресурсам компьютера. Уровень безопасности C2, однако, не предусматривает защиту передаваемой по сети информации и не гарантирует защищенности информации от физического доступа.

Это означает, что информация, хранящаяся на диске, а также передаваемая по сети, не кодируется. Кроме того, если жесткий диск перенести физически на другой компьютер и там посмотреть на физическом уровне его содержимое, то информация будет доступна. Однако можно воспользоваться средствами физической защиты жестких дисков и запретить загрузку иной операционной системы, что обеспечит полную конфиденциальность информации, находящейся на локальных дисках. Для защиты информации, передаваемой по каналам связи, необходимо использовать различные методы кодирования.

Наиболее известной системой кодирования для сетей Windows NT является продукт Trusted Network Technology (TNT) фирмы Blue Ridge Software Inc., обеспечивающий средства защиты и страховки уровня TCSEC B1 при работе в сетях, использующих протокол TCP/IP. Продукт позволяет рабочим станциям Windows NT работать в условиях доверительной сети. Администратор сети назначает каждой рабочей станции в доверительной сети различные уровни чувствительности. После этого TNT обрабатывает всю вводимую, хранящуюся или передаваемую информацию в соответствии с назначенным уровнем. Этот уровень ассоциируется с исходящими датаграммами. На приемной стороне TNT препятствует просмотру входящих сетевых пакетов программами, уровень чувствительности которых выходит за



уровень, назначенный для данной станции. Кроме этого, все не охваченные доверительными отношениями станции группируются и отмечаются в системе TNT таким образом, что могут работать в сети только на этом уровне.

*Windows NT Server как сервер резервирования данных* обеспечивает возможность резервного копирования файлов на магнитную ленту. Администратор системы определяет пользователя, ответственного за эту операцию, и только он регулярно выполняет копирование данных на стример. При необходимости эту операцию можно автоматизировать.

*Windows NT Server как сервер связи* предназначен для соединения между собой различных сегментов сети. При этом предусмотрены возможности сопряжения разнородных сетей, в первую очередь с самой распространенной сетью — NetWare. В состав Windows NT Server входит Microsoft Service Compatible Workstation for NetWare, который позволяет осуществлять взаимодействие с сервером Novell NetWare, предоставлять возможность доступа к файлам и принтерам.

Для многих задач использование одного файл-сервера оказывается недостаточным. Тогда в сеть могут включаться несколько серверов. В результате локальная вычислительная сеть, предназначенная для обслуживания учреждения, может принять вид, приведенный на рис. 9.20.

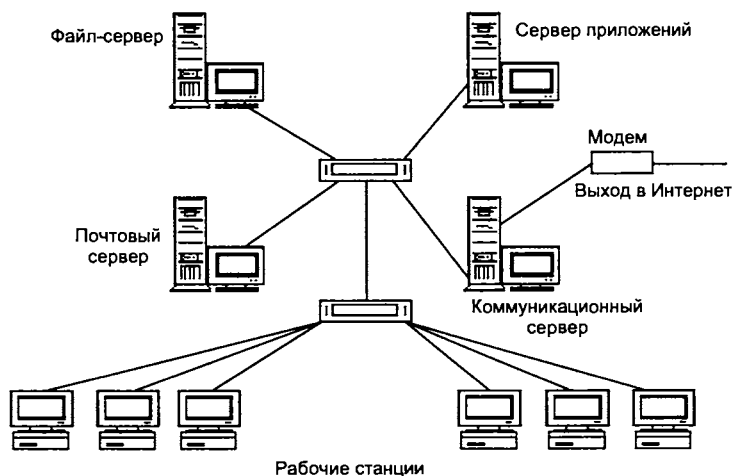


Рис. 9.20. Вариант ЛВС учреждения

#### 9.2.4. Способы объединения ЛВС

По мере эксплуатации ЛВС может возникнуть проблема расширения ее функциональных возможностей, например необходимость объединения ЛВС, различных отделов и филиалов одного и того же учреждения. Расширение конфигурации сети может осуществляться как в пределах ограниченного пространства, так и с выходом во внешнюю среду. Стремление получить выход на определенные информационные ресурсы может потребовать подключения ЛВС к сетям более высокого уровня.

Известны несколько способов объединения ЛВС. Самым простым из них является мост — объединение одинаковых сетей в пределах ограниченного пространства. В этом случае в пределах допустимой длины строится отрезок сети — сетевой сегмент. Для объединения сетевых сегментов используются мосты — устройства, соединяющие две сети, использующие одинаковые методы передачи данных. Сети, которые объединяет мост, должны иметь одинаковые сетевые уровни модели взаимодействия открытых систем, нижние уровни могут иметь некоторые отличия.

Ограничения, связанные с применением мостов и коммутаторов, привели к тому, что в ряду коммуникационных устройств появился еще один тип оборудования — маршрутизатор (router). Маршрутизаторы более надежно и более эффективно, чем мосты, изолируют трафик отдельных частей сети друг от друга. Маршрутизаторы образуют логические сегменты посредством явной адресации. В этом случае в адресах имеется поле номера сети, так что все компьютеры, у которых значение этого поля одинаково, принадлежат к одному сегменту, называемому в данном случае подсетью (subnet).

Маршрутизатор выполняет свои функции на сетевом уровне, поэтому он зависит от протоколов обмена данными, но не зависит от типа сети. С помощью двух адресов — адреса сети и адреса узла маршрутизатор однозначно выбирает определенную станцию сети. Маршрутизатор также может выбрать наилучший путь для передачи сообщения абоненту сети, фильтрует информацию, проходящую через него, направляя в одну из сетей только ту информацию, которая ей адресована. Кроме того, маршрутизатор обеспечивает балансировку нагрузки в сети, перенаправляя потоки сообщений по свободным каналам связи.

Для объединения ЛВС совершенно различных типов, работающих по существенно отличающимся друг от друга протоко-

лам, предусмотрены специальные устройства — шлюзы, позволяющие организовать обмен данными между двумя сетями, использующими различные протоколы взаимодействия.

Шлюз осуществляет свои функции на уровнях выше сетевого. Он не зависит от используемой передающей среды, но зависит от используемых протоколов обмена данными. Обычно шлюз выполняет преобразование между двумя протоколами. С помощью шлюзов можно подключить локальную вычислительную сеть к главному компьютеру, а также локальную сеть — к глобальной. Конструктивно мосты, маршрутизаторы и шлюзы выполняются в виде плат, устанавливаемых в компьютерах.

Таким образом, компьютерные сети представляют собой сложные системы, для классификации которых могут быть использованы различные признаки. Так, в зависимости от назначения сети могут быть информационными, вычислительными и информационно-вычислительными.

По территориальному расположению абонентских систем сети можно разделить на глобальные, региональные и локальные.

Сети могут иметь топологию «шина», «кольцо», «звезда» или смешанную. Кроме того, сети могут быть одноранговыми или с выделенным сервером.

Для объединения локальных сетей используются мосты, маршрутизаторы или шлюзы, специальные устройства и программы, управляющие ими.

## **Контрольные вопросы**

1. Назначение компьютерных сетей и их классификация.
2. Линии связи, используемые для создания компьютерных сетей, их основные характеристики, достоинства и недостатки.
3. Понятие топологии компьютерной сети. Основные виды топологий.

### 10.1. Основы функционирования Интернет

Интернет представляет собой всемирную информационную компьютерную сеть, которая объединяет в единое целое множество компьютерных сетей, работающих по единым правилам. Интернет не является коммерческой организацией и никому не принадлежит. Пользователи Интернет имеются практически во всех странах мира.

Прообраз сети Интернет был создан в конце шестидесятых годов по заказу Министерства обороны США. В то время существовало не очень много мощных компьютеров, и для проведения научных исследований возникла потребность обеспечить доступ многочисленных ученых к этим компьютерам. При этом Министерство обороны поставило условие, чтобы сеть продолжала работать при уничтожении ее части, поэтому повышенная надежность Интернет была заложена при ее создании.

Днем рождения Интернет можно назвать 2 января 1969 г. В этот день Управление перспективных исследований (ARPA — Advanced Research Projects Agency), являющееся одним из подразделений Министерства обороны США, начало работу над проектом связи компьютеров оборонных организаций. В результате исследований была создана сеть ARPANET, в основе функционирования которой лежали принципы, на которых позже была построена сеть Интернет.

Следующим этапом в развитии Интернет было создание сети Национального научного фонда США (NSF). Сеть, названная NSFNET, объединила научные центры Соединенных Штатов. При этом основой сети стали пять суперкомпьютеров, соединенных между собой высокоскоростными линиями связи. Все ос-

тальные пользователи подключались к сети и могли использовать возможности, предоставляемые этими компьютерами.

Сеть NSFNET быстро заняла место ARPANET, и последняя была ликвидирована в 1990 г. Развитие сети потребовало ее реорганизации, и в 1987 г. был создан NSFNET Backbone [Бэкбон] — базовая часть или хребет сети. В настоящее время используется термин «опорная сеть». В момент создания (1987 г.) опорная сеть состояла из тринадцати центров, соединенных друг с другом высокоскоростными линиями связи. Центры располагались в разных частях США. Таким образом, появилась сеть Интернет в США.

Одновременно были созданы национальные сети в других странах. Компьютерные сети разных стран стали объединяться и в девяностых годах появилась Интернет в ее сегодняшнем виде. Сейчас Интернет объединяет тысячи разных сетей, расположенных по всему миру. К Интернет имеют доступ десятки миллионов пользователей. Рост и развитие Интернет продолжается, следствием чего является значительное повышение роли Интернет во всех информационных технологиях.

Бурный рост числа пользователей Интернет в России начался в 1996 г. Сегодня и в нашей стране Интернет превратилась в инструмент повседневной деятельности. Это можно увидеть и по развитию русской части Интернет. Если несколько лет назад почти вся информация в сети приводилась на английском языке и предназначалась, в основном, для иностранцев, то сегодня в основном поставщики информации ориентируются на отечественных пользователей и в Интернет можно найти самую разнообразную информацию на русском языке.

В последнее время ведутся активные работы по передаче новых видов информации через Интернет. Уже сегодня через Интернет можно слушать радио и просматривать телепередачи. Глобальная сеть позволяет проводить селекторные совещания и видеоконференции. С помощью Интернет многие служащие смогут работать дома, обмениваясь документами со своими коллегами, которые находятся за тысячи километров от них. Постепенно Интернет становится одним из основных средств связи, главным способом получения и передачи информации.

Таким образом, умение использовать Интернет так же, как и умение работать на компьютере, является на сегодняшний день обязательным условием для достижения успехов практически в любой области деятельности.

### 10.1.1. Структура Интернет

Как и любая другая компьютерная сеть, Интернет состоит из множества компьютеров, соединенных между собой линиями связи, и установленных на этих компьютерах программ. К сети могут быть подключены как отдельные компьютеры, так и локальные сети. В последнем случае можно считать, что к Интернет подключены все компьютеры данной локальной сети, хотя линией связи с Интернет соединен только один компьютер. Очевидно, что при объединении локальных сетей могут возникнуть проблемы, вызванные тем, что в различных ЛВС могут быть установлены различные сетевые операционные системы. Эти проблемы решаются путем использования специальных аппаратных и программных средств. В настоящее время для этих целей применяются три вида устройств:

- мосты — объединяют сети разных топологий, но работающие под управлением одноплатных сетевых операционных систем;
- маршрутизаторы — соединяют сети разного типа, но использующие одну операционную систему;
- шлюзы — позволяют организовать обмен данными между двумя сетями, использующими различные протоколы взаимодействия.

Рассмотрим условную схему части Интернет (рис. 10.1). Ее основу составляет опорная сеть, представляющая собой узлы, соединенные между собой линиями связи с высокой помехозащищенностью, пропускной способностью и низким затуханием. Как правило, это оптоволоконные или спутниковые каналы связи. Узел опорной сети представляет собой, как правило, несколько мощных компьютеров, находящихся в состоянии постоянного подключения. Эти компьютеры должны обладать высоким быстродействием и большим объемом как внешней, так и оперативной памяти. Они называются хост-компьютерами (host — хозяин) и работают в круглосуточном режиме, что обеспечивается за счет дублирования. Очевидно, что в идеале желательно, чтобы опорная сеть имела полносвязную топологию. Однако по ряду причин до сих пор достичь этого до настоящего времени не удалось.

С узлами опорной сети соединяются локальные сети или индивидуальные пользователи. Соединение может быть временным или постоянным. Временное соединение устанавливается при

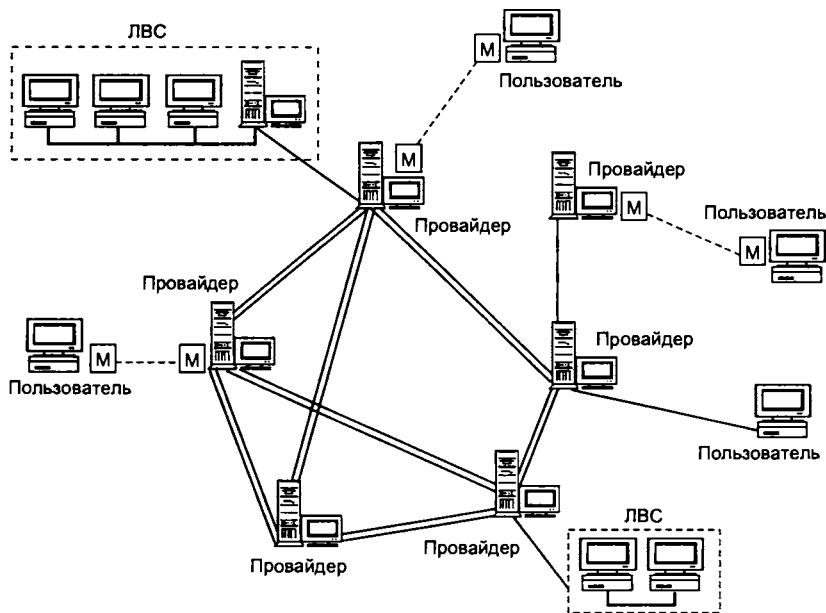


Рис. 10.1. Схема Интернет

использовании телефонных каналов связи. Так как при наборе телефонного номера для установки связи двух абонентов на АТС происходит переключение (коммутация) линий связи, то телефонные линии часто называют коммутируемыми. С помощью коммутируемых линий связь устанавливается только на некоторое время, необходимое для телефонного разговора двух людей или передачи информации между компьютерами. В отличие от коммутируемых линий так называемые выделенные линии связывают компьютеры постоянно, круглый год, 24 часа в сутки. Выделенные линии могут быть созданы с помощью кабелей или радиосвязи и позволяют обмениваться информацией с огромными скоростями.

Организации, обеспечивающие подключение пользователей к Интернет называются поставщиками услуг Интернет — провайдерами. К Интернет могут быть подключены как отдельные компьютеры, так и локальные сети. В последнем случае можно считать, что к Интернет подключены все компьютеры данной локальной сети, хотя линией связи с Интернет соединен только один компьютер. Провайдеры обладают большим количеством

линий для подключения пользователей и высокоскоростные линии для связи с остальной частью Интернет. Часто мелкие провайдеры подключены к более крупным, которые, в свою очередь, подключены к другим поставщикам. Если поставщик подключен непосредственно к опорной сети, то скорость передачи информации будет максимальной.

### 10.1.2. Адресация в сетях

При соединении трех и более компьютеров возникает проблема их адресации. К адресу узла сети и схеме его назначения предъявляется ряд требований:

- адрес должен уникально идентифицировать компьютер в сети любого масштаба;
- схема назначения адресов должна сводить к минимуму ручной труд администратора и вероятность дублирования адресов;
- адрес должен иметь иерархическую структуру, удобную для построения больших сетей. В больших сетях, состоящих из многих тысяч узлов, отсутствие иерархии адреса может привести к большим издержкам — конечным узлам, и коммуникационному оборудованию придется оперировать с таблицами адресов, состоящими из тысяч записей;
- адрес должен быть удобен для пользователей сети, а это значит, что он должен иметь символическое представление, например, [www.mgsu.info](http://www.mgsu.info);
- адрес должен иметь по возможности компактное представление, чтобы не перегружать память коммуникационной аппаратуры — сетевых адаптеров, маршрутизаторов и т. п.

Так как все перечисленные требования трудно совместить в рамках какой-либо одной схемы адресации, то на практике обычно используется сразу несколько схем, так что компьютер одновременно имеет несколько адресов-имен. Каждый адрес используется в той ситуации, когда соответствующий вид адресации наиболее удобен. Для этого используются специальные вспомогательные протоколы, которые по адресу одного типа могут определить адреса других типов.

В настоящее время наибольшее распространение получили три схемы адресации узлов.



**Аппаратные (hardware) адреса.** Эти адреса предназначены для сети небольшого или среднего размера, поэтому они не имеют иерархической структуры. Типичным представителем адреса такого типа является адрес сетевого адаптера локальной сети. Такой адрес обычно используется только аппаратурой, поэтому его стараются сделать по возможности компактным и записывают в виде двоичного или шестнадцатеричного значения, например **0031004a24e5**. При задании аппаратных адресов обычно не требуется выполнение ручной работы, так как они либо встраиваются в аппаратуру компанией-изготовителем, либо генерируются автоматически при каждом новом запуске оборудования, причем уникальность адреса в пределах сети обеспечивает оборудование. Помимо отсутствия иерархии, недостатки использования аппаратных адресов связаны с тем, что:

- при замене аппаратуры, например сетевого адаптера, изменяется и адрес компьютера;
- при установке нескольких сетевых адаптеров у компьютера появляется несколько адресов.

**Символьные адреса или имена.** Эти адреса предназначены для запоминания людьми и поэтому обычно несут смысловую нагрузку. Символьные адреса легко использовать как в небольших, так и крупных сетях. Для работы в больших сетях символьное имя может иметь сложную иерархическую структуру, например **<http://www.ed.gov.ru/edusupp/metodobesp/>**. Этот адрес указывает на то, что данный компьютер поддерживает ftp-архив Федерального агентства по образованию и эта сеть относится к Министерству образования и науки РФ.

**Числовые составные адреса.** Символьные имена удобны для людей, но из-за переменного формата и потенциально большой длины их передача по сети не очень экономична. Поэтому во многих случаях для работы в больших сетях в качестве адресов узлов используют числовые составные адреса. Эти адреса имеют фиксированный и компактный формат. Типичными представителями адресов этого типа являются IP-адреса. В них поддерживается двухуровневая иерархия, адрес делится на старшую часть — номер сети и младшую — номер узла. Такое деление позволяет передавать сообщения между сетями только на основании номера сети, а номер узла используется только после доставки сообщения в нужную сеть; точно так же, как название улицы используется почтальоном только после того, как письмо доставлено в нужный город. В последнее время, чтобы сделать

маршрутизацию в крупных сетях более эффективной, предлагаются более сложные варианты числовой адресации, в соответствии с которыми адрес имеет три и более составляющих.

В современных сетях для адресации узлов применяются, как правило, одновременно все три приведенные ранее схемы. Пользователи адресуют компьютеры символьными именами, которые автоматически заменяются в сообщениях, передаваемых по сети, на числовые номера. С помощью этих числовых номеров сообщения передаются из одной сети в другую, а после доставки сообщения в сеть назначения вместо числового номера используется аппаратный адрес компьютера. Сегодня такая схема характерна даже для небольших автономных сетей, где, казалось бы, она явно избыточна — это делается для того, чтобы при включении этой сети в большую сеть не нужно было менять состав операционной системы.

**Адресация в Интернет.** Каждый хост-компьютер имеет свой постоянный адрес, отличающий его от всех остальных компьютеров в Интернет. Он называется IP-адресом.

IP-адрес любого узла определяется четырехбайтовым числом, состоящим из двух частей. Первая — сетевая является общей для всех узлов выбранной сети, а вторая — хост-часть у каждого узла строго индивидуальна. Для большего удобства все существующие сети разбиты на классы (рис. 10.2).

Класс А имеет 0 в старшем бите адреса. У узлов, входящих в эту категорию, на сетевой адрес отводятся младшие семь бит первого байта, а на хост-часть — 3 байта. Таких сетей может быть только 126 с 16 млн узлов в каждой.



Рис. 10.2. Схема IP-адреса

Сети класса В содержат 10 в двух старших битах адреса, сетевой адрес у них имеет 6 бит первого байта и второй байт целиком, а на хост-часть отводится 2 байта. Таких сетей может быть 16 тыс. по 65 тыс. узлов в каждой.

Сети класса С имеют 110 в трех старших битах адреса, и на их сетевой адрес отводятся младшие 5 бит первого байта, второй и третий байты, тогда как на хост-часть остается 1 байт. Их может быть около 2 млн по 254 узла в каждой.

Каждый байт IP-адреса записывается десятичным числом от 0 до 256, а сами байты разделяются точкой: **192.56.1.45**.

Можно предположить, что такой способ адресации создает иерархическую структуру Интернет (рис. 10.3).

Однако, наличие связей между хост-машинами одного уровня преобразует такую структуру в сетевую. Тем не менее в данной структуре можно выделить группы компьютеров, подключенных к узлу более высокого уровня (старшему узлу). Компьютеры, входящие в такую группу, имеют общую границу администрирования и общую базу учетных записей. В группе реализуется единая политика безопасности. Для обозначения такой группы компьютеров используется термин домен (domain).

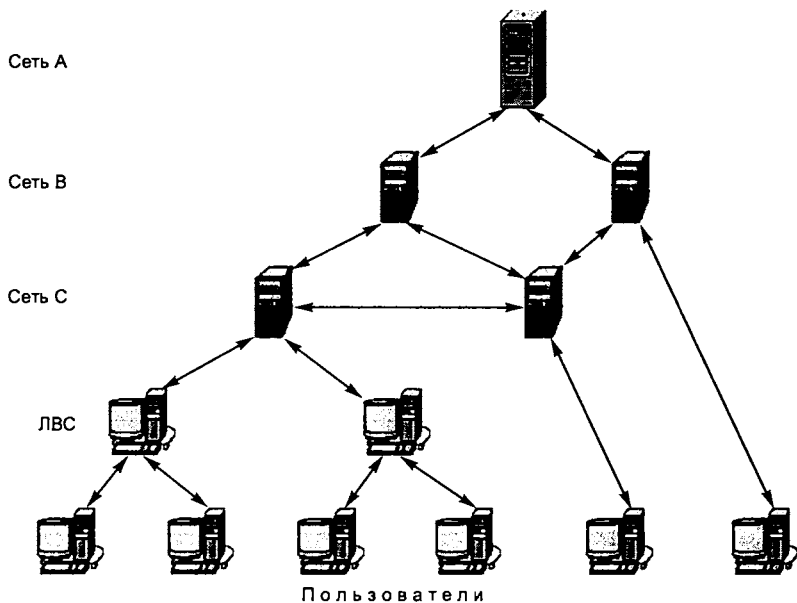


Рис. 10.3. Возможная структура фрагмента сети Интернет

Идентификация узлов Интернет с помощью IP-адресов позволяет создать на каждой хост-машине список адресов подключенных к нему младших узлов. Кроме того, туда вносятся сведения об адресе своего головного узла. В свою очередь, это обстоятельство позволяет реализовать следующий алгоритм поиска компьютера в Интернет:

- при наборе адреса какого-нибудь узла в сети создается запрос;
- запрос отправляется на первичный узел;
- первичный узел ищет заданный IP-адрес в своем списке;
- если поиск удачен, запрос переходит к компьютеру, чей адрес совпадает с тем, который указан в нем, после чего налаживается соединение между абонентами сети. Данная операция происходит независимо от степени удаленности узлов друг от друга;
- при отсутствии заданного IP-адреса в таблице запрос отправляется на узел более высокого уровня;
- узел высшего уровня ищет заданный IP-адрес в своей таблице;
- положительный результат поиска приводит к соединению с заданным компьютером, а отрицательный — к выведению на экран компьютера, создавшего запрос, сообщения об отсутствии в сети машины с указанным вами адресом.

Применение IP-адресов неудобно для человека, а потому с самых ранних времен компьютерам, подключенным к Интернет, стали давать имена, причем таким образом, чтобы каждому имени соответствовал уникальный IP-адрес. Эти имена называются доменными. Доменное имя (domain name) — уникальный идентификатор, который назначается определенному IP-адресу. Доменное имя дает возможность обращаться к компьютеру по имени вида **www.company.com**, что позволяет избавиться от необходимости запоминать его числовой эквивалент.

Таким образом, любой компьютер, подключенный к Интернет, имеет свой уникальный адрес цифровой или IP-адрес (от Internet work Protocol — межсетевой протокол) и доменный. Оба адреса равноправны, но IP-адрес удобен для обработки на ЭВМ, а символьный доменный адрес — для восприятия человеком. Поэтому компьютеры при пересылке информации используют цифровые адреса, а пользователи в работе с Интернет используют, в основном, доменные.

Даже при временном соединении по коммутируемому каналу компьютеру выделяется адрес, однозначно идентифицирующий его. В любой момент времени все компьютеры, подключенные к Интернет, имеют разные адреса. Так же, как почтовый адрес однозначно определяет местонахождение человека, адрес в Интернет однозначно определяет местонахождение компьютера в сети. Таким образом, адреса в Интернете являются важнейшей его частью.

Хотя в Интернет отсутствует единый центр управления, но существуют организации, занимающиеся проверкой и выдачей адресов. Так что ни один пользователь не может самостоятельно присвоить себе адрес по своему усмотрению. В доменной системе имена назначаются путем возложения на различные группы пользователей ответственности за подмножество имен. Каждый уровень в такой системе называется доменом. Для названия доменов верхнего уровня существует специальное соглашение, согласно которому каждой стране присваивается определенное имя. Например, Россия — RU (ru), Австралия — AU (au), Австрия — AT (at), Германия — DE (de), США — US (us). Домены распределяются по иерархическому принципу: получить адрес домена второго уровня можно у организации, владеющей доменом первого. Доменом RU владеет организация РОСНИИРОС (Российский НИИ развития общественных сетей). При данном НИИ организован Технический центр национального домена RU. В свою очередь, при этом центре аккредитованы пять организаций, наделенных правами регистратора (табл. 10.1).

Таблица 10.1. Список организаций — регистраторов Рунета

Наименование организации	Доменный адрес в Интернете
Региональный сетевой информационный центр (RU-center)	www.nic.ru
Гарант-Парк-Телеком	www.r01.ru
Демос-Интернет	www.demos-internet.ru
ООО «СЦС-Совинтел»	www.sovintel.com
Объединение сетей «Фринет»	www.freenet.net

Действия по регистрации доменов в российской части Интернет, которую иногда называют Рунет, регламентируются «Соглашением о порядке регистрирования зоны RU» от 4 декабря 1993 г.

Однако не всегда домен верхнего уровня обозначает страну. Часто встречается трехбуквенный адрес, обозначающий род деятельности. Для коммерческих организаций используется домен **com**, а сетевых организаций — домен **net**. В США не принято указывать название страны, а используются обозначения области деятельности. Кроме доменов **net** и **com** в Соединенных Штатах используются домены: **edu** — для учебных и научных заведений, **gov** — для правительственных учреждений, **mil** — для военных организаций и **org** — для прочих организаций. Кстати, стоимость регистрации зависит от зоны регистрации домена. Эти данные для российских регистраторов представлены в табл. 10.2.

Таблица 10.2. Стоимость регистрации домена в РФ

Зона	Регистрация*	Продление регистрации*
RU	20 у. е. в год	15 у. е. в год
SU	100 у. е. в год	100 у. е. в год
NET, COM (RU-CENTER)	20 у. е. в год	15 у. е. в год

С помощью корпорации DomainPeople ([www.domainpeople.com](http://www.domainpeople.com)) можно зарегистрировать домен и в других зонах [11] (табл. 10.3).

Таблица 10.3. Стоимость регистрации домена в других зонах Интернет

Зона	Регистрация*	Продление регистрации*
NET, COM (DomainPeople)	—	25 у. е. в год
ORG (DomainPeople)	25 у. е. в год	25 у. е. в год
BIZ, INFO (DomainPeople)	50 у. е. за 2 года	50 у. е. за 2 года

Имена доменов в составном доменном имени отделяются друг от друга точками, например [www.mgsu.info](http://www.mgsu.info). В обычных почтовых адресах используется последовательное уточнение страны, города, улицы и дома, в который следует доставить письмо. Также в Интернете доменная система имен использует принцип последовательных уточнений. Домен верхнего уровня располагается в имени правее, а домен нижнего уровня — левее.

В имени может быть любое число доменов, но чаще всего используются имена с количеством доменов от трех до пяти. Каж-

дая группа, имеющая домен, может создавать и изменять адреса, находящиеся под ее контролем. Например, если в фирме с адресом `company.ru` будет создано новое подразделение аналитических исследований, то для ее именованья фирма не должна спрашивать ни у кого разрешения. Достаточно добавить новое имя, например, `analysts`, в описание адресов своего домена. В итоге любой пользователь Интернет может обращаться к этой группе по адресу `analysts.company.ru`.

Рассмотрим принцип образования адреса на примере некоего учреждения (рис. 10.4).

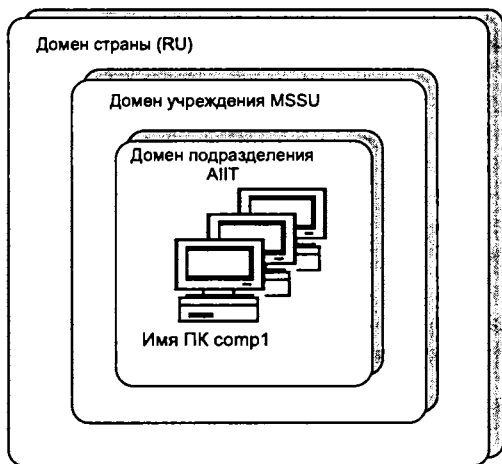


Рис. 10.4. Структура доменного адреса

Для пользователей, включенных в сеть данного учреждения, первым (старшим) доменом будет домен с именем `ru`. В России множество пользователей Интернет, и следующий уровень определяет организацию, которой принадлежит данный адрес. В нашем случае это организация с Интернет-адресом — `MSSU.ru`. Все компьютеры этого учреждения, подключенные к Интернет, объединяются в группы, каждая из которых имеет свой адрес. Так, внутри области администрирования этой организации одному из подразделений выделен свой домен с именем `АИТ`. Его полное имя будет, соответственно, `АИТ.MSSU.ru`. Одному из компьютеров в данном подразделении присвоено имя `comp1`. В результате полный Интернет-адрес этого компьютера будет `comp1.АИТ.MSSU.ru`. Доменная система образования адресов га-

рантирует, что во всей сети Интернет больше не найдется другого компьютера с таким же адресом.

Подавляющее большинство крупных фирм имеет свои страницы во Всемирной паутине. Даже не зная точного адреса, часто можно его угадать. В большинстве случаев адрес состоит из имени **www**, указывающего на тип информации, собственного имени компании и имени **com**. Например, найти во Всемирной паутине фирму Philips можно по адресу **www.philips.com**.

Преобразование имен в цифровые адреса происходит автоматически. При вводе символьного имени, компьютер обращается к серверам DNS, которые хранят информацию о соответствии символьных и цифровых имен. DNS [Дэ Эн Эс] — Domain Name System (Доменная система имен) — это база данных, обеспечивающая преобразование имен компьютеров, подключенных к Интернет, в числовые IP-адреса. DNS-сервер — это программа, обеспечивающая преобразование символьных адресов в цифровые. При подключении к Интернет следует один раз указать цифровой адрес сервера DNS, после чего можно забыть про цифровые адреса и пользоваться исключительно символьными, хотя существует возможность использования цифровых адресов наряду с символьными.

В Интернет, как и в ЛВС, имеется несколько уровней протоколов, взаимодействующих друг с другом. Принципиальной особенностью Интернет является использование на транспортном уровне протокола TCP [Ти Си Пи] — Transmission Control Protocol (Протокол управления передачей), а на сетевом — IP [Ай Пи] — Internet Protocol (Протокол Интернет). Так как эти два протокола тесно взаимосвязаны, то часто их объединяют и говорят, что в Интернет базовым протоколом является TCP/IP [Ти Си Пи Ай Пи]. Все остальные протоколы строятся на основе именно протоколов TCP/IP.

Объяснить работу протоколов TCP/IP можно с помощью достаточно условного примера (рис. 10.5), иллюстрирующего взаимодействие двух компьютеров при передаче информации с одного на другой через Интернет.

В соответствии с протоколом TCP информация делится на пакеты, которые нумеруются для того, чтобы при получении информации ее можно было бы правильно собрать. Далее с помощью протокола IP все части передаются получателю, где с помощью протокола TCP проверяется, все ли части получены. Так как отдельные части могут путешествовать по Интернет самыми



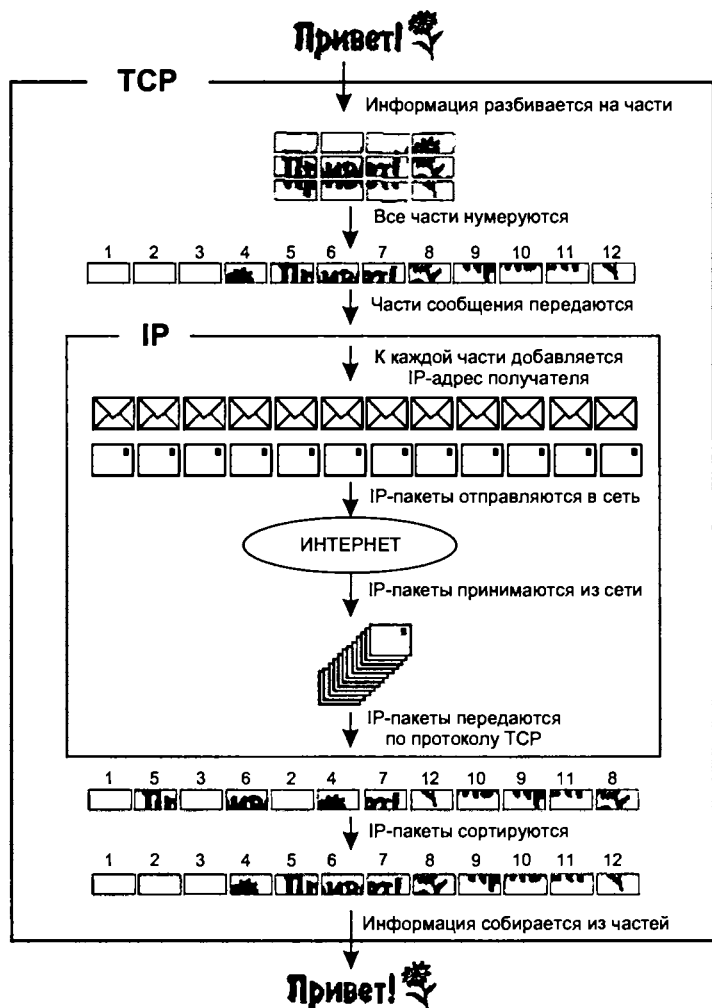


Рис. 10.5. Схема взаимодействия двух компьютеров по протоколу TCP/IP

разными путями, то порядок прихода частей может быть нарушен. После получения всех частей TCP располагает их в нужном порядке и собирает в единое целое. Для протокола TCP не имеет значения, какими путями информация распространяется по Интернет. Этим занимается протокол IP. К каждой полученной порции информации протокол IP добавляет служебную инфор-

мацию, из которой можно узнать адреса отправителя и получателя информации. Если следовать аналогии с почтой, то данные помещаются в конверт или пакет, на котором пишется адрес получателя. Далее протокол IP так же, как и обычная почта, обеспечивает доставку всех пакетов получателю. При этом скорость и пути прохождения разных конвертов могут быть различными. Пути прохождения информации пользователям неизвестны, но правильно оформленные IP-пакеты доходят до получателя.

Таким образом, протоколы TCP/IP обеспечивают передачу информации между двумя компьютерами. При этом в соответствии с протоколом TCP передаваемое сообщение разбивается на TCP-пакеты на отправляющем сервере и восстанавливается в исходном виде на принимающем. Доставка каждого TCP-пакета к месту назначения организуется по протоколу IP. При этом маршруты для разных пакетов, составляющих одно сообщение, могут быть различными. Это объясняется тем, что каждый пакет передается на ближайший свободный в данный момент времени узел.

Отличительной особенностью Интернет является высокая надежность. При выходе из строя части компьютеров и линий связи сеть будет продолжать функционировать. Такая надежность обеспечивается тем, что в Интернете нет единого центра управления. Если выходят из строя некоторые линии связи или компьютеры, то сообщения могут быть переданы по другим линиям связи, так как сетевая структура Интернет всегда обеспечивает несколько путей передачи информации.

## 10.2. Основные службы Интернет

Анализируя основные формы взаимодействия между абонентскими ЭВМ в сети, рассмотренные ранее, можно сделать вывод, что Интернет может обеспечить выполнение двух основных функций:

- 1) быть средством общения между удаленными пользователями;
- 2) быть средством доступа к общим информационным ресурсам, размещенным в Интернет.

Иногда этим функциям присваивают наименование коммуникационной и информационной, соответственно.

Очевидно, что каждая из этих функций может быть реализована с помощью различных средств, что обеспечивает многообразие услуг, предоставляемых пользователям Интернет. Средства обеспечения определенных услуг для пользователей глобальной сети принято называть службами Интернет. Иногда кроме термина «служба» используется термин «сервис». Число различных служб (сервисов) Интернет не остается неизменным: постоянно проявляются новые, в то время как использование некоторых прекращается.

Рассмотрим некоторые самые известные службы, разделив их на коммуникационные и информационные. При этом необходимо помнить, что каждая служба или сервис Сети поддерживается работой определенного узлового сервера и определенного клиента на компьютере пользователя. Работа связки сервер—клиент подчиняется определенному протоколу. Протокол ТСР/ІР является базовым протоколом Интернет, а протоколы служб — прикладными. Иногда их называют протоколами второго уровня.

### 10.2.1. Коммуникационные службы

Очевидно, что может существовать только два режима общения в сети: режим непосредственного общения в реальном масштабе времени, когда пользователи во время общения соединены между собой. Аналогом такого общения является разговор по телефону. Иногда для обозначения такого режима используется термин *on-line*.

Другим режимом является режим отложенного общения (*off-line*). Примером такого общения в быту является отправка письма или телеграммы.

#### 10.2.1.1. Прямое общение в Интернете

**Форумы прямого общения — IRC (Internet Relay Chat).** Эта система позволяет пользователям Интернет и Интранет беседовать в реальном времени. Для получения этой услуги пользователи должны объединиться в каналы, поддерживающие различные темы обсуждения.

Любые символы, введенные через программу IRC, появляются на экранах всех остальных участников вашего канала. Технология использования IRC имеет много общего с работой в

конференциях Usenet. Но если там общение происходит в отложенном режиме, то здесь может вестись живой разговор. Особенность этого общения заключается в том, что текст сообщения вводится с помощью клавиатуры, а затем попадает на общий дисплей. Таким образом, при использовании этой службы общение между участниками происходит в режиме on-line в письменной форме. Подобно телеконференциям участники chat-конференции делятся по тематическим группам.

В настоящее время общение в «чате» используется как своеобразная игра, в которой каждый участник обычно придумывает для себя какой-то «образ» и обыгрывает его. Между участниками «чата» не принято выходить на реальное общение. Впрочем, chat-службой можно воспользоваться и для серьезного общения — как коллективного, так и персонального. Одной из популярных программ, обеспечивающих работу в данном режиме, является ICQ.

**Интернет-телефония.** Эта служба предоставляет возможность голосового общения через Сеть в режиме on-line. Это новая, развивающаяся служба. Ее основное преимущество перед телефоном — низкая цена. Качество Интернет-телефонии пока уступает телефонной связи (задержки во времени, искажение звука), но с течением времени эти недостатки постепенно преодолеваются.

Общий принцип действия телефонных серверов IP-телефонии таков: с одной стороны сервер связан с телефонными линиями и может соединиться с любым телефоном мира. С другой стороны сервер связан с Интернет и может связаться с любым компьютером в мире. Сервер принимает стандартный телефонный сигнал, оцифровывает его (если он исходно не цифровой), сжимает, разбивает на пакеты и отправляет через Интернет по назначению с использованием протокола Интернет (ТСР/IP). Для пакетов, приходящих из Сети на телефонный сервер и уходящих в телефонную линию, операция происходит в обратном порядке.

Технология IP-телефонии объединяет сети с коммутацией каналов (передающие голосовую информацию) и сети с коммутацией пакетов (передающие данные) в единую коммуникационную сеть. Бесперебойное распознавание голоса и его передача из одной сети в другую решаются с помощью различных шлюзов. Как показано на рис. 10.6, шлюз представляет собой устройство, в которое с одной стороны включаются телефонные линии, а с другой — IP-сеть.

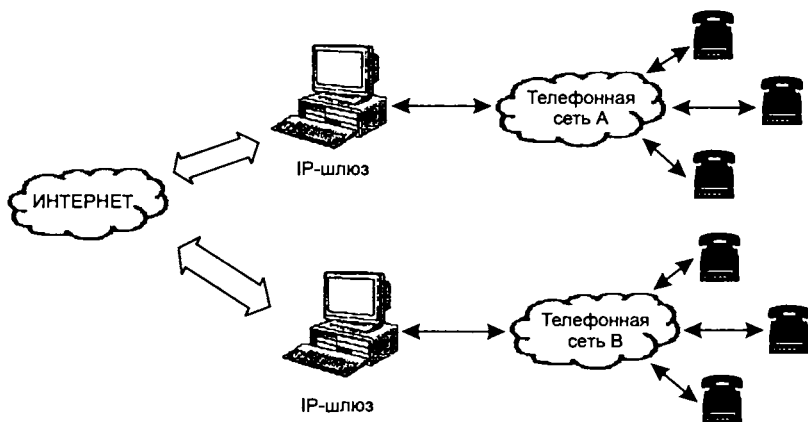


Рис. 10.6. Схема функционирования IP-телефонии

Голос, как аналоговые колебания в системе IP-телефонии, существует только в телефонной трубке. На остальных участках канала передачи от абонента к абоненту речь оцифровывается и передается в виде IP-пакетов. Пакеты, приходящие на ближайший к другому абоненту шлюз, преобразуются обратно в аналоговый вид (голосовой сигнал) и поступают в телефонную линию.

**Служба Telnet.** Данный сервис обеспечивает взаимодействие с удаленным компьютером. Он позволяет превратить компьютер пользователя в удаленный терминал другого компьютера. Поэтому данный сервис еще называют эмуляцией удаленного терминала. Терминал от обычного компьютера отличается тем, что не выполняет собственные вычисления. Все, что вводится на клавиатуре рабочей станции, передается удаленному компьютеру, а получаемые результаты передаются обратно и выводятся на монитор рабочей станции. В качестве удаленных компьютеров в основном используются машины, работающие под управлением операционной системы UNIX [Юникс]. Поэтому для работы в режиме удаленного терминала требуется знание основных команд данной операционной системы. С развитием графических операционных систем, таких, как Windows, командный режим работы стал менее популярен, и сервис Telnet в последнее время большинство пользователей не применяют. Многие информационные системы, ранее доступные исключительно с помощью Telnet, сегодня доступны из Всемирной паутины, о которой речь пойдет далее.

Установив связь с помощью **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. Реально **Telnet** предоставляет открытый доступ, но организация взаимодействия полностью определяется удаленным компьютером. Два вида услуг Интернет требуют подключения к серверам через **Telnet**: библиотечные каталоги и электронные доски объявлений (**BBS**).

Работа с удаленной системой может вестись в «прозрачном» режиме, когда программы на сервере и у клиента только обеспечивают протокол соединения, и в командном, когда клиент получает в свое распоряжение набор команд сервера.

Следует заметить, что из соображений безопасности намечается тенденция сокращения числа узлов Интернет, позволяющих использовать **Telnet** для подключения к ним.

### 10.2.1.2. Отложенное общение в Интернете

**Электронная почта — E-mail.** Это наиболее старая и одна из самых массовых служб Сети. Ее назначение — поддержка обмена электронными письмами между пользователями. По своей сущности электронная почта — это система обмена электронными сообщениями в компьютерных сетях (в режиме отложенного общения — *offline*).

Схема работы электронной почты отражена на рис. 10.7. Почтовый сервер — это своеобразное почтовое отделение, куда

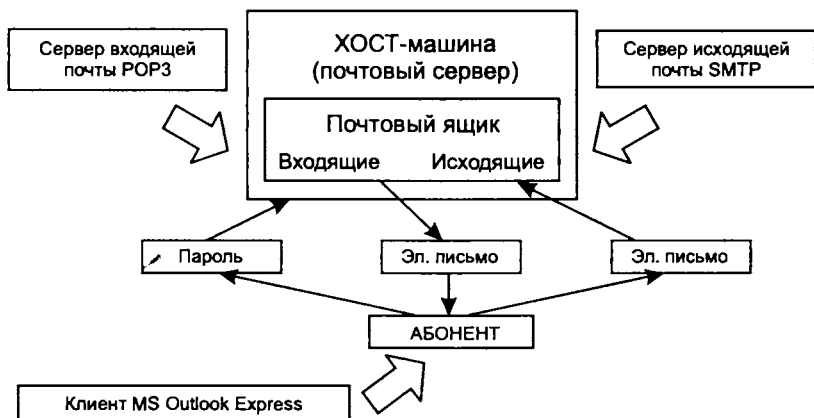


Рис. 10.7. Схема функционирования электронной почты

поступает входящая и исходящая корреспонденция зарегистрированных на нем пользователей. Эта корреспонденция помещается в «почтовые ящики» пользователей — специально отведенные разделы на жестком диске. Каждый пользователь получает персональный почтовый адрес, по которому к нему будут поступать письма. Следует отметить, что адреса электронной почты несколько отличаются от других адресов Интернет, но очень похожи. Они состоят из двух частей, разделенных символом @. Справа от символа располагается Интернет-адрес компьютера, на котором располагается почтовое отделение абонента. Этот адрес формируется так же, как и любое другое доменное имя в Интернете. Слева от символа @ расположено имя абонента. Примером почтового адреса может служить: **myname@mail.ru**.

Для того чтобы эта служба стала доступной пользователю, необходимо выполнить ряд процедур. Сначала пользователь должен зарегистрироваться на почтовом сервере. При этом фиксируется адрес клиента электронной почты и пароль — набор символов, позволяющий идентифицировать пользователя. Наличие пароля обеспечивает конфиденциальность данной услуги. Зарегистрированный пользователь имеет возможность написать текст письма, указав адрес получателя. Для этого в состав клиент-программы электронной почты включен редактор подготовки писем. Подготовленные письма помещаются в папку **Исходящие**. Сервер принимает все письма из папки **Исходящие** и, кроме того, передает поступившие письма, которые помещаются в папку **Входящие**. Соединение пользователя с почтовым сервером необходимо только для передачи письма и доставки на компьютер пользователя писем из папки входящие. Почтовый сервер работает постоянно. Он периодически просматривает «почтовые ящики» и организует передачу по сети исходящих писем. Входящую корреспонденцию почтовый сервер раскладывает по «ящикам».

Клиент-программа электронной почты, кроме функции приема-передачи писем во время сеанса связи, может выполнять еще множество сервисных функций:

- подготовка и редактирование писем;
- организация адресной книги;
- просмотр почтового архива;
- сортировка и удаление писем из почтового архива и пр.

Популярным клиентом E-mail является программа Outlook Express, входящая в стандартную поставку операционной системы MS Windows.

Сервер и клиент электронной почты работают по разным протоколам. Сервер программа POP3 (Post Office Protocol — протокол почтового отделения), кроме всего прочего, выполняет функцию защиты информации. Во время сеанса связи она устанавливает личность пользователя, обеспечивает связь с его персональным ящиком. При работе клиент-программы никакого установления личности не требуется. Ее задача — передать на сервер исходящие письма и принять поступившие. Здесь используется более простой протокол SMTP (Simple Mail Transfer Protocol — простейший протокол передачи почты).

**Служба телеконференций.** Телеконференция — это система обмена электронными сообщениями на определенную тему между абонентами сети (в режиме отложенного общения —offline). Каждый участник получает все материалы на свой почтовый адрес (E-mail). Каждое электронное письмо абонента публикуется на сервере телеконференции и доходит до всех участников (рис. 10.8).



Рис. 10.8. Схема функционирования телеконференции

В отличие от электронной почты, когда пользователь отправляет свое письмо персонально какому-то абоненту или группе абонентов, в телеконференциях письмо направляется одновременно всем ее участникам. В свою очередь, все сообщения, которые поступают в адрес конференции, будут поступать в почтовый ящик пользователя и загружаться в его компьютер во время сеанса связи. Чтобы стать участником телеконференции, на нее нужно подписаться. Для этих целей существуют определенные адреса. Всякая конференция посвящена определенной теме, поэтому переписка в ней происходит только в рамках темы. По некоторым данным число постоянно действующих в настоящее время конференций в Интернете превысило 50 000.



Службу телеконференций называют по-разному: *группы новостей*, *служба Usenet*. На узлах Сети работу телеконференций обслуживают *серверы новостей*. На ПК пользователя должна быть установлена *клиент-программа новостей*. Упомянутая выше программа MS Outlook Express является одновременно почтовым клиентом и клиентом новостей.

Телеконференции объединяют в себе как коммуникационную, так и информационную функции. С одной стороны, здесь происходит личностное общение, с другой — материалы конференции содержат большой объем полезной информации, которая определенное время хранится на сервере. Эта информация может рассматриваться как некоторый информационный ресурс (электронная газета). Это особенно важно для специалистов, участвующих в конференциях по профессиональной тематике: наука, производство, бизнес, торговля и пр. В материалах конференции можно найти ценные советы, консультации, которые помогут в принятии важных решений.

### 10.2.2. Информационные службы

Информационные службы предоставляют пользователям возможность доступа к определенным информационным ресурсам, хранящимся в Интернете. Такими ресурсами являются либо файлы в одном из общепринятых форматов, либо различные документы. Использование этих ресурсов обеспечивается с помощью соответствующих служб.

#### 10.2.2.1. Служба передачи файлов

Часто эту службу называют по имени используемого протокола: FTP (File Transfer Protocol — протокол передачи файлов). Со стороны Сети работу службы обеспечивают так называемые FTP-серверы, а со стороны пользователей — FTP-клиенты.

Назначение FTP-сервера — хранение набора файлов самого разнообразного назначения (обычно в архивированном виде). Чаще всего это программные файлы: средства системного и прикладного программного обеспечения. Но в наборах могут храниться файлы и любых других форматов: графические, звуковые, документы MS Word, MS Excel и пр. Вся эта информация образует иерархическую структуру папок (каталогов и подкаталогов).

После соединения FTP-клиента с сервером на экране пользователя открывается файловый интерфейс хранилища папок и файлов на сервере (наподобие Проводника Windows). Далее работа происходит так же, как с файловой системой на ПК: папки и файлы можно просматривать, сортировать, копировать на свои диски.

Клиент FTP входит в состав программы Internet Explorer и поэтому всегда имеется на ПК, работающем под управлением MS Windows.

#### 10.2.2.2. Службы поиска информации

**Информационная система Gopher [Гуфер].** Данное название переводится на русский язык как «суслик». Эта система была прообразом Всемирной паутины, выполняет функции, аналогичные WWW, которые будут рассмотрены далее.

Вся информация на Gopher-серверах хранится в виде иерархической системы меню. Начальный каталог Gopher является вершиной, а все остальные каталоги и файлы представляются элементами меню, разветвленных, как норы суслика. Gopher поддерживает разные типы файлов — текстовые, звуковые, программные и др. В настоящее время эта система постепенно отмирает, хотя некоторые ее приверженцы, особенно в США, где она была разработана, до сих пор пытаются ее поддерживать.

**Система WAIS [Вэйс].** Wide Area Information Service (Информационный сервис широкой области) также является альтернативой Всемирной паутины при осуществлении поиска. В основу системы положен принцип поиска информации с использованием логических запросов, основанных на применении ключевых слов. Клиент просматривает все серверы WAIS, проверяя наличие в них документов, удовлетворяющих запросу. В настоящее время данная система используется как поисковая машина в других информационных системах, например WWW и Gopher.

**Информационная система WHOIS.** Эта служба Интернет содержит информацию о пользователях Сети, их имена, электронные и почтовые адреса. WHOIS представляет собой распределенную систему, в которой информация содержится на значительном количестве серверов.

**World Wide Web (WWW).** WWW — это распределенная информационная система с гиперсвязями, существующая на технической базе всемирной компьютерной сети Интернет. Данная

информационная система представляет собой сеть документов, связанных между собой гиперссылками. Такие документы называются гипертекстовыми. Так как ссылки могут указывать на любой документ, находящийся в Интернете в любом месте земного шара, данная система и названа Всемирной паутиной. Изучение возможностей WWW предполагает использование новых терминов.

Гиперссылка — это некоторое ключевое слово или объект в документе, с которым связан указатель для перехода на другую страницу, в другой документ. Гипертекст — множество отдельных документов, связанных гиперссылками. Пример гипертекста приведен на рис. 10.9.

Наименьшей информационной единицей WWW является Web-страница, представляющая собой совокупность текстовых, графических и мультимедийных файлов, связанных гиперссылками. Группы Web-страниц, принадлежащие одному владельцу и связанные между собой по содержанию, составляют Web-сайт. Хост-компьютер, предназначенный для хранения Web-страниц и Web-сайтов, называется Web-сервером. Клиент-программа, предназначенная для просмотра Web-сайтов, называется браузером (от англ. browse — просматривать, пролистывать).

Для работы с WWW используется специальный протокол HTTP [Эйч Ти Ти Пи] — Hyper Text Transfer Protocol (Протокол

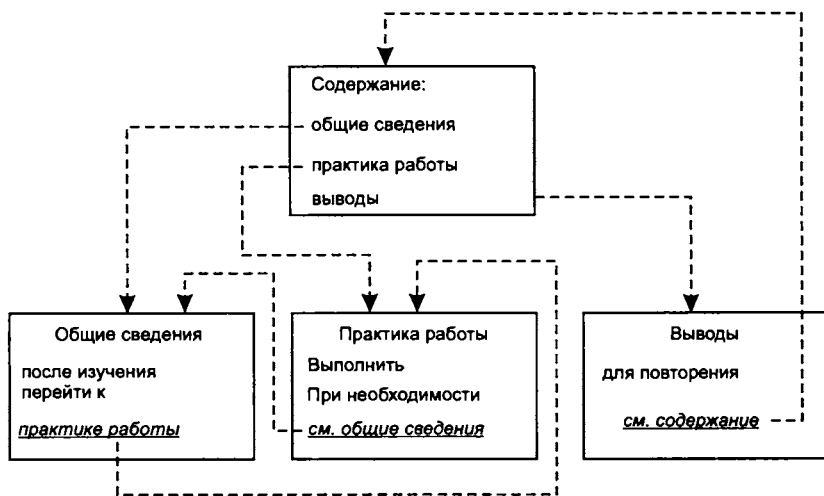


Рис. 10.9. Пример гипертекста

передачи гипертекста). Гипертекстовые документы создаются с помощью специального языка HTML [Эйч Ти Эм Эл] — Hyper Text Markup Language (Язык разметки гипертекста). С этой точки зрения Web-страница представляет собой документ во Всемирной паутине, составленный на языке HTML и доступный для просмотра с помощью браузера.

Кроме перечисленных, существует еще ряд сервисов в Интернет, основанных на базовых протоколах TCP/IP. Некоторые из них только в стадии зарождения, а другие слишком слабо распространены, чтобы упоминать о них. Наиболее перспективной областью использования Интернет является электронная коммерция. Можно осуществлять сделки непосредственно с помощью Интернет, управляя средствами на своем счету в банке. Другим интересным направлением является дистанционное обучение. Очевидно, что в основе этих сервисов используются как коммуникационные, так и информационные возможности Интернет.

Анализ описания служб Интернет позволяет сделать следующие выводы:

- в каждой службе используется свой протокол. Таким образом, понятия «прикладной протокол», «сервис Интернет», «услуги и возможности, предоставляемые Интернет» имеют одно и то же содержание;
- в Интернет имеется ряд протоколов, построенных на базовых протоколах TCP/IP и предлагающих разнообразный сервис.

Как и во всякой другой сети, в Интернет существует 7 уровней взаимодействия между компьютерами: физический, логический, сетевой, транспортный, уровень сеансов связи, представительский и прикладной. Каждому уровню взаимодействия соответствует свой набор протоколов (т. е. правил взаимодействия):

Протоколы физического уровня определяют вид и характеристики линий связи между компьютерами. Для каждого типа линий связи разработан соответствующий протокол логического уровня, занимающийся управлением передачи информации по каналу.

Протоколы сетевого уровня отвечают за передачу данных между устройствами в разных сетях, т. е. занимаются маршрутизацией пакетов в сети. К протоколам сетевого уровня принадлежат IP (Internet Protocol) и ARP (Address Resolution Protocol).

Протоколы транспортного уровня управляют передачей данных из одной программы в другую. К протоколам транспортного

уровня принадлежат TCP (Transmission Control Protocol) и UDP (User Datagram Protocol).

Протоколы уровня сеансов связи отвечают за установку, поддержание и уничтожение соответствующих каналов. В Интернет этим занимаются уже упомянутые протоколы TCP и UDP, а также протокол UUCP (Unix to Unix Copy Protocol).

Протоколы представительского уровня занимаются обслуживанием прикладных программ. К таким протоколам принадлежат запускаемые, к примеру, на Unix-сервере протоколы, предоставляющие различные услуги абонентам. К ним относятся также протоколы: Telnet, FTP, Gopher, NFS, NNTP (Net News Transfer Protocol), SMTP (Simple Mail Transfer Protocol), POP2 и POP3 (Post Office Protocol) и т. д.

К протоколам прикладного уровня относятся сетевые службы и программы их предоставления. Представление о взаимодействии протоколов различного уровня дает рис. 10.10.

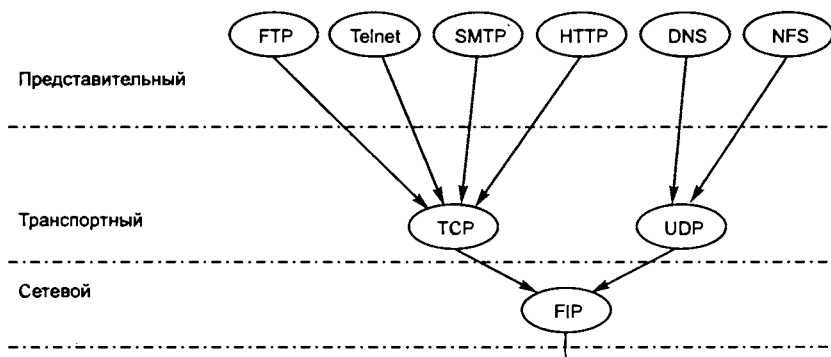


Рис. 10.10. Взаимодействие протоколов в Интернет

Такой характер взаимодействия протоколов позволяет использовать при работе в Интернет не просто доменные адреса, а универсальные указатели ресурсов, называемые URL [Ю Эр Эль] — Universal Resource Locator. URL — это адрес любого ресурса в Интернет вместе с указанием того, с помощью какого протокола следует к нему обращаться. В указателе, кроме собственно адреса, имеются сведения о том, каким протоколом следует обращаться к данному ресурсу, какую программу для этого следует запустить на сервере и к какому конкретному файлу следует обратиться на сервере.

Примером указателя может быть <http://www.microsoft.com/ie>. Название протокола **http** в начале указывает, что далее следует адрес Web-страницы, а название **ie** в конце указателя описывает каталог с именем **ie** на сервере **www.microsoft.com**.

Указатель <ftp://www.mycompany.ru/business/index.html> описывает, что к файлу **index.html**, расположенному в каталоге **business** на сервере **www.mycompany.ru**, следует обратиться по протоколу передачи файлов **FTP**.

### 10.2.3. Поиск информации в Интернете

В настоящее время для поиска информации в Интернете наиболее широко используется служба WWW. В качестве основных источников информационных ресурсов WWW могут рассматриваться:

- информационные агентства;
- средства массовой информации;
- специализированные базы данных;
- Web-сайты юридических и физических лиц.

Эти ресурсы могут храниться в форме файлов стандартных форматов либо разного рода документов, которые можно просмотреть, распечатать, сохранить.

Информационные агентства являются источником наиболее достоверной и оперативной информации о различных событиях и явлениях в различных сферах жизни. Эти агентства располагают развитой сетью корреспондентских пунктов. Считается, что они являются одним из наиболее объективных и точных источников информации, в меньшей степени подверженных конъюнктурным влияниям. Достоинством этой информации считается минимум комментариев в сообщении. К числу наиболее авторитетных российских агентств относятся ИТАР-ТАСС, ИНТЕРФАКС, РИА «Новости». Все эти агентства имеют свои сайты в Интернете.

Средства массовой информации. Информация на сайтах традиционных СМИ менее оперативна из-за дискретности выхода номеров изданий (ежедневное, еженедельное, ежемесячное и т. п.). Естественно, содержание изданий во многом определяется тем, что поступает с лент информационных агентств, затем записывается в том виде, в каком представляют журналисты, учитывающие интересы хозяев. Средства массовой инфор-

мации — некий фильтр, в котором из общего потока событий от информагентств выбираются необходимые, определяется порядок и приоритеты их подачи, расставляются требуемые (с точки зрения конкретных людей) акценты.

Информация от агентств препарируется журналистами в СМИ и разбавляется сведениями из собственных источников. Она становится основой для публикаций, отражающих точку зрения либо авторов, либо руководства издания. С одной стороны, это добавляет «информационного шума», но с другой — по нему можно анализировать реакцию на события конкретных групп влияния.

Обычно разделяют центральные и региональные, традиционные и электронные, специальные и развлекательные СМИ. С точки зрения наблюдателя в Интернет это деление весьма условно. Большинство традиционных газет и журналов имеют сегодня свои Интернет-серверы. Содержание и информационная насыщенность новых Интернет-изданий не уступает большинству СМИ с многолетним опытом работы. Поэтому очень скоро СМИ, представленные в Интернет, можно будет классифицировать только по степени ангажированности и возможности поиска в архивах газетных публикаций. При оценке информации, полученной из этого источника, следует учитывать их принадлежность государственным учреждениям, коммерческим структурам или различным группам влияния.

Один из крупнейших собственников СМИ — Правительство Российской Федерации. Его официальный сайт находится по адресу: <http://www.gov.ru>. Ему же принадлежат сайты Росинформцентра и «Российской газеты». Собственными СМИ располагают Федеральное собрание РФ, правительство Москвы, отдельные министерства.

Вместе с тем значительные информационные ресурсы в Интернете имеют такие структуры, как «Газпром», финансово-промышленная группа «Интеррос» и другие, принадлежащие олигархам и обслуживающие их интересы.

Специализированные базы данных в Интернет — это сайты, поддерживаемые, как правило, информационными или консалтинговыми фирмами. В них сконцентрированы данные из многих наиболее интересных источников СМИ, описанных ранее. Основное их преимущество — возможность поиска с использованием специализированных языков запросов. Количество подобных продуктов на рынке информацион-

ных услуг постоянно увеличивается. В качестве наиболее типичных примеров таких баз можно рассматривать информационно-поисковую систему «Артефакт» (<http://www.integrum.com>), информационно-аналитическую систему «Галактика-Zoom» (<http://zoom2.galaktika.ru>) и междуниверситетскую информационную систему «Россия» (<http://www.cir.ru>).

По некоторым данным в настоящее время общее число документов, размещенных в WWW, превышает миллиард. Такое обилие документов порождает проблему поиска. Очевидно, что сам факт использования при создании WWW протокола HTTP обуславливает три способа поиска необходимой информации во «всемирной паутине».

**Поиск по адресу.** Данный вид поиска применяется в тех случаях, когда пользователю известен адрес Web-сайта, содержащего необходимую ему информацию. Форма адреса (IP, доменный или URL) в этом случае значения не имеет.

**Поиск по гиперссылкам.** В этом случае пользователь осуществляет поиск необходимой информации, пользуясь гиперссылками, размещенными на том или ином Web-сайте.

**Поиск с помощью поисковых серверов.** Такая возможность обеспечивается наличием в Интернете поисковых серверов. *Поисковыми серверами* называют выделенные хост-компьютеры, в которых размещаются базы данных ресурсов Интернет. Достаточно передать такому серверу фразу или набор ключевых слов, описывающих тему, интересующую пользователя, и сервер возвратит список ресурсов, соответствующих данному запросу. Современные поисковые системы поддерживают индексы, включающие весьма значительную часть ресурсов Интернет. Если в Интернет есть необходимая пользователю информация, ее наверняка можно найти с помощью поисковых серверов.

Для понимания принципов организации поиска информации с помощью поисковых серверов важное значение имеют два момента:

- во-первых, доступ к определенному документу осуществляется по каналам связи, обладающим конечной пропускной способностью, зависящей от технических характеристик канала и его текущей загрузки. Следовательно, анализ документа на его соответствие запросу требует определенного времени. Это означает, что если загрузка и анализ одного документа информационно-поисковой системой составляет всего 1 секунду, то при этом для анализа 20 млн доку-



ментов, присутствующих в ресурсе WWW только русскоязычной части Интернет (РУНет), потребуется около 8 месяцев;

- во-вторых, ресурсы Сети не имеют единого администрирования. Это означает, что в Сети могут изменяться, появляться новые и исчезать существующие документы и целые сайты, и при этом независимо от других. Частота обновления информации в документах для разных сайтов различна: для одних — это несколько раз в час, для других — раз в сутки, день, месяц и т. д. В то же время некоторые документы могут оставаться неизменными с момента своего появления практически неограниченное время. Поэтому конечный процесс анализа информационных объектов на соответствие какому-либо запросу в принципе невозможен.

В связи с этим, все информационно-поисковые системы вынуждены иметь специальную базу данных, в которой содержится описание части документов Сети на определенный момент времени. Поэтому очень важно понимать, что при использовании информационно-поисковых систем для нахождения информации в Интернет поиск осуществляется не на реальном пространстве документов Сети, а в некоторой модели, содержание которой может значительно отличаться от действительного содержания Интернет в момент проведения поиска. Это объясняется тем, что каждая информационно-поисковая система имеет в своем составе некоторую базу данных, содержащую информацию о документах.

Принципиальная особенность таких баз данных состоит в том, что при включении в нее документа с ним связывается некоторый набор данных, который в дальнейшем позволяет по предъявленному запросу осуществить проверку на соответствие. Такой набор данных может содержать множество ключевых слов, или классификационную информацию, позволяющую определить место документа в иерархическом рубрикаторе. Поэтому при анализе соответствия документа запросу с помощью поисковых серверов используется не текст документа, а связанный с ним набор данных. Следовательно, особенности построения базы данных информационно-поисковой системы существенно влияют на полноту и точность поиска.

Пополнение базы данных информационно-поисковой системы данными о документах может производиться автоматически

или вручную — это зависит от типа системы и определяет скорость ее наполнения, что, в свою очередь, также влияет на полноту поиска. По способу создания баз данных ресурсов Интернет можно выделить две группы поисковых серверов: каталоги и поисковые машины.

Второй способ классификации поисковых служб — классификация по глубине охвата ресурса. По данному признаку системы делятся на *глобальные* и *локальные*. Глобальные каталоги и поисковые машины действуют на всем пространстве WWW. Это означает, что все документы WWW потенциально могут попасть в базу данных такой информационно-поисковой системы.

В локальных информационно-поисковых системах поисковое пространство является частью пространства WWW и, следовательно, не все документы этого ресурса могут попасть в базу данных таких систем. Среди локальных каталогов и поисковых машин выделяют *региональные* системы, информационно-поисковое пространство которых строится по региональному или национальному признаку, и *специализированные* системы, которые действуют на пространстве документов, соответствующих какой-либо определенной тематике.

Рассмотрим подробнее особенности различных поисковых серверов.

Каталоги — иерархические базы данных о документах сети, пополнение которых осуществляется человеком. Основной характеристикой каталогов является объем базы данных, т. е. количество представленных в ней документов. Для глобальных каталогов эта величина не превышает 5 млн документов, что составляет около 0,5 % общего пространства WWW. Поэтому использование каталогов не может дать результатов, обладающих высокой полнотой. По сравнению с поисковыми машинами скорость наполнения баз данных каталогов достаточно мала, поскольку их пополнение происходит вручную. Однако это может рассматриваться и как преимущество поисковых систем данного типа, поскольку ручной отбор позволяет повысить точность поиска.

В качестве второй характеристики каталогов следует выделить количество категорий или рубрик в них. Естественно, чем выше это значение, тем большей точностью характеризуются результаты поиска. Для глобальных каталогов это значение в настоящее время составляет несколько десятков тысяч.

Принципы работы с каталогами достаточно просты. Как правило, главная страница Web-сайта каталога содержит в себе

список категорий, или рубрик, каждая из которых представлена гиперссылкой. По такой гиперссылке пользователь может перейти на страницу, содержащую список подразбук или документов, соответствующих этой рубрике. В табл. 10.4 представлены некоторые каталоги.

Таблица 10.4. Каталоги в WWW

Глобальные	Российские региональные
Britanica ( <a href="http://www.britanica.com">www.britanica.com</a> )	@Rus ( <a href="http://www.atrus.ru">www.atrus.ru</a> )
Licos ( <a href="http://www.lycos.com">www.lycos.com</a> )	City.ru ( <a href="http://www.city.ru">www.city.ru</a> )
Look Smart ( <a href="http://www.looksmart.com">www.looksmart.com</a> )	List.ru ( <a href="http://www.list.ru">www.list.ru</a> )
NBCi ( <a href="http://www.nbciconi.com">www.nbciconi.com</a> )	
Open Directory( <a href="http://dmoz.org">dmoz.org</a> )	
YAHOO ( <a href="http://www.yahoo.com">www.yahoo.com</a> )	

Поисковые машины — иерархические базы данных о документах сети, пополнение которых осуществляется специальными программами. Поисковые машины устроены гораздо сложнее каталогов. В структуре поисковой машины выделяют обычно три составляющие:

- специальную программу, осуществляющую сканирование Сети, именуемую *поисковым роботом*;
- базу данных документов, называемую *индексом*;
- интерфейсную часть, предназначенную для взаимодействия с пользователем.

Поисковый робот осуществляет просмотр Сети с целью выявления новых документов для включения их в базу данных поисковой машины — индекс. Процесс занесения информации о документе в индекс называется *индексированием*. Чтобы понять, каким образом робот осуществляет отбор документов для анализа, достаточно вспомнить, что большинство опубликованных Web-страниц содержат гиперссылки на другие объекты. Наличие гиперссылок позволяет роботу автоматически формировать очередь документов. В процессе занесения документов в индекс поисковая машина составляет список слов документа, которые являются ключевыми. Каждая поисковая машина имеет свои алгоритмы индексирования и форматы индексных файлов, составляющих базу данных. В табл. 10.5 представлен список некоторых поисковых машин.

Таблица 10.5. Поисковые машины в WWW

Глобальные	Российские
Altavista ( <a href="http://www.altavista.com">www.altavista.com</a> )	Апорт ( <a href="http://www.aport.ru">www.aport.ru</a> )
Excite ( <a href="http://www.excite.com">www.excite.com</a> )	Rambler ( <a href="http://www.rambler.ru">www.rambler.ru</a> )
Fast ( <a href="http://www.allthcweb.com">www.allthcweb.com</a> )	Яндекс( <a href="http://www.yandex.ru">www.yandex.ru</a> )
Go(Infoseek) ( <a href="http://infoseek.go.com">infoseek.go.com</a> )	
Google! ( <a href="http://www.google.com">www.google.com</a> )	
HotBot ( <a href="http://www.hotbot.com">www.hotbot.com</a> )	
Northen Light ( <a href="http://www.nlsearch.com">www.nlsearch.com</a> )	

Следует отметить, что совершенствование методов поиска информации в Интернет привело к созданию нового инструмента — метапоисковых систем. Такие системы не имеют собственных поисковых инструментов и собственной базы данных документов. Их основная задача состоит в том, чтобы передать запрос пользователя «настоящим» поисковым системам. Важным достоинством систем метапоиска является то, что они позволяют задействовать сразу несколько поисковых служб, при этом у пользователя нет необходимости подключаться к каждой из этих служб и многократно вводить запрос.

Метапоисковые системы бывают двух видов: представленные в виде Web-сайта в Интернет и реализованные как отдельные программы, требующие установки на пользовательский компьютер. В качестве систем первой группы можно назвать глобальные службы Search ([www.search.com](http://www.search.com)) и MetaCrawler ([www.metacrawler.com](http://www.metacrawler.com)), а также российскую систему «Следопыт» ([www.medialingua.ru/www/wwwsearc.htm](http://www.medialingua.ru/www/wwwsearc.htm)).

Примером систем второй группы можно назвать программу Internet Explorer, оснащенную некоторыми функциями метапоиска. Запрос, который пользователь вводит через форму «Поиск», переадресовывается различным поисковым машинам. Набор используемых систем поиска зависит от того, для какого региона локализована версия браузера. Для русской версии — это отечественные системы Yandex, Rambler, Aport, глобальная машина Excite и система поиска от Microsoft «MSN Web Search». К сожалению, добавить в этот список новую систему крайне затруднительно для обычного пользователя, не имеющего детального представления о программной архитектуре Windows и

Internet Explorer, однако можно легко отказаться от использования той или иной службы.

При использовании поисковых машин и метапоисковых систем основным средством поиска является поиск по ключевым словам. При этом существенное значение имеет формулировка запроса с использованием ключевых слов. В каждой поисковой системе используется свой синтаксис при составлении запроса. Пример операторов, используемых с этой целью, приведен в табл. 10.6.

Таблица 10.6. Операторы, используемые при создании запросов

Оператор	RAMBLER	YANDEX	APORT
Логическое И	And & пробел	& пробел (в пределах предложения)	AND & + И пробел
Логическое ИЛИ	Or 		OR   или
Логическое НЕ	NOT !	~ (оператор «И НЕ» в пределах предложения) ~~ (оператор «И НЕ» в пределах документа)	NOT — НЕ
Группировка	()	()	()

### Контрольные вопросы

1. Состав и структура Интернет.
2. Способы адресации в Интернет: IP и доменные адреса, URL.
3. Краткая характеристика основных служб Интернет.
4. Основные информационные ресурсы Интернет.
5. Основные способы поиска информации в Интернет.

### 11.1. Угрозы безопасности информационных систем

В соответствии с [15] информация представляет собой сведения о лицах, предметах, фактах, событиях, явлениях и процессах независимо от формы их представления. Исследования специалистов компании Ontrack показывают, что один час простоя из-за потери информации обходится от 28 тыс. долларов (доставка посылок) до 645 млн долларов (брокерские компании). В странах, вступивших в постиндустриальный этап развития, десятидневное бездействие сервера способно привести к необратимому финансовому ущербу любой фирме [11].

#### 11.1.1. Информация как объект защиты

Информация может являться предметом собственности и подлежит защите в соответствии с требованиями правовых документов или требованиями, устанавливаемыми собственником информации [15]. При этом не имеет значения, кто является собственником информации: государство, юридическое лицо, группа физических лиц, отдельное физическое лицо. В РФ в основу организации работы с информацией положен «Федеральный закон об информации, информатизации и защите информации» от 10 января 2003 г. № 15-ФЗ. Он регулирует отношения, возникающие при:

- формировании и использовании информационных ресурсов на основе создания, сбора, обработки, накопления, хранения, поиска, распространения и предоставления потребителю документированной информации;
- создании и использовании информационных технологий и средств их обеспечения;

- защите информации, прав субъектов, участвующих в информационных процессах и информатизации.

Особенность информации по сравнению с другими объектами права состоит в том, что она зачастую является результатом интеллектуального труда. Поэтому отношения, возникающие в связи с созданием и использованием программ для ЭВМ и баз данных, регулируются Законом Российской Федерации «Об авторском праве и смежных правах<sup>1</sup>». Учитывая особенности юридической терминологии, следует помнить, что в нем подразумевается, что:

- база данных — объективная форма представления и организации совокупности данных (статей, расчетов и т. д.), систематизированных таким образом, чтобы эти данные могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ);
- программа для ЭВМ — объективная форма представления совокупности данных и команд, предназначенных для функционирования ЭВМ и других компьютерных устройств с целью получения определенного результата, включая подготовительные материалы, полученные в ходе разработки программы для ЭВМ и порождаемые ею аудиовизуальные отображения;

Кроме того, информация может составлять государственную, служебную или коммерческую тайну. Наказание за разглашение государственной тайны предусмотрено статьей 283 УК РФ. Гражданский кодекс РФ определяет, что информация составляет служебную или коммерческую тайну в случае, когда:

- она имеет действительную или потенциальную коммерческую ценность в силу неизвестности ее третьим лицам;
- к ней нет свободного доступа на законном основании;
- обладатель информации принимает меры к охране ее конфиденциальности.

Анализ требований законов позволяет установить, что человек всегда находится в определенных отношениях с информацией. При этом он может быть:

- источником (поставщиком) информации;
- пользователем (потребителем) информации;

---

<sup>1</sup> Современная редакция закона установлена Федеральным законом от 20.07.2004 № 72-ФЗ и уточнена Федеральным законом от 19.07.1995 № 110-ФЗ.

- собственником (владельцем, распорядителем) информации;
- лицом, о котором собирается информация;
- владельцем системы сбора и обработки информации;
- участником информационных процессов.

Рассматривая эти отношения, нужно помнить, что собственность представляет собой форму присвоения материальных благ. В соответствии с законодательством РФ право собственности включает в себя права владения, распоряжения и пользования объектом собственности. При этом следует понимать, что владение означает фактическое обладание предметом собственности, распоряжение — право решения об использовании или применении и пользования объектом собственности, а пользование — извлечение из объекта собственности полезных свойств и доходов.

### **11.1.2. Понятие безопасности информационных систем**

Достижение успеха в любой отрасли человеческой деятельности невозможно без использования той или иной информации. Каждый из субъектов информационных отношений заинтересован в том, чтобы информация, с которой он взаимодействует, отвечала некоторым требованиям. Анализ показывает, что эти требования зависят от места, занимаемого тем или иным субъектом, в информационном процессе. К числу этих требований относятся :

- своевременный доступ к необходимой информации;
- конфиденциальность (сохранение в тайне) определенной части информации;
- достоверность (полнота, точность, адекватность) информации;
- защищенность от дезинформации, т. е. от получения ложной (недостоверной, искаженной) информации;
- защищенность информации от незаконного ее тиражирования, т. е. обеспечение возможности защиты авторских прав или прав собственника информации;
- разграничение ответственности за нарушения законных прав (интересов) других субъектов информационных отношений и установленных правил обращения с информацией;
- возможность осуществления непрерывного контроля и управления процессами обработки и передачи информации.



Воздействие на информацию либо на ее носители, а также неправомерное использование информации может нанести ущерб субъекту информационных отношений. Этот ущерб может быть прямым или косвенным, материальным или моральным. Поэтому все субъекты информационных отношений заинтересованы в обеспечении своей информационной безопасности (конечно в различной степени в зависимости от величины ущерба, который им может быть нанесен).

Под безопасностью информации (БИ) принято понимать ее способность сохранять неизменность своих свойств при воздействии случайных или преднамеренных внешних воздействий. К числу свойств, обеспечивающих безопасность информации, относятся:

1) доступность информации — способность обеспечивать своевременный беспрепятственный доступ субъектов к интересующей их информации;

2) целостность информации — способность существовать в неискаженном виде (неизменном по отношению к некоторому фиксированному ее состоянию). Это свойство включает физическую целостность данных, их логическую структуру и содержание;

3) конфиденциальность информации — способность системы (среды) сохранять информацию втайне от субъектов, не имеющих полномочий на доступ к ней.

Как известно, средой, в которой существует информация, является информационная система — совокупность данных, программно-аппаратных средств и персонала, обеспечивающая хранение, обработку и выдачу информации для решения прикладных задач. Из определения безопасности информации следует, что основными формами нарушения безопасности информации, которые может понести субъект в результате ее нарушения, являются:

1) нарушение доступности информации вследствие полной или частичной утраты работоспособности системы. Очевидно, что вывод из строя или недопустимое изменение режимов работы компонентов системы обработки информации может приводить к получению неверных результатов расчетов, отказам системы и/или отказам в обслуживании конечных пользователей;

2) нарушение целостности информации, которое может быть вызвано ее полным или частичным уничтожением, а также преднамеренным или случайным искажением;

3) нарушение конфиденциальности информации. Для закрытой информации это означает ее раскрытие, а для открытой — ее несанкционированное тиражирование, например, программ, баз данных, разного рода документации, литературных произведений и т. д. в нарушение прав собственников информации, авторских прав и т. п.

Итак, нарушение безопасности информации может нанести ущерб тому или иному субъекту информационных отношений. Формы и степень ущерба зависят от формы и степени нарушения БИ. Поэтому далее целесообразно рассмотреть возможные причины этого.

### **11.1.3. Угрозы безопасности информационных систем и их классификация**

Под угрозой (вообще) обычно понимают потенциально возможное событие, действие (воздействие), процесс или явление, которые могут привести к нанесению ущерба чьим-либо интересам. Исходя из этого определения, угрозой безопасности информационных систем (УБИС) принято называть потенциально возможное событие, процесс или явление, которые посредством воздействия на информацию или другие компоненты информационных систем могут прямо или косвенно привести к нанесению ущерба интересам субъектов информационных отношений. Реализация угрозы приводит к нарушению информационной безопасности, что, в свою очередь, наносит ущерб интересам субъектов информационных отношений.

Статистика свидетельствует, что основными причинами нарушений информационной безопасности являются [5]:

- стихийные бедствия и аварии (наводнение, ураган, землетрясение, пожар и т. п.), приводящие к повреждениям элементов информационных систем;
- сбои и отказы оборудования (технических средств) информационных систем;
- последствия ошибок проектирования и разработки компонентов информационных систем (аппаратных средств, технологии обработки информации, программ, структур данных и т. п.);
- ошибки эксплуатации (пользователей, операторов и другого персонала);

- преднамеренные действия нарушителей и злоумышленников (обиженных лиц из числа персонала, преступников, шпионов, диверсантов и т. п.).

Анализ этих причин позволяет сделать вывод, что существует большое количество угроз безопасности информации различного происхождения. Поэтому их следует классифицировать. Учитывая выводы, сформулированные в [15], источники угроз безопасности информационных систем можно классифицировать по различным признакам.

**1. Направленность угроз.** Этот признак определяет одно из свойств, определяющих безопасность информации. По этому признаку можно выделить:

- угрозы доступности информации путем воздействия на аппаратные элементы информационных систем;
- угрозы целостности информации, предполагающие:
  - а) нарушение физической целостности путем уничтожения или повреждения данных;
  - б) нарушение логической структуры данных путем ее искажения;
  - в) нарушение содержания данных путем их несанкционированной модификации;
- угрозы конфиденциальности информации в результате несанкционированного ее получения. Следствием этого может стать присвоение чужого права собственности.

**2. Происхождение угроз.** Предполагается существование двух возможных классов угроз с точки зрения их происхождения: случайное и преднамеренное.

Случайное происхождение угроз обусловлено спонтанными обстоятельствами, возникающими в информационных системах в процессе их функционирования независимо от воли людей. Наиболее известными событиями такого рода являются отказы, сбои, ошибки, стихийные бедствия и побочные влияния. Сущность перечисленных событий (кроме стихийных бедствий, сущность которых ясна) определяется следующим образом:

- отказ — нарушение работоспособности какого-либо элемента системы, приводящее к невозможности выполнения им основных своих функций;
- сбой — временное нарушение работоспособности какого-либо элемента системы, следствием чего может быть неправильное выполнение им в этот момент своей функции;

- ошибка — неправильное (разовое или систематическое) выполнение элементом одной или нескольких функций, происходящее вследствие специфического (постоянного или временного) его состояния;
- побочное влияние — негативное воздействие на систему в целом или на отдельные ее элементы, оказываемое какими-либо явлениями, происходящими внутри системы или во внешней среде.

Преднамеренное происхождение угрозы обуславливается злоумышленными действиями людей.

**3. Предпосылки появления угроз.** Различаются две возможные разновидности предпосылок.

**Объективные:**

а) количественная недостаточность элементов системы означает физическую нехватку одного или нескольких элементов системы, вызывающую нарушения технологического процесса обработки данных и/или перегрузку имеющихся элементов;

б) качественная недостаточность — несовершенство конструкции (организации) элементов системы, вследствие чего могут появляться возможности случайного или преднамеренного негативного воздействия на обрабатываемую или хранимую информацию.

**Субъективные:**

а) **деятельность разведорганов иностранных государств** — специально организуемая деятельность государственных органов, профессионально ориентированных на добывание необходимой информации всеми доступными средствами. К основным видам разведки относятся агентурная и техническая. Агентурная разведка — это несанкционированная деятельность профессиональных разведчиков, завербованных агентов и так называемых «доброжелателей». Техническая разведка включает радиоразведку (перехват радиоэлектронными средствами информации, циркулирующей в телекоммуникационных каналах), радиотехническую разведку (регистрацию спецсредствами электромагнитных излучений технических систем) и космическую разведку (использование космических кораблей и искусственных спутников Земли для наблюдения за территорией, ее фотографирования, регистрации радиосигналов и получения полезной информации любыми другими доступными способами);

б) **промышленный шпионаж** — негласная деятельность организации или ее представителей по добыванию информации, специ-

ально охраняемой от несанкционированной ее утечки или хищения, с целью создания для себя благоприятных условий и получения максимальных выгод (недобросовестная конкуренция);

в) **злоумышленные действия уголовных элементов** — хищение информации или компьютерных программ в целях наживы;

г) **действия недобросовестных сотрудников** — хищение (копирование) или уничтожение информационных массивов и/или программ по эгоистическим или корыстным мотивам, а также в результате несоблюдения установленного порядка работы с информацией.

**4. Источники угроз.** Угрозы безопасности информационных систем реализуются путем воздействия на структурно-функциональные элементы информационных систем. Для того чтобы уяснить, каким образом можно осуществить такое воздействие, следует рассмотреть структуру ИС. В общем случае возможная структура информационной системы может иметь вид, представленный на рис. 11.1. На основании анализа состава данной схемы можно сделать вывод о том, что в составе информационной системы имеется две группы элементов: персонал и аппаратно-программные средства. Обе группы элементов могут быть как объектами воздействия угроз безопасности, так и источниками этих угроз.

**Персонал** выполняет функции пользователей, администраторов системы, администраторов баз данных, специалистов по эксплуатации аппаратных средств, программистов и т. д.

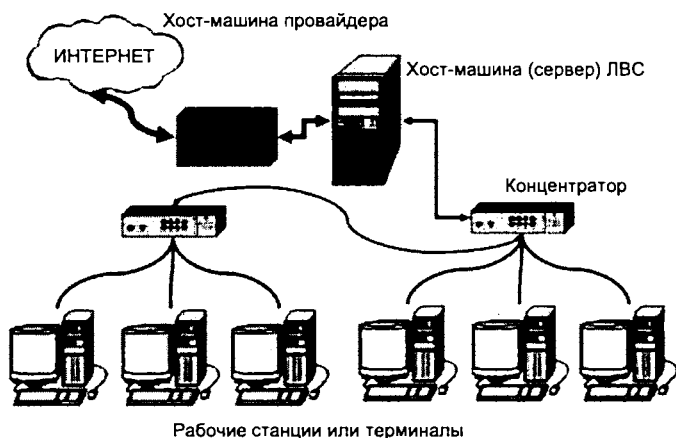


Рис. 11.1. Возможная структура информационной системы

**Аппаратно-программные средства** составляют техническую основу информационной системы. В зависимости от назначения ИС количество, состав и способы соединения этих элементов в разных ИС могут различаться.

Тем не менее можно с высокой степенью достоверности утверждать, что основными аппаратными элементами информационных систем являются:

- *рабочие станции* — отдельные ЭВМ или удаленные терминалы сети, на которых реализуются автоматизированные рабочие места пользователей, т. е. пользователей (потребителей) информации или участников информационных процессов;
- *серверы* (файлов, печати, баз данных и т. п.) — высокопроизводительные ЭВМ, предназначенные для реализации функций хранения, печати данных, обслуживания рабочих станций сети и подобных действий;
- *межсетевые мосты* (шлюзы, центры коммутации пакетов, коммуникационные ЭВМ) — элементы, обеспечивающие соединение нескольких сетей передачи данных либо нескольких сегментов одной и той же сети, имеющих различные протоколы взаимодействия;
- *каналы связи* (локальные либо с узлами коммутации).

Воздействие угроз безопасности на аппаратные элементы информационных систем приводит к нарушению их работоспособности. Принято различать три степени таких нарушений:

- *отказ* — нарушение работоспособности элемента системы, приводящее к невозможности выполнения ею своих основных функций;
- *сбой* — временное нарушение работоспособности элемента системы, приводящее к неправильному выполнению ею своих основных функций;
- *ошибка* — неправильное выполнение элементом системы одной или нескольких функций.

С точки зрения обеспечения безопасности информации эти аппаратные элементы отличаются своими свойствами, определяющими их устойчивость по отношению к угрозам безопасности информации.

**Рабочие станции** являются наиболее доступными компонентами сетей и именно с них могут быть предприняты наиболее многочисленные попытки совершения несанкционированных действий. С рабочих станций осуществляется управление про-

цессами обработки информации, запуск программ, ввод и корректировка данных, на дисках рабочих станций могут размещаться важные данные и программы обработки. На видеомониторы и печатающие устройства рабочих станций выводится информация при работе пользователей (операторов), выполняющих различные функции и имеющих разные полномочия по доступу к данным и другим ресурсам системы. Именно поэтому рабочие станции должны быть надежно защищены от доступа посторонних лиц, а их операционные системы должны содержать средства разграничения доступа к ресурсам со стороны законных пользователей, имеющих разные полномочия. Кроме того, средства защиты должны предотвращать нарушения нормальной настройки рабочих станций и режимов их функционирования, вызванные неумышленным вмешательством неопытных (невнимательных) пользователей.

**Серверы (Host-машины) и мосты** нуждаются в особой защите, так как они особо привлекательны для злоумышленников. Первые — так как в них концентрируются большие объемы информации, а вторые — как элементы, в которых осуществляется преобразование (возможно, через открытую, нешифрованную форму представления) данных при согласовании протоколов обмена в различных участках сети.

Таким образом, можно сделать вывод, что объектами воздействия внешних угроз безопасности информационных систем могут стать:

- а) персонал, обслуживающий эту информационную систему, или ее пользователи;
- б) аппаратно-программные элементы информационной системы, среди которых можно выделить:
  - аппаратные устройства информационной системы;
  - кабельные системы, включающие внутренние линии связи и силовые кабели;
  - данные, хранящиеся в информационной системе и/или обрабатываемые ею;
  - программное обеспечение информационной системы.

Результатом отказов, сбоев и ошибок работы аппаратного оборудования могут стать потери или искажение данных. В свою очередь, это приводит к нарушению целостности или доступности информации.

Более серьезными угрозами безопасности являются действия персонала, которые могут быть как случайными, так и предна-

меренными. Случайные действия представляют собой результат ошибок, вызванных некомпетентностью, халатностью или болезнью людей. Побудительными мотивами для преднамеренных действий могут стать злонамеренные помыслы пользователей или специалистов, а также их подкуп или вербовка представителями преступных группировок или конкурентов.

Ошибки обслуживающего персонала и пользователей или их умышленные действия могут привести к случайному уничтожению или изменению данных, т. е. к нарушению целостности или доступности информации. Кроме того, только действия людей могут привести к нарушению конфиденциальности информации в результате несанкционированного копирования, уничтожения или подделки информации или же ознакомления посторонних лиц с конфиденциальной информацией, составляющей тайну.

Под источником угроз безопасности информационных систем принято понимать непосредственный генератор или носитель этой угрозы [15]. Из вышеизложенного вытекает, что такими источниками могут быть:

- а) люди из числа пользователей, персонал, обслуживающий информационную систему, или посторонние лица;
- б) технические устройства ввода, хранения, обработки и передачи данных;
- в) программы системные или прикладные;
- г) внешняя среда, генерирующая побочные шумы или сигналы.



Рис. 11.2. Распределение причин потери информации



Для оценки степени значимости этих угроз можно использовать статистические данные [5], свидетельствующие о том, что основными причинами потери информации в коммерческих фирмах являются: сбой жесткого диска и операционной системы, ошибки пользователя, сбой в программах, компьютерные вирусы, природные катаклизмы (рис. 11.2).

### **11.1.3.1. Роль жесткого диска и операционной системы в обеспечении безопасности информации**

Магнитные диски персонального компьютера в настоящее время являются основными носителями информации, предназначенными для длительного и надежного ее хранения. В процессе работы персонального компьютера непрерывно происходит обмен информацией между дисками и оперативной памятью, при этом наиболее интенсивно происходит обмен с жестким диском. Несмотря на высокое качество изготовления дисков и дисковых устройств, в практике регулярной работы на компьютере нередко возникают ситуации, когда не удается прочитать информацию с дисков, происходят нарушения в работе файловой системы, значительно сокращается свободное пространство на дисках или диски оказываются переполненными. Нередко ошибочно удаляются нужные файлы. Эти нарушения в работе дисков могут возникать по следующим причинам:

- при физическом повреждении диска;
- загрязнении магнитной поверхности диска;
- аварийном отключении компьютера;
- несвоевременном извлечении дискет из дисководов;
- перезагрузке операционной системы после аварийного завершения задания;
- воздействии программных вирусов.

Кроме того, при интенсивной эксплуатации компьютера на жестких дисках накапливаются такие изменения в расположении файлов, которые, если не принимать мер, могут привести к существенному замедлению обмена с ними информацией. Это явление получило название фрагментации файлов и объясняется теми принципами, которые положены в основу организации хранения информации при использовании накопителей на магнитных дисках.

Несмотря на то, что внешняя память может быть технически реализована на разных материальных носителях (например, в виде гибкого магнитного диска или магнитной ленты), их объединяет принятый в операционной системе принцип организации хранения логически связанных наборов информации в виде так называемых файлов<sup>1</sup>. Файл служит учетной единицей информации в операционной системе. Любые действия с информацией в MS DOS осуществляются над файлами: запись на диск, вывод на экран, ввод с клавиатуры, печать, считывание информации CD-ROM и пр.

Минимальной единицей пространства диска, которое может быть отведено файлу, является кластер. Самый маленький файл занимает один кластер, большие файлы — несколько десятков кластеров<sup>2</sup>. Файл размещается на диске по кластерам, которые пронумерованы. Эти кластеры могут находиться в разных местах диска, и соответственно файл будет храниться на диске в виде отдельных фрагментов в свободных на момент записи на диск кластерах. В этом случае говорят, что файл фрагментирован. Желательно, чтобы кластеры, выделенные для хранения файла, шли подряд, так как это позволяет сократить время его поиска. Однако это возможно сделать только с помощью специальной программы, и подобная процедура получила название дефрагментации файла. И в том, и в другом случае для организации доступа к файлу операционная система должна иметь сведения о номерах кластеров, где размещается каждый файл. В этом ей помогает FAT-таблица, которая предназначена для хранения информации о размещении файлов на диске и их поиска. Эта таблица хранится на диске в определенном месте. Учитывая ее крайне важную роль в организации файловой системы, предусмотрено создание двух ее экземпляров, т. е. на диске хранятся две одинаковые таблицы — основная и дублирующая. При повреждении

---

<sup>1</sup> Файл — логически связанная совокупность данных или программ, для размещения которой во внешней памяти выделяется именованная область.

<sup>2</sup> Кластер — группа смежных секторов, на которые разбивается жесткий диск при форматировании. Объем сектора составляет 512 байт. На гибком диске обычно 80 дорожек. На каждой дорожке 3,5" диска — 18 секторов. Кластер для гибкого диска — 1...2 сектора (0,512...1 Кбайт). Кластер для жесткого диска может состоять из 4, 8 или 16 секторов (от 2 до 8 Кбайт).

основной таблицы можно восстановить информацию о размещении файлов с помощью дублирующей. Следует заметить, что для ускорения доступа к таблице производится ее предварительная загрузка в оперативную память. Количество ячеек FAT-таблицы определяется количеством кластеров на диске. Каждая ячейка содержит номер кластера. В свою очередь, в каталоге хранятся записи о файлах, где наряду с другими характеристиками указан номер его первого кластера. При необходимости доступа к файлу сначала производится обращение к ячейке FAT-таблицы, адрес которой определяется первым номером, хранящимся в записи о файле (рис. 11.3). В этой ячейке хранится номер второго кластера этого файла. Обратившись к ячейке таблицы, соответствующей номеру второго кластера, операционная система найдет там номер третьего кластера и т. д. Так будет создана цепочка кластеров, где расположен файл. В последней клетке таблицы, завершающей данную цепочку, должен находиться код FFF или FFFF для указания ее конца. Так определяется цепочка кластеров, где хранится файл.

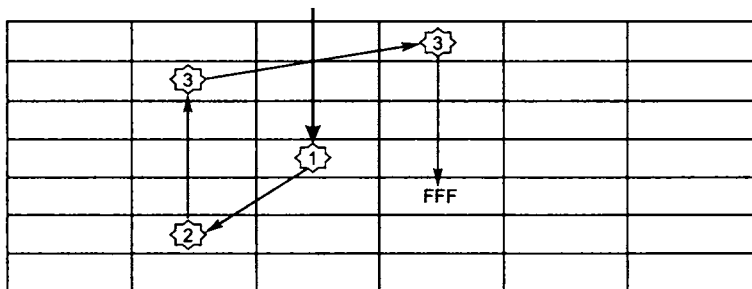


Рис. 11.3. Схема организации файловой системы

Причины сбоев, связанные с работой операционной системы, обуславливаются особенностями взаимодействия оперативной и внешней памяти компьютера, организуемого операционной системой MS Windows (см. п. 5.3.1).

### 11.1.3.2. Компьютерные вирусы и их воздействие на информационные системы

Компьютерный вирус представляет собой специальную программу, созданную с целью нарушения работы программ, порчи файлов и каталогов, создания всевозможных помех в работе на

компьютере. Основанием считать программу компьютерным вирусом является наличие у нее специфических свойств, к числу которых относятся:

- способность самопроизвольно присоединяться к другим программам;
- способность создавать свои копии и внедрять их в файлы, системные области компьютера и в вычислительные сети.

Причины появления и распространения компьютерных вирусов, с одной стороны, скрываются в психологии человеческой личности и ее теневых сторонах (зависти, мести, тщеславии непризнанных творцов, невозможности конструктивно применить свои способности), с другой — обусловлены отсутствием аппаратных средств защиты и противодействия со стороны операционной системы персонального компьютера. Несмотря на принятые во многих странах законы о борьбе с компьютерными преступлениями и разработку специальных программных средств защиты от вирусов, количество новых программных вирусов постоянно растет. Это требует от пользователя персонального компьютера знаний о природе вирусов, способах заражения вирусами и защиты от них.

После запуска программы, содержащей вирус, становится возможным заражение других файлов. Наиболее часто вирусом заражаются загрузочный сектор диска и исполняемые файлы, имеющие расширения EXE, COM, SYS или BAT.

В зависимости от того, какая часть диска оказалась зараженной, различают зараженные диски — это диск, в загрузочном секторе которого находится программа-вирус, или зараженные программы — это программа, содержащая внедренную в нее программу-вирус.

Основными путями проникновения вирусов в компьютер являются съемные диски (гибкие и лазерные), а также компьютерные сети. При заражении компьютера вирусом очень важно своевременно его обнаружить. Для этого следует знать об основных признаках проявления вирусов. К ним можно отнести следующие:

- прекращение работы или неправильная работа ранее успешно функционировавших программ;
- медленная работа компьютера;
- невозможность загрузки операционной системы;

- исчезновение файлов и каталогов или искажение их содержимого;
- изменение даты и времени модификации файлов;
- изменение размеров файлов;
- неожиданное значительное увеличение количества файлов на диске;
- существенное уменьшение размера свободной оперативной памяти;
- вывод на экран непредусмотренных сообщений или изображений;
- подача непредусмотренных звуковых сигналов;
- частые зависания и сбои в работе компьютера.

В настоящее время число известных вирусов превышает 10 000. Для их классификации используются различные признаки.

**В зависимости от среды обитания** вирусы можно разделить на:

1) сетевые вирусы, распространяющиеся по различным компьютерным сетям;

2) файловые вирусы, внедренные главным образом в исполняемые модули, т. е. в файлы, имеющие расширения COM и EXE. Эти вирусы могут внедряться и в другие типы файлов, но, как правило, записанные в таких файлах, они никогда не получают управление и, следовательно, теряют способность к размножению;

3) загрузочные вирусы, внедренные в загрузочный сектор диска (Boot-сектор) или в сектор, содержащий программу загрузки системного диска (Master Boot Record);

4) файлово-загрузочные вирусы, заражающие как файлы, так и загрузочные сектора дисков.

**По способу заражения** вирусы делятся на резидентные и нерезидентные:

1) резидентный вирус при заражении (инфицировании) компьютера оставляет в оперативной памяти свою резидентную часть, которая потом перехватывает обращение операционной системы к объектам заражения (файлам, загрузочным секторам дисков и т. п.) и внедряется в них. Резидентные вирусы находятся в памяти и являются активными вплоть до выключения или перезагрузки компьютера;

2) нерезидентные вирусы не заражают память компьютера и являются активными ограниченное время.

**По степени воздействия** вирусы можно разделить на следующие виды:

1) неопасные. Эти вирусы не мешают работе компьютера, но уменьшают объем свободной оперативной памяти и памяти на дисках. Действия таких вирусов проявляются в каких-либо графических или звуковых эффектах;

2) опасные вирусы, которые могут привести к различным нарушениям в работе компьютера;

3) очень опасные, воздействие которых может привести к потере программ, уничтожению данных, стиранию информации в системных областях диска.

**По особенностям алгоритма** различаются:

1) паразитические вирусы, которые изменяют содержимое файлов и секторов диска и могут быть достаточно легко обнаружены и уничтожены;

2) вирусы-репликаторы, называемые червями, которые распространяются по компьютерным сетям, вычисляют адреса сетевых компьютеров и записывают по этим адресам свои копии;

3) вирусы-невидимки, называемые стелс-вирусами, которые очень трудно обнаружить и обезвредить, так как они перехватывают обращения операционной системы к пораженным файлам и секторам дисков и подставляют вместо своего тела незараженные участки диска;

4) вирусы-мутанты, содержащие алгоритмы шифровки-расшифровки, благодаря которым копии одного и того же вируса не имеют ни одной повторяющейся цепочки байтов. Эти вирусы наиболее трудно обнаруживаются;

5) квазивирусные или «троянские» программы, которые хотя и не способны к самораспространению, но очень опасны, так как, маскируясь под полезную программу, разрушают загрузочный сектор и файловую систему дисков.

Таким образом, воздействие вирусов может привести к отказам аппаратно-программных элементов информационных систем, в результате чего нарушаются целостность и доступность информации.

Наличие многочисленных угроз безопасности информационных систем и возможность причинения этими угрозами значительного, а иногда фатального ущерба, как отдельным субъектам информационных отношений, так и организациям в целом, требует разработки и использования специальных мер по защите информации.

## 11.2. Противодействие угрозам безопасности информационных систем

### 11.2.1. Обеспечение безопасности информации и ее виды

Все вопросы, относящиеся к защите информации, прав субъектов, участвующих в информатизационных процессах, и информатизации в Российской Федерации, регламентируются федеральным законом «Об информации, информатизации и защите информации». В соответствии с ним целями этой защиты являются:

- предотвращение утечки, хищения, утраты, искажения, подделки информации;
- предотвращение угроз безопасности личности, общества, государства;
- предотвращение несанкционированных действий по уничтожению, модификации, искажению, копированию, блокированию информации; предотвращение других форм незаконного вмешательства в информационные ресурсы и информационные системы, обеспечение правового режима документированной информации как объекта собственности.

Очевидно, что реализация этих целей обеспечивает безопасность информации. Однако для этого требуется выполнение большого числа различных по своему характеру мероприятий. Это обусловлено разнообразием и многочисленностью угроз безопасности информационных систем. Поэтому в настоящее время принято считать [15], что обеспечение информационной безопасности представляет собой комплекс мероприятий, проводимых с целью сохранения свойств информации неизменными при воздействии случайных или преднамеренных внешних воздействий.

Как правило, работа по обеспечению информационной безопасности должна включать в себя три этапа: предупреждение угроз безопасности; выявление фактов реализации угроз безопасности; минимизация причиненного ущерба. На каждом из этих этапов могут быть использованы различные по своему характеру меры, которые могут быть сгруппированы следующим образом:

**Правовые меры защиты информации** — действующие в стране законы, указы и другие нормативные акты, регламентирующие

правила обращения с информацией и ответственность за их нарушения, препятствующие тем самым неправомерному ее использованию и являющиеся сдерживающим фактором для потенциальных нарушителей.

**Организационные (административные) меры защиты** — это меры, регламентирующие процессы функционирования системы обработки данных, использование ее ресурсов, деятельность персонала, а также порядок взаимодействия пользователей с системой таким образом, чтобы в наибольшей степени затруднить или исключить возможность реализации угроз безопасности циркулирующей в ней информации.

**Физические меры защиты** — это разного рода механические, электро- или электронно-механические устройства и сооружения, специально предназначенные для создания физических препятствий на возможных путях проникновения и доступа потенциальных нарушителей к компонентам информационных систем и защищаемой информации, а также технические средства визуального наблюдения, связи и охранной сигнализации. При реализации физических мер защиты используются технические (аппаратно-программные) средства — различные электронные устройства и специальные программы, входящие в состав информационных систем. Они выполняют функции защиты информации (авторизацию и аутентификацию пользователей, разграничение доступа к ресурсам, регистрацию событий, криптографическое закрытие информации и т. д.).

Поэтому различают три вида обеспечения информационной безопасности: правовое, организационное и инженерно-техническое.

**Правое обеспечение безопасности информации.** Данный вид обеспечения защиты информации является основой, фундаментом для реализации двух остальных видов обеспечения защиты. Необходимость правового обеспечения информационной безопасности обусловлена возможностью осуществления компьютерных преступлений.

*По своему содержанию, объекту и последствиям* различают преступления экономические, против общественных интересов и против частных интересов.

*По способу совершения* компьютерных преступлений Генеральный Секретариат Интерпола различает:

- несанкционированный доступ и перехват информации;
- изменение компьютерных данных;



- компьютерное мошенничество;
- незаконное копирование программ и баз данных;
- компьютерный саботаж.

Основным методом правового обеспечения является уголовная и административная ответственность. Соответствующие статьи (272, 273 и 274) Уголовного кодекса РФ предусматривают ответственность за неправомерный доступ к компьютерной информации, создание, использование и распространение вредоносных программ для ЭВМ и нарушение правил эксплуатации ЭВМ, системы ЭВМ или их сети.

Административная ответственность наступает при незаконном использовании в коммерческих целях программ или баз данных в случаях, если:

- экземпляры программ или баз данных являются контрафактными в соответствии с законодательством Российской Федерации об авторском праве и смежных правах;
- на экземплярах программ или баз данных указана ложная информация об их изготовителях и о местах производства, а также иная информация, которая может ввести в заблуждение потребителей;
- на экземплярах программ или баз данных уничтожен либо изменен знак охраны авторского права или знак охраны смежных прав, проставленные обладателем авторских или смежных прав.

Эта ответственность предусматривает наложение штрафа на граждан в размере от пяти до десяти минимальных размеров оплаты труда, а на должностных лиц в размере от десяти до двадцати минимальных размеров оплаты труда с конфискацией контрафактных экземпляров программ или баз данных.

При повторных нарушениях размеры штрафов серьезно увеличиваются: штраф на граждан до десяти—двадцати минимальных размеров оплаты труда, а на должностных лиц до тридцати—пятидесяти минимальных размеров оплаты труда с конфискацией контрафактных экземпляров программ или баз данных.

Российским законодательством предусмотрена ответственность для лиц, незаконными методами получивших информацию, которая составляет служебную или коммерческую тайну. Эта ответственность заключается в обязанности возместить причиненные убытки. Такая же обязанность возлагается на работников, разгласивших служебную или коммерческую тайну вопреки

трудовому договору, в том числе контракту, и на контрагентов, сделавших это вопреки гражданско-правовому договору.

Правовое обеспечение защиты информации организуется и осуществляется государством.

**Организационное обеспечение безопасности информации.** Этот вид обеспечения играет особую роль в силу следующих причин:

1) организационные меры обеспечивают исполнение существующих нормативных актов и строятся с учетом существующих правил поведения, принятых в стране и/или организации;

2) воплощение организационных мер требует создания нормативных документов;

3) для эффективного применения организационные меры должны быть поддержаны физическими и техническими средствами;

4) применение и использование технических средств защиты требует соответствующей организационной поддержки.

Организационное обеспечение безопасности информационных систем предполагает организацию и осуществление:

- мероприятий, осуществляемых при проектировании, строительстве и оборудовании вычислительных центров и других объектов систем обработки данных;
- мероприятий по разработке правил доступа пользователей к ресурсам системы (разработка политики безопасности);
- мероприятий, осуществляемых при подборе и подготовке персонала системы;
- надежных систем охраны и пропускного режима;
- учета, хранения, использования и уничтожения документов и носителей с информацией;
- распределения реквизитов разграничения доступа (паролей, ключей шифрования и т. п.);
- явного и скрытого контроля работы пользователей;
- мероприятий, осуществляемых при проектировании, разработке, ремонте и модификациях оборудования и программного обеспечения и т. п.

Анализ содержания этих мероприятий позволяет сделать вывод о том, что они организуются и осуществляются на предприятии (в учреждении и т. п.). Очевидно, что ответственность за этот вид обеспечения БИ несет руководитель этого предприятия. В зависимости от масштабов учреждения выполнение ряда мероприятий может быть возложено на руководителей структурных подразделений и сотрудников.

Осуществления некоторых специфических мероприятий по обеспечению безопасности информации возлагается на системного администратора, администратора базы данных или другого специалиста по информационным технологиям.

**Инженерно-техническое обеспечение безопасности информации.** Этот вид обеспечения реализуется в двух направлениях: а) физической защиты элементов информационных систем и б) программной и аппаратно-программной защиты данных.

Первое направление реализуется путем физической защиты кабельных сетей и систем электроснабжения, а также защиты данных путем их дублирования. Кабельная система остается уязвимым местом большинства локальных вычислительных сетей. В связи с этим кабельной системе должно уделяться особое внимание с самого момента проектирования сети. Наилучшим способом защиты кабеля от физических (а иногда и температурных, и химических воздействий, например, в производственных цехах) является прокладка кабелей с использованием в различной степени защищенных коробов. При прокладке сетевого кабеля вблизи источников электромагнитного излучения необходимо выполнять следующие требования:

а) неэкранированная витая пара должна отстоять минимум на 15—30 см от электрического кабеля, розеток, трансформаторов и т. д.;

б) требования к коаксиальному кабелю менее жесткие — расстояние до электрической линии или электроприборов должно быть не менее 10—15 см.

Наиболее надежным средством предотвращения потерь информации при кратковременном отключении электроэнергии в настоящее время является установка источников бесперебойного питания. Различные по своим техническим и потребительским характеристикам, подобные устройства могут обеспечить питание всей локальной сети или отдельного компьютера в течение промежутка времени, достаточного для восстановления подачи напряжения или для сохранения информации на магнитные носители. Большинство источников бесперебойного питания одновременно выполняет функции и стабилизатора напряжения, что является дополнительной защитой от скачков напряжения в сети. Многие современные сетевые устройства — серверы, концентраторы, мосты и т. д. — оснащены собственными дублированными системами электропитания.

За рубежом крупные корпорации имеют собственные аварийные электрогенераторы или резервные линии электропитания. Эти линии подключены к разным подстанциям, и при выходе из строя одной из них электроснабжение осуществляется с резервной подстанции.

Что касается физической защиты данных, здесь наиболее эффективным средством является дублирование информации. С этой целью для хранения резервных копий могут использоваться дополнительные устройства внешней памяти, к числу которых относятся CD-ROM, устройства ZIP, Jazz, стримеры, а в крупных организациях — серверы резервного копирования.

Для осуществления программной и аппаратно-программной защиты данных используются разнообразные средства, среди которых можно выделить:

1) средства предупреждения угроз безопасности, включающие:

- системы разграничения доступа;
- криптографические средства;
- антивирусные мониторы и фильтры;
- межсетевые экраны (firewall) и шлюзы;

2) средства выявления угроз безопасности, к которым относятся:

- системы аудита и «ревизоры»;
- антивирусные сканеры;
- системные и сетевые мониторы;

3) средства минимизации ущерба:

- системы резервного копирования;
- средства обеспечения отказоустойчивости.

Применение многих из перечисленных средств требует специальных знаний и осуществляется специалистами. В то же самое время пользователь должен уметь пользоваться некоторыми из них.

### **11.2.2. Использование разграничения доступа для защиты информации**

Разграничение доступа является достаточно эффективным средством предупреждения возможного ущерба вследствие нарушения целостности или конфиденциальности информации. В том случае, если доступ к самому компьютеру или к его ре-

сурсам может получить пользователь, который имеет злой умысел или недостаточный уровень подготовки, он может случайно или преднамеренно исказить информацию или уничтожить ее полностью или частично. Это же обстоятельство может привести к раскрытию закрытой информации или несанкционированному тиражированию открытой, например, программ, баз данных, разного рода документации, литературных произведений и т. д. в нарушение прав собственников информации, авторских прав и т. п.

С точки зрения разграничения доступа в информационных системах следует различать субъекты доступа и объекты доступа. В число субъектов доступа могут войти либо персонал информационной системы, либо посторонние лица. Объектами доступа являются аппаратно-программные элементы информационных систем. Чаще всего в качестве объектов доступа рассматриваются файлы (в том числе папки и файлы программ). Доступ к объекту может рассматриваться либо как чтение (получение информации из него), либо как изменение (запись информации в него). Тогда виды доступа определяются следующими возможными сочетаниями этих операций:

- ни чтение, ни изменение;
- только чтение;
- только изменение;
- и чтение, и изменение.

Очевидно, что различие функциональных обязанностей субъектов обуславливает необходимость предоставления им соответствующих видов доступа.

Разграничение доступа реализуется операционной системой. Операционные системы семейства MS Windows, начиная с версий NT/2000/XP/.Net, обеспечивают полноценное разграничение доступа на уровне пользователей. При этом предполагается, что рабочие станции под управлением этих операционных систем могут состоять в составе рабочей группы или домена. Такие группы, как правило, организуются в соответствии со структурой организации (например, бухгалтерия, отдел кадров, склад, др. подразделения). На рабочих станциях, членах рабочих групп доступ пользователей и глобальные параметры конфигурируются на самой рабочей станции. Управление доступом пользователей и глобальными параметрами на членах домена осуществляется на двух уровнях: локальной системы и домена. На отдельных компьютерах доступ пользователей конфигурируют на уровне

локальной системы, а одновременно для нескольких систем или ресурсов, входящих в домен, — на уровне домена.

Права доступа пользователя определяются руководителем организации и прописываются на рабочей станции системным администратором (администратором домена). Процедура проверки прав доступа включает авторизацию и аутентификацию. Авторизация предполагает проверку уровня доступа к объекту, а аутентификация — проверку подлинности пользователя. Для аутентификации обычно используются имя пользователя (login) и пароль (password). Операционные системы MS Windows NT/2000/XP/.Net, имея развитую систему прав доступа и систему привилегий, обеспечивают полноценную аутентификацию. Кроме того, предоставляют авторизованному пользователю возможность шифрования данных.

Системный администратор осуществляет разграничение прав доступа в соответствии с заданной системной политикой, которая предполагает:

- ограничения на минимальную длину, сложность и срок действия пароля;
- требование уникальности паролей;
- блокировку пользователя при неудачной аутентификации;
- ограничение времени и места работы пользователя.

Система ограничения прав доступа не может дать полной гарантии безопасности информации. Дело в том, что злоумышленник может получить или подобрать пароль легального пользователя. Кроме того, опытный специалист может обойти систему разграничения доступа. Средством обнаружения несанкционированного доступа к ресурсам служат системы аудита, которые автоматически фиксируют доступ к файлам и папкам и системные события.

### 11.2.3. Антивирусная защита

**Поиск и уничтожение известных вирусов.** При поиске и уничтожении известных вирусов наиболее распространенным является метод сканирования. Указанный метод заключается в выявлении компьютерных вирусов по их уникальному фрагменту программного кода. Для этого создается некоторая база данных сканирования с фрагментами кодов известных компьютерных вирусов. Обнаружение вирусов осуществляется путем сравнения

данных памяти компьютера с фиксированными кодами базы данных сканирования. В случае выявления и идентификации кода нового вируса он может быть введен в базу данных сканирования. В этом случае существует возможность корректного восстановления (обеззараживания) зараженных файлов и областей.

Антивирусные программы, выявляющие известные компьютерные вирусы, называются сканерами или детекторами. Программы, включающие функции восстановления зараженных файлов, называют фагами, докторами или дезинфекторами. Примером сканера-полифага является знакомая программа Aidstest. Принято разделять сканеры на следующие:

- транзитные, периодически запускаемые для выявления и ликвидации вирусов. Такие программы осуществляют поиск вирусов по запросу пользователя. Этот способ получил название «сканирование по запросу» («on-demand»). В этом режиме антивирусная программа неактивна до тех пор, пока не будет вызвана пользователем из командной строки, командного файла или программы-расписания (system scheduler);
- резидентные (постоянно находящиеся в оперативной памяти), проверяющие заданные области памяти системы при возникновении связанных с ними событий (например, проверка файла при его копировании или переименовании). Эти программы осуществляют так называемое «сканирование на лету» («real-time», «on-the-fly») — постоянную проверку на вирусы объектов, к которым происходит обращение (запуск, открытие, создание и т. п.). В этом режиме антивирус постоянно активен, он присутствует в памяти «резидентно» и проверяет объекты без запроса пользователя.

К недостаткам сканеров следует отнести то, что они позволяют обнаружить вирусы, которые уже проникли в вычислительные системы и изучены. Для эффективной работы сканеров необходимо оперативно пополнять базу данных сканирования. Однако с увеличением объема базы данных сканирования и числа различных типов искомых вирусов снижается скорость антивирусной проверки. Само собой, если время сканирования будет приближаться ко времени восстановления, то необходимость в антивирусном контроле может стать не столь актуальной.

Некоторые вирусы (мутанты и полиморфные) кодируют или видоизменяют свой программный код. Это затрудняет или

исключает возможность обнаружения вирусов методом сканирования.

**Поиск и уничтожение неизвестных вирусов.** Выявление и ликвидация неизвестных вирусов необходимы для защиты от вирусов, пропущенных на первом уровне антивирусной защиты. Наиболее эффективным методом является контроль целостности системы (обнаружение изменений). Данный метод заключается в проверке и сравнении текущих параметров вычислительной системы с эталонными, соответствующими ее незараженному состоянию. Понятно, что контроль целостности не является прерогативой исключительно системы антивирусной защиты. Он обеспечивает защищенность информационного ресурса от несанкционированных модификации и удаления в результате различного рода нелегитимных воздействий, сбоев и отказов системы и среды.

Для реализации указанных функций используются программы, называемые *ревизорами*. Работа ревизора состоит из двух этапов: фиксирование эталонных характеристик вычислительной системы (в основном диска) и периодическое сравнение их с текущими характеристиками. Обычно контролируемыми характеристиками являются контрольная сумма, длина, время, атрибут «только для чтения» файлов, дерево каталогов, сбойные кластеры, загрузочные сектора дисков. В сетевых системах могут накапливаться среднестатистические параметры функционирования подсистем (в частности, исторический профиль сетевого трафика), которые сравниваются с текущими.

Ревизоры, как и сканеры, делятся на транзитные и резидентные.

К недостаткам ревизоров, в первую очередь резидентных, относят создаваемые ими всякие неудобства и трудности в работе пользователя. Например, многие изменения параметров системы вызваны не вирусами, а работой системных программ или действиями пользователя-программиста. По этой же причине ревизоры не используют для контроля зараженности текстовых файлов, которые постоянно меняются. Таким образом, необходимо соблюдение некоторого баланса между удобством работы и контролем целостности системы.

Ревизоры обеспечивают высокий уровень выявления неизвестных компьютерных вирусов, однако они не всегда обеспечивают корректное лечение зараженных файлов. Для лечения зараженных файлов неизвестными вирусами обычно используются



эталонные характеристики файлов и предполагаемые способы их заражения. Кроме этого, ревизоры не определяют зараженные файлы, создаваемые или копируемые в систему. Примерами ревизоров являются программы: MSAV ОС MS-DOS и ADInf фирмы «ДиалогНаука».

Разновидностью контроля целостности системы является метод программного самоконтроля, именуемый вакцинацией. Идея метода состоит в присоединении к защищаемой программе модуля (*вакцины*), контролирующего характеристики программы, обычно ее контрольную сумму.

Помимо статистических методов контроля целостности, для выявления неизвестных и маскирующихся вирусов используют эвристические методы. Они позволяют выявить по известным признакам (определенным в базе знаний системы) некоторые маскирующиеся или новые модифицированные вирусы известных типов. В качестве примера признака вируса можно привести код, устанавливающий резидентный модуль в памяти, меняющий параметры таблицы прерываний и др. Программный модуль, реализующий эвристический метод обнаружения вирусов, называют *эвристическим анализатором*. Примером сканера с эвристическим анализатором является программа Dr Web фирмы «ДиалогНаука».

К недостаткам эвристических анализаторов можно отнести ошибки 1- и 2-го рода: ложные срабатывания и пропуск вирусов. Соотношение указанных ошибок зависит от уровня эвристики. Понятно, что если для обнаруженного эвристическим анализатором компьютерного вируса сигнатура в базе данных сканирования отсутствует, то лечение зараженных данных может быть некорректным.

**Блокировка проявления вирусов.** Блокировка проявления вирусов предназначена для защиты от деструктивных действий и размножения компьютерных вирусов, которым удалось преодолеть первые два уровня защиты. Методы основаны на перехвате характерных для вирусов функций. Известны два вида указанных антивирусных средства:

- программы-фильтры;
- аппаратные средства контроля.

*Программы-фильтры*, называемые также резидентными сторожами и мониторами, постоянно находятся в оперативной памяти и перехватывают заданные прерывания с целью контроля

подозрительных действий. При этом они могут блокировать «опасные» действия или выдавать запрос пользователю.

Действия, подлежащие контролю, могут быть следующими: модификация главной загрузочной записи (MBR) и загрузочных записей логических дисков и ГМД, запись по абсолютному адресу, низкоуровневое форматирование диска, оставление в оперативной памяти резидентного модуля и др. Как и ревизоры, фильтры часто являются «навязчивыми» и создают определенные неудобства в работе пользователя. Примером фильтра является программа Vsafe ОС MS-DOS.

Встроенные аппаратные средства ПК обеспечивают контроль модификации системного загрузчика и таблицы разделов жесткого диска, находящихся в главном загрузочном секторе диска (MBR). Включение указанных возможностей в ПК осуществляется с помощью программы Setup, расположенной в ПЗУ. Следует указать, что программу Setup можно обойти в случае замены загрузочных секторов путем непосредственного обращения к портам ввода-вывода контроллеров жесткого и гибкого дисков.

Наиболее полная защита от вирусов может быть обеспечена с помощью специальных контроллеров аппаратной защиты. Такой контроллер подключается к ISA-шине ПК и на аппаратном уровне контролирует *все* обращения к дисковой подсистеме компьютера. Это не позволяет вирусам маскировать себя. Контроллер может быть сконфигурирован так, чтобы контролировать отдельные файлы, логические разделы, «опасные» операции и т. д. Кроме того, контроллеры могут выполнять различные дополнительные функции защиты, например обеспечивать разграничение доступа и шифрование. Примером специальных контроллеров аппаратной защиты является плата Sheriff предприятия Фом-Софт. К недостаткам указанных контроллеров, как ISA-плат, относятся отсутствие системы автоконфигурирования, и как следствие, возможность возникновения конфликтов с некоторыми системными программами, в том числе антивирусными.

Одним из наиболее распространенных антивирусных средств является система AVP, созданная в России «Лабораторией Касперского». Она первоначально представляла тандем двух программ: сканера и ревизора. В настоящее время AVP претерпела существенные усовершенствования и включает все основные элементы антивирусной программной защиты компьютера и локальной сети. Имеются версии для DOS, Windows 3.1x, Windows 95, Windows NT, Novell NetWare.

По мировым оценкам последние 5 лет система AVP неоднократно входила в тройку наилучших в мире антивирусных продуктов поиска и удаления компьютерных вирусов в автономном компьютере. Система имеет одну из самых больших баз данных сканирования. Постоянное пополнение этой базы позволяет системе обнаруживать практически все полиморфные вирусы. Следует отметить, что программный продукт AVP сертифицирован Гостехкомиссией и фирмой Microsoft.

В настоящий момент «Лаборатория Касперского» выпускает следующие программные средства антивирусной защиты:

1) сканеры-полифаги AVP с эвристическим анализатором (AVP Code Analyzer) для DOS (пакетный вариант — AVP Lite), Windows 95, Windows NT Workstation, Windows NT Server (AVP F-Secure) и Novell Netware (AVPN);

2) ревизор диска (CRC-сканер) AVP Inspector для Windows 95 и Windows NT;

3) фильтры (резидентные мониторы) AVP Monitor для Windows 95, Windows NT Workstation, Windows NT Server (AVP F-Secure) и Novell Netware (AVPN);

4) электронный мультимедийный справочник по вирусологии AVP Virus Encyclopedia;

5) различные утилиты и дополнительные продукты, как-то: макрополифаг AVP Macro Stop (AVP MacroKiller), сканер сообщений электронной почты на рабочих станциях AVP Mail Checker для Windows 95 и др.

**Рекомендации пользователя по обеспечению безопасности информации.** В связи с тем, что наибольшая часть угроз безопасности исходит от пользователя [15], представляется целесообразным дать некоторые рекомендации, позволяющие предотвратить возможный ущерб.

1. Использование антивирусных программ:

- регулярная проверка всех дисков компьютера;
- регулярное обновление антивирусных баз.

2. Использование мер предосторожности при работе с входящей электронной почтой:

- не открывать письма с вложениями от незнакомых людей;
- проверять каждый файл перед открытием;
- использовать антивирусные мониторы;
- не использовать предварительный просмотр сообщений;
- отключать макросы в Word/Excel.

3. Использование мер предосторожности при работе в локальной сети:

- установка атрибутов «только для чтения» и «только для запуска»;
- использование антивирусных мониторов;
- проверка нового ПО на специальном компьютере, не подключенном к ЛВС.

4. Использование лицензионного ПО.

5. Хранение дистрибутивов на носителях, защищенных от записи.

6. Использование мер предосторожности при работе с непроверенными файлами:

- использование антивирусных мониторов;
- ограничение круга пользователей компьютером.

7. Регулярное создание резервных копий.

### **Контрольные вопросы**

1. Раскройте понятие «безопасность информации».
2. Классификация угроз безопасности информационных систем.
3. Компьютерные вирусы и их классификация.
4. Цели защиты информации.
5. Виды мер защиты информации.

# Литература

---

1. Microsoft® Office Excel 2003 (11.6355.6360). Part of Microsoft Office Professional Edition 2003. © Корпорация Майкрософт (Microsoft Corporation), 1985—2003.
2. Введение в правовую информатику. Справочные правовые системы КонсультантПлюс: Учебник для вузов / Под общ. ред. Д. Б. Новикова, В. Л. Камынина. 3-е изд., доп. и испр. М: ООО НПО «Вычислительная математика и информатика», 2000.
3. *Вентцель Е. С.* Исследование операций: задачи, принципы, методология. 2-е изд., стер. М.: Наука, 1988.
4. *Гарнаев А. Ю.* Excel, VBA, Internet в экономике и финансах. СПб.: БХВ-Петербург, 2003.
5. *Гейн А. Г., Сенокосов А. И.* Справочник по информатике для школьников. Екатеринбург: У-Фактория, 2003.
6. *Дженнингс, Роджер.* Использование Microsoft Access 2002. Специальное издание / Пер. с англ. М.: Издательский дом «Вильямс», 2002.
7. *Додж М., Стинсон К.* Эффективная работа с Microsoft Excel 2000. СПб.: Питер, 2003.
8. *Еремин Е. А.* Популярные лекции об устройстве компьютера. СПб.: БХВ-Петербург, 2003.
9. Информатика. Базовый курс / С. В. Симонович и др. СПб.: Питер, 2000.
10. Информатика: Учебник / Под ред. проф. Н. В. Макаровой. М: Финансы и статистика, 1998.
11. *Крупник А.* Поиск в Интернете: самоучитель (знакомство, работа, развлечение). СПб.: Питер, 2001.
12. *Ожегов С. И.* Словарь русского языка: ок. 57 000 слов / Под ред. докт. филол. наук, проф. Н. Ю. Шведовой. 16-е изд., испр. М.: Рус. яз., 1984.
13. *Острейковский В. А.* Информатика: Учеб. для вузов. М: Высш. шк., 2001.
14. *Семакин И. Г., Вараксин Г. С.* Информатика. Структурированный конспект базового курса. М.: Лаборатория Базовых Знаний, 2001.
15. Федеральный закон от 20 февраля 1995 г. № 24-ФЗ «Об информации, информатизации и защите информации».

## Функции табличного процессора MS Excel

### Функции баз данных

**ДСРЗНАЧ** — возвращает среднее значение выбранных записей базы данных.

**БСЧЁТ** — подсчитывает количество числовых ячеек в базе данных.

**БСЧЁТА** — подсчитывает количество непустых ячеек в базе данных.

**БИЗВЛЕЧЬ** — извлекает из базы данных одну запись, удовлетворяющую заданному условию.

**ДМАКС** — возвращает максимальное значение среди выделенных записей базы данных.

**ДМИН** — возвращает минимальное значение среди выделенных записей базы данных.

**БДПРОИЗВЕД** — перемножает значения определенного поля в записях базы данных, удовлетворяющих условию.

**ДСТАНДОТКЛ** — оценивает стандартное отклонение по выборке из выделенных записей базы данных.

**ДСТАНДОТКЛП** — вычисляет стандартное отклонение по генеральной совокупности из выделенных записей базы данных.

**БДСУММ** — суммирует числа в поле для записей базы данных, удовлетворяющих условию.

**БДДИСП** — оценивает дисперсию по выборке из выделенных записей базы данных

**БДДИСПП** — вычисляет дисперсию по генеральной совокупности из выделенных записей базы данных.

**ПОЛУЧИТЬ.ДАННЫЕ.СВОДНОЙ.ТАБЛИЦЫ** — возвращает данные, хранящиеся в сводной таблице.

### Функции даты и времени

**ДАТА** — возвращает заданную дату в числовом формате Microsoft Excel.

**ДАТАЗНАЧ** — преобразует дату из текстового формата в числовой.

**ДЕНЬ** — преобразует дату в числовом формате в день месяца.

**ДНЕЙ360** — вычисляет количество дней между двумя датами на основе 360-дневного года.

**ДАТАМЕС** — возвращает дату в числовом формате, отстоящую на заданное число месяцев вперед или назад от начальной даты.

**КОНМЕСЯЦА** — возвращает дату в числовом формате для последнего дня месяца, отстоящего вперед или назад на заданное число месяцев.

**ЧАС** — преобразует дату в числовом формате в часы.

**МИНУТЫ** — преобразует дату в числовом формате в минуты.

**МЕСЯЦ** — преобразует дату в числовом формате в месяцы.

**ЧИСТРАБДНИ** — возвращает количество рабочих дней между двумя датами.

**ТДАТА** — возвращает текущую дату и время в числовом формате.

**СЕКУНДЫ** — преобразует дату в числовом формате в секунды.

**ВРЕМЯ** — возвращает заданное время в числовом формате.

**ВРЕМЗНАЧ** — преобразует время из текстового формата в числовой формат.

**СЕГОДНЯ** — возвращает числовой формат текущей даты.

**ДЕНЬНЕД** — преобразует дату в числовом формате в день недели.

**НОМНЕДЕЛИ** — преобразует числовое представление в число, которое указывает, на какую неделю года приходится указанная дата.

**РАБДЕНЬ** — возвращает дату в числовом формате, отстоящую вперед или назад на заданное количество рабочих дней.

**ГОД** — преобразует дату в числовом формате в год.

**ДОЛЯГОДА** — возвращает долю года, которую составляет количество дней между начальной и конечной датами.

## **Инженерные функции**

**БЕССЕЛЬ.І** — возвращает модифицированную функцию Бесселя  $I_p(x)$ .

**БЕССЕЛЬ.Ј** — возвращает функцию Бесселя  $J_p(x)$ .

**БЕССЕЛЬ.К** — возвращает модифицированную функцию Бесселя  $K_p(x)$ .

**БЕССЕЛЬ.У** — возвращает функцию Бесселя  $Y_p(x)$ .

**ДВ.В.ДЕС** — преобразует двоичное число в десятичное.

- ДВ.В.ШЕСТИ** — преобразует двоичное число в шестнадцатеричное.
- ДВ.В.ВОСЬМ** — преобразует двоичное число в восьмеричное.
- КОМПЛЕКСН** — преобразует коэффициенты при вещественной и мнимой частях комплексного числа в комплексное число.
- ПРЕОБР** — преобразует число из одной системы мер в другую.
- ДЕС.В.ДВ** — преобразует десятичное число в двоичное.
- ДЕС.В.ШЕСТИ** — преобразует десятичное число в шестнадцатеричное.
- ДЕС.В.ВОСЬМ** — преобразует десятичное число в восьмеричное.
- ДЕЛЬТА** — проверяет равенство двух значений.
- ФОШ** — возвращает функцию ошибки.
- ДФОШ** — возвращает дополнительную функцию ошибки.
- ПОРОГ** — проверяет, не превышает ли данное число порогового значения.
- ШЕСТИ.В.ДВ** — преобразует шестнадцатеричное число в двоичное.
- ШЕСТИ.В.ДЕС** — преобразует шестнадцатеричное число в десятичное.
- ШЕСТИ.В.ВОСЬМ** — преобразует шестнадцатеричное число в восьмеричное.
- МНИМ.ABS** — возвращает абсолютную величину (модуль) комплексного числа.
- МНИМ.ЧАСТЬ** — возвращает коэффициент при мнимой части комплексного числа.
- МНИМ.АРГУМЕНТ** — возвращает значение аргумента комплексного числа (тета) — угол, выраженный в радианах.
- МНИМ.СОПРЯЖ** — возвращает комплексно-сопряженное комплексное число.
- МНИМ.COS** — возвращает косинус комплексного числа.
- МНИМ.ДЕЛ** — возвращает частное от деления двух комплексных чисел.
- МНИМ.EXP** — возвращает экспоненту комплексного числа.
- МНИМ.LN** — возвращает натуральный логарифм комплексного числа.
- МНИМ.LOG10** — возвращает обычный (десятичный) логарифм комплексного числа.
- МНИМ.LOG2** — возвращает двоичный логарифм комплексного числа.



**МНИМ.СТЕПЕНЬ** — возвращает комплексное число, возведенное в целую степень.

**МНИМ.ПРОИЗВЕД** — возвращает произведение двух комплексных чисел.

**МНИМ.ВЕЩ** — возвращает коэффициент при вещественной части комплексного числа.

**МНИМ.SIN** — возвращает синус комплексного числа.

**МНИМ.КОРЕНЬ** — возвращает значение квадратного корня из комплексного числа.

**МНИМ.РАЗН** — возвращает разность двух комплексных чисел.

**МНИМ.СУММ** — возвращает сумму комплексных чисел.

**ВОСЬМ.В.ДВ** — преобразует восьмеричное число в двоичное.

**ВОСЬМ.В.ДЕС** — преобразует восьмеричное число в десятичное.

**ВОСЬМ.В.ШЕСТН** — преобразует восьмеричное число в шестнадцатеричное.

## **Финансовые функции**

**НАКОПДОХОД** — возвращает накопленный доход по ценным бумагам с периодической выплатой процентов.

**НАКОПДОХОДПОГАШ** — возвращает накопленный доход по ценным бумагам, процент по которым выплачивается в срок вступления в силу.

**АМОРИУМ** — возвращает величину амортизации для каждого периода, используя коэффициент амортизации.

**АМОРИУВ** — возвращает величину амортизации для каждого отчетного периода.

**ДНЕЙКУПОНДО** — возвращает количество дней между началом периода купона и датой соглашения.

**ДНЕЙКУПОН** — возвращает число дней в периоде купона, который содержит дату соглашения.

**ДНЕЙКУПОНПОСЛЕ** — возвращает число дней от даты соглашения до срока следующего купона.

**ДАТАКУПОНПОСЛЕ** — возвращает следующую дату купона после даты соглашения.

**ЧИСЛКУПОН** — возвращает количество купонов, которые могут быть оплачены между датой соглашения и сроком вступления в силу.

**ДАТАКУПОНДО** — возвращает предыдущую дату купона перед датой соглашения.

**ОБЩПЛАТ** — возвращает общую выплату, произведенную между двумя периодическими выплатами.

**ОБЩДОХОД** — возвращает общую выплату по займу между двумя периодами.

**ФУО** — возвращает амортизацию имущества на заданный период, используя метод постоянного учета амортизации.

**ДДОБ** — возвращает величину амортизации имущества для указанного периода при использовании метода двукратного учета амортизации или иного явно указанного метода.

**СКИДКА** — возвращает норму скидки для ценных бумаг.

**РУБЛЬ.ДЕС** — преобразует цену в рублях, выраженную в виде дроби, в цену в рублях, выраженную десятичным числом.

**РУБЛЬ.ДРОБЬ** — преобразует цену в рублях, выраженную десятичным числом, в цену в рублях, выраженную в виде дроби.

**ДЛИТ** — возвращает ежегодную продолжительность действия ценных бумаг с периодическими выплатами по процентам.

**ЭФФЕКТ** — возвращает действующие ежегодные процентные ставки.

**БС** — возвращает будущее значение вклада.

**БЗРАСПИС** — возвращает будущее значение начального предложения после учета ряда сложных процентных ставок.

**ИНОРМА** — возвращает ставку доходности полностью обеспеченной ценной бумаги.

**ПРПЛТ** — возвращает величину выплаты прибыли на вложения за данный период.

**ВСД** — возвращает внутреннюю ставку доходности (отдачи) для серии потоков денежных средств.

**ПРОЦПЛАТ** — вычисляет выплаты за указанный период инвестиции.

**МДЛИТ** — возвращает модифицированную длительность Макелля для ценных бумаг с предполагаемой номинальной стоимостью 100 рублей.

**МВСД** — возвращает внутреннюю ставку доходности, при которой положительные и отрицательные денежные потоки имеют разную ставку.

**НОМИНАЛ** — возвращает номинальную годовую процентную ставку.

**КПЕР** — возвращает общее количество периодов выплаты для данного вклада.

**ЧПС** — возвращает чистую приведенную стоимость инвестиции, основанной на серии периодических денежных потоков и ставке дисконтирования.

**ЦЕНАПЕРВНЕРЕГ** — возвращает цену за 100 рублей нарицательной стоимости ценных бумаг с нерегулярным первым периодом.

**ДОХОДПЕРВНЕРЕГ** — возвращает доход по ценным бумагам с нерегулярным первым периодом.

**ЦЕНАПОСЛНЕРЕГ** — возвращает цену за 100 рублей нарицательной стоимости ценных бумаг с нерегулярным последним периодом.

**ДОХОДПОСЛНЕРЕГ** — возвращает доход по ценным бумагам с нерегулярным последним периодом.

**ПЛТ** — возвращает величину выплаты за один период годовой ренты.

**ОСПЛТ** — возвращает величину выплат на основной капитал для вклада в заданный период.

**ЦЕНА** — возвращает цену за 100 рублей нарицательной стоимости ценных бумаг, по которым производится периодическая выплата процентов.

**ЦЕНАСКИДКА** — возвращает цену за 100 рублей нарицательной стоимости ценных бумаг, на которые сделана скидка.

**ЦЕНАПОГАШ** — возвращает цену за 100 рублей нарицательной стоимости ценных бумаг, по которым выплачивается прибыль в момент вступления в силу.

**ПС** — возвращает приведенную (к настоящему моменту) стоимость инвестиции.

**СТАВКА** — возвращает процентную ставку по аннуитету за один период.

**ПОЛУЧЕНО** — возвращает сумму, полученную в срок вступления в силу полностью обеспеченных ценных бумаг.

**АПЛ** — возвращает величину непосредственной амортизации имущества за один период.

**АСЧ** — возвращает величину амортизации актива за данный период, рассчитанную методом «суммы (годовых) чисел».

**РАВНОКЧЕК** — возвращает эквивалентный облигации доход по казначейскому чеку.

**ЦЕНАКЧЕК** — возвращает цену за 100 рублей нарицательной стоимости для казначейского чека.

**ДОХОДКЧЕК** — возвращает доход по казначейскому чеку.

**ПУО** — возвращает величину амортизации имущества для явно указанного или соответствующего периода при использовании метода разового учета амортизации.

**ЧИСТВНДОХ** — возвращает внутреннюю ставку доходности запланированных непериодических денежных потоков.

**ЧИСТНЗ** — возвращает чистую текущую стоимость инвестиции, вычисляемую на основе ряда поступлений наличных, которые не обязательно являются периодическими.

**ДОХОД** — возвращает доход от ценных бумаг, по которым производятся периодические выплаты процентов.

**ДОХОДСКИДКА** — возвращает годовой доход по ценным бумагам, на которые сделана скидка. Пример — казначейские чеки.

**ДОХОДПОГАШ** — возвращает годовой доход от ценных бумаг, процент по которым выплачивается в срок погашения.

## Информационные функции

**ЯЧЕЙКА** — возвращает информацию о формате, местоположении или содержимом ячейки.

**ТИП.ОШИБКИ** — возвращает номер, соответствующий одному из типов ошибок Microsoft Excel.

**ИНФОРМ** — возвращает информацию о текущей операционной среде.

**ЕПУСТО** — возвращает логическое значение ИСТИНА, если аргумент является ссылкой на пустую ячейку.

**ЕОШ** — возвращает логическое значение ИСТИНА, если аргумент ссылается на любое значение ошибки, кроме #Н/Д.

**ЕОШИБКА** — возвращает логическое значение ИСТИНА, если аргумент ссылается на любое значение ошибки.

**ЕЧЁТН** — возвращает логическое значение ИСТИНА, если аргумент — четное число.

**ЕЛОГИЧ** — возвращает логическое значение ИСТИНА, если аргумент ссылается на логическое значение.

**ЕНД** — возвращает логическое значение ИСТИНА, если аргумент ссылается на значение ошибки #Н/Д (значение недоступно).

**ЕНЕТЕКСТ** — возвращает логическое значение ИСТИНА, если аргумент ссылается на значение, которое не является текстом.

**ЕЧИСЛО** — возвращает логическое значение ИСТИНА, если аргумент ссылается на число.

**ЕНЕЧЁТ** — возвращает логическое значение ИСТИНА, если аргумент — нечетное число.

**ЕССЫЛКА** — возвращает логическое значение ИСТИНА, если аргумент ссылается на ссылку.

**ЕТЕКСТ** — возвращает логическое значение ИСТИНА, если аргумент ссылается на текст.

**Ч** — возвращает значение, преобразованное в число.

**НД** — возвращает значение ошибки #Н/Д.

**ТИП** — возвращает тип значения.

## **Логическая**

**И** — возвращает значение ИСТИНА, если все аргументы имеют значение ИСТИНА.

**ЛОЖЬ** — возвращает логическое значение ЛОЖЬ.

**ЕСЛИ** — выполняет проверку условия.

**НЕ** — меняет на противоположное логическое значение своего аргумента.

**ИЛИ** — возвращает ИСТИНА, если хотя бы один аргумент имеет значение ИСТИНА.

**ИСТИНА** — возвращает логическое значение ИСТИНА.

## **Функции ссылки и автоподстановки**

**АДРЕС** — возвращает ссылку на отдельную ячейку рабочего листа в виде текста.

**ОБЛАСТИ** — возвращает количество областей в ссылке.

**ВЫБОР** — выбирает значение из списка значений по индексу.

**СТОЛБЕЦ** — возвращает номер столбца, на который указывает ссылка.

**ЧИСЛСТОЛБ** — возвращает количество столбцов в массиве или ссылке.

**ГПР** — ищет значение в первой строке массива и возвращает значение из ячейки в найденном столбце и указанной строке.

**ГИПЕРССЫЛКА** — создает ссылку, открывающую документ, находящийся на жестком диске, сервере сети или в Интернете.

**ИНДЕКС** — использует индекс для выбора значения из ссылки или массива.

**ДВССЫЛ** — возвращает ссылку, заданную текстовым значением.

**ПРОСМОТР** — ищет значения в векторе или массиве.

**ПОИСКПОЗ** — ищет значения в ссылке или массиве.

**СМЕЩ** — возвращает смещение ссылки относительно заданной ссылки.

**СТРОКА** — возвращает номер строки, определяемой ссылкой.

**ЧСТРОК** — возвращает количество строк в ссылке.

**ТРАНСП** — возвращает транспонированный массив.

**ВПР** — ищет значение в первом столбце массива и возвращает значение из ячейки в найденной строке и указанном столбце.

## **Арифметические и тригонометрические функции**

**ABS** — возвращает модуль (абсолютную величину) числа.

**ACOS** — возвращает арккосинус числа.

**ACOSH** — возвращает гиперболический арккосинус числа.

**ASIN** — возвращает арксинус числа.

**ASINH** — возвращает гиперболический арксинус числа.

**ATAN** — возвращает арктангенс числа.

**ATAN2** — возвращает арктангенс для заданных координат  $x$  и  $y$ .

**ATANH** — возвращает гиперболический арктангенс числа.

**ОКРВВЕРХ** — округляет число до ближайшего целого или до ближайшего кратного указанному значению.

**ЧИСЛКОМБ** — возвращает количество комбинаций для заданного числа объектов.

**COS** — возвращает косинус числа.

**COSH** — возвращает гиперболический косинус числа.

**ГРАДУСЫ** — преобразует радианы в градусы.

**ЧЁТН** — округляет число до ближайшего четного целого.

**EXP** — возвращает число  $e$ , возведенное в указанную степень.

**ФАКТР** — возвращает факториал числа.

**ДВФАКТР** — возвращает двойной факториал числа.

**ОКРВНИЗ** — округляет число до ближайшего меньшего по модулю целого.

**НОД** — возвращает наибольший общий делитель.

**ЦЕЛОЕ** — округляет число до ближайшего меньшего целого.

**НОК** — возвращает наименьшее общее кратное.

**LN** — возвращает натуральный логарифм числа.

**LOG** — возвращает логарифм числа по заданному основанию.

**LOG10** — возвращает десятичный логарифм числа.

- МОПРЕД** — возвращает определитель матрицы, хранящейся в массиве.
- МОБР** — возвращает обратную матрицу (матрица хранится в массиве).
- МУМНОЖ** — возвращает произведение матриц, хранящихся в массивах.
- ОСТАТ** — возвращает остаток от деления.
- ОКРУГЛТ** — возвращает число, округленное с требуемой точностью.
- МУЛЬТИНОМ** — возвращает мультиномиальный коэффициент множества чисел.
- НЕЧЁТ** — округляет число до ближайшего нечетного целого.
- ПИ** — возвращает число «пи».
- СТЕПЕНЬ** — возвращает результат возведения числа в степень.
- ПРОИЗВЕД** — возвращает произведение аргументов.
- ЧАСТНОЕ** — возвращает целую часть частного при делении.
- РАDIАНЫ** — преобразует градусы в радианы.
- СЛУЧИС** — возвращает случайное число в интервале от 0 до 1.
- СЛУЧМЕЖДУ** — возвращает случайное число в заданном интервале.
- РИМСКОЕ** — преобразует число в арабской записи к числу в римской как текст.
- ОКРУГЛ** — округляет число до указанного количества десятичных разрядов.
- ОКРУГЛВНИЗ** — округляет число до ближайшего меньшего по модулю целого.
- ОКРУГЛВВЕРХ** — округляет число до ближайшего по модулю большего целого.
- РЯД.СУММ** — возвращает сумму степенного ряда, вычисленную по заданной формуле.
- ЗНАК** — возвращает знак числа.
- SIN** — возвращает синус заданного угла.
- SINH** — возвращает гиперболический синус числа.
- КОРЕНЬ** — возвращает положительное значение квадратного корня.
- КОРЕНЬПИ** — возвращает значение квадратного корня из числа «пи».
- ПРОМЕЖУТОЧНЫЕ.ИТОГИ** — возвращает промежуточный итог в списке или базе данных.
- СУММ** — суммирует аргументы.

**СУММЕСЛИ** — суммирует ячейки, удовлетворяющие заданному условию.

**СУММПРОИЗВ** — возвращает сумму произведений соответствующих элементов массивов.

**СУММКВ** — возвращает сумму квадратов аргументов.

**СУММРАЗНКВ** — возвращает сумму разностей квадратов соответствующих значений в двух массивах.

**СУММСУММКВ** — возвращает сумму сумм квадратов соответствующих элементов двух массивов.

**СУММКВРАЗН** — возвращает сумму квадратов разностей соответствующих значений в двух массивах.

**TAN** — возвращает тангенс числа.

**TANH** — возвращает гиперболический тангенс числа.

**ОТБР** — отбрасывает дробную часть числа.

## Статистические функции

**СРОТКЛ** — возвращает среднее абсолютных значений отклонений точек данных от среднего.

**СРЗНАЧ** — возвращает среднее арифметическое аргументов.

**СРЗНАЧА** — возвращает среднее арифметическое аргументов, включая числа, текст и логические значения.

**БЕТАРАСП** — возвращает интегральную функцию плотности бета-вероятности.

**БЕТАОБР** — возвращает обратную функцию к интегральной функции плотности бета-вероятности.

**БИНОМРАСП** — возвращает отдельное значение биномиального распределения.

**ХИ2РАСП** — возвращает одностороннюю вероятность распределения хи-квадрат.

**ХИ2ОБР** — возвращает обратное значение односторонней вероятности распределения хи-квадрат.

**ХИ2ТЕСТ** — возвращает тест на независимость.

**ДОВЕРИТ** — возвращает доверительный интервал для среднего значения по генеральной совокупности.

**КОРРЕЛ** — возвращает коэффициент корреляции между двумя множествами данных.

**СЧЁТ** — подсчитывает количество чисел в списке аргументов.



- СЧЁТЗ** — подсчитывает количество значений в списке аргументов.
- СЧИТАТЬПУСТОТЫ** — подсчитывает количество пустых ячеек в заданном диапазоне.
- СЧЁТЕСЛИ** — подсчитывает количество непустых ячеек, удовлетворяющих заданному условию внутри диапазона.
- КОВАР** — возвращает ковариацию, т. е. среднее произведений отклонений для каждой пары точек.
- КРИТБИНОМ** — возвращает наименьшее значение, для которого биномиальная функция распределения меньше или равна заданному значению.
- КВАДРОТКЛ** — возвращает сумму квадратов отклонений.
- ЭКСПРАСП** — возвращает экспоненциальное распределение.
- ФРАСП** — возвращает  $F$ -распределение вероятности.
- ФРАСПОБР** — возвращает обратное значение для  $F$ -распределения вероятности.
- ФИШЕР** — возвращает преобразование Фишера.
- ФИШЕРОБР** — возвращает обратное преобразование Фишера.
- ПРЕДСКАЗ** — возвращает значение линейного тренда.
- ЧАСТОТА** — возвращает распределение частот в виде вертикального массива.
- ФТЕСТ** — возвращает результат  $F$ -теста.
- ГАММАРАСП** — возвращает гамма-распределение.
- ГАММАОБР** — возвращает обратное гамма-распределение.
- ГАММАНЛОГ** — возвращает натуральный логарифм гамма-функции.
- СРГЕОМ** — возвращает среднее геометрическое.
- РОСТ** — возвращает значения в соответствии с экспоненциальным трендом.
- СРГАРМ** — возвращает среднее гармоническое.
- ГИПЕРГЕОМЕТ** — возвращает гипергеометрическое распределение.
- ОТРЕЗОК** — возвращает отрезок, отсекаемый на оси линией линейной регрессии.
- ЭКСЦЕСС** — возвращает эксцесс множества данных.
- НАИБОЛЬШИЙ** — возвращает  $k$ -е наибольшее значение из множества данных.
- ЛИНЕЙН** — возвращает параметры линейного тренда.
- ЛГРФПРИБЛ** — возвращает параметры экспоненциального тренда.

- ЛОГНОРМОБР** — возвращает обратное логарифмическое нормальное распределение.
- ЛОГНОРМРАСП** — возвращает интегральное логарифмическое нормальное распределение.
- МАКС** — возвращает максимальное значение из списка аргументов.
- МАКСА** — возвращает максимальное значение из списка аргументов, включая числа, текст и логические значения.
- МЕДИАНА** — возвращает медиану заданных чисел.
- МИН** — возвращает минимальное значение из списка аргументов.
- МИНА** — возвращает минимальное значение из списка аргументов, включая числа, текст и логические значения.
- МОДА** — возвращает значение моды множества данных.
- ОТРБИНОМРАСП** — возвращает отрицательное биномиальное распределение.
- НОРМРАСП** — возвращает нормальную функцию распределения.
- НОРМОБР** — возвращает обратное нормальное распределение.
- НОРМСТРАСП** — возвращает стандартное нормальное интегральное распределение.
- НОРМСТОБР** — возвращает обратное значение стандартного нормального распределения.
- ПИРСОН** — возвращает коэффициент корреляции Пирсона.
- ПЕРСЕНТИЛЬ** — возвращает  $k$ -ю перцентиль для значений из интервала.
- ПРОЦЕНТРАНГ** — возвращает процентную норму значения в множестве данных.
- ПЕРЕСТ** — возвращает количество перестановок для заданного числа объектов.
- ПУАССОН** — возвращает распределение Пуассона.
- ВЕРОЯТНОСТЬ** — возвращает вероятность того, что значение из диапазона находится внутри заданных пределов.
- КВАРТИЛЬ** — возвращает квартиль множества данных.
- РАНГ** — возвращает ранг числа в списке чисел.
- КВПИРСОН** — возвращает квадрат коэффициента корреляции Пирсона.
- СКОС** — возвращает асимметрию распределения.
- НАКЛОН** — возвращает наклон линии линейной регрессии.

**НАИМЕНЬШИЙ** — возвращает  $k$ -е наименьшее значение в множестве данных.

**НОРМАЛИЗАЦИЯ** — возвращает нормализованное значение.

**СТАНДОТКЛОН** — оценивает стандартное отклонение по выборке.

**СТАНДОТКЛОНА** — оценивает стандартное отклонение по выборке, включая числа, текст и логические значения.

**СТАНДОТКЛОНП** — вычисляет стандартное отклонение по генеральной совокупности.

**СТАНДОТКЛОНПА** — вычисляет стандартное отклонение по генеральной совокупности, включая числа, текст и логические значения.

**СТОШУХ** — возвращает стандартную ошибку предсказанных значений  $y$  для каждого значения  $x$  в регрессии.

**СТЮДРАСП** — возвращает  $t$ -распределение Стьюдента.

**СТЮДРАСПОБР** — возвращает обратное  $t$ -распределение Стьюдента.

**ТЕНДЕНЦИЯ** — возвращает значения в соответствии с линейным трендом.

**УРЕЗСРЕДНЕЕ** — возвращает среднее внутренности множества данных.

**ТТЕСТ** — возвращает вероятность, соответствующую критерию Стьюдента.

**ДИСП** — оценивает дисперсию по выборке.

**ДИСПА** — оценивает дисперсию по выборке, включая числа, текст и логические значения.

**ДИСПР** — вычисляет дисперсию для генеральной совокупности.

**ДИСПРА** — вычисляет дисперсию для генеральной совокупности, включая числа, текст и логические значения.

**ВЕЙБУЛЛ** — возвращает распределение Вейбулла.

**ЗТЕСТ** — возвращает двустороннее  $P$ -значение  $z$ -теста.

## Текстовые функции

**ASC** — преобразует полноширинные (двухбайтные) английские буквы или знаки катакана в текстовой строке в полуширинные (один байт).

**БАТТЕКСТ** — преобразует число в текст, используя денежный формат Я (БАТ).

**СИМВОЛ** — возвращает знак с заданным кодом.

**ПЕЧСИМВ** — удаляет все непечатаемые знаки из текста.

**КОДСИМВ** — возвращает числовой код первого знака в текстовой строке.

**СЦЕПИТЬ** — объединяет несколько текстовых элементов в один.

**РУБЛЬ** — преобразует число в текст, используя денежный формат доллара.

**СОВПАД** — проверяет идентичность двух текстов.

**НАЙТИ** — ищет вхождение одного текста в другой (с учетом регистра).

**ФИКСИРОВАННЫЙ** — форматирует число и преобразует его в текст с заданным числом десятичных знаков.

**JIS** — преобразует полуширинные (однобайтные) английские буквы или знаки катакана в текстовой строке в полноширинные (два байта).

**ЛЕВСИМВ** — возвращает самые левые знаки текстового значения.

**ДЛСТР** — возвращает количество знаков в текстовой строке.

**СТРОЧН** — делает все буквы в тексте строчными.

**ПСТР** — возвращает определенное число знаков из строки текста, начиная с указанной позиции.

**PHONETIC** — извлекает фонетические (фуригана) знаки из текстовой строки.

**ПРОПНАЧ** — делает прописной первую букву в каждом слове текста.

**ЗАМЕНИТЬ** — заменяет знаки в тексте.

**ПОВТОР** — повторяет текст заданное число раз.

**СЖПРОБЕЛЫ** — удаляет из текста пробелы.

**ПРОПИСН** — делает все буквы в тексте прописными.

**ЗНАЧЕН** — преобразует текстовый аргумент в число.

**ПРАВСИМВ** — возвращает самые правые знаки текстовой строки.

**ПОИСК** — ищет вхождение одного текста в другой (без учета регистра).

**ПОДСТАВИТЬ** — заменяет в текстовой строке старый текст новым.

**T** — преобразует аргумент в текст.

**ТЕКСТ** — форматирует число и преобразует его в текст.

## Примеры выражений, используемых в запросах

Таблица П2.1. Примеры выражений, используемых в вычисляемых полях

Выражение	Результаты отбора
<b>Обработка текстовых значений</b>	
ПолноеИмя: [Имя] & " " & [Фамилия]	Отображение в поле «Полное имя» значения полей «Имя» и «Фамилия», разделенные пробелом
Адрес2: [Город] & " " & [Район] & " " & [Индекс]	Отображение в поле «Адрес2» значения полей «Город», «Область» и «Индекс», разделенные пробелами
ПерваяБукваТовара: Left([НазваниеТовара], 1)	Отображение в поле «ПерваяБукваТовара» первого знака значения поля «НазваниеТовара»
КодТипа: Right([КодПродукта], 2)	Отображение в поле «КодТипа» последних двух знаков значения поля «КодПродукта»
КодРайона: Mid([Телефон], 2, 3)	Отображение в поле «КодРайона» трех знаков, начиная со второго, значения поля «Телефон»
<b>Обработка и вычисление дат</b>	
ВремяДоставки: DateDiff("d", [ДатаРазмещения], [ДатаИсполнения])	Отображение в поле «ВремяДоставки» числа дней между датой размещения и датой исполнения
ГодНайма: DatePart("yyyy", [ДатаНайма])	Отображение в поле «ГодНайма» года найма сотрудника
ЧислоМесяцев: DatePart("M", [ДатаИсполнения])	Отображение в поле «ЧислоМесяцев» числа месяцев
ПредшествующаяДата: Date( ) - 30	Отображение в поле «ПредшествующаяДата» даты, предшествующей текущей на 30 дней
<b>Выполнение арифметических операций</b>	
ПервоначальныйФрахт: [Фрахт] * 1.1	Отображение в поле «ПервоначальныйФрахт» стоимости фрахта плюс 10 процентов

Продолжение табл. П2.1

Выражение	Результаты отбора
ОбъемЗаказа: [Количество] * [Цена]	Отображение в поле «ОбъемЗаказа» произведения значений полей «Количество» и «Цена»
ВремяДоставки: [ДатаРазмещения] — [ДатаИсполнения]	Отображение в поле «ВремяДоставки» разницы между значениями полей «ДатаРазмещения» и «ДатаИсполнения»
ВсегоВНаличии: [НаСкладе] + [Ожидается]	Отображение в поле «ВсегоВНаличии» суммы значений полей «НаСкладе» и «Ожидается»
ДоляФрахта: Sum([Фрахт])/Sum([Общий Итог]) * 100	<p>Отображение в поле «ДоляФрахта» процента на оплату фрахта для каждого итога. Для этого сумма всех значений поля «Фрахт» делится на сумму всех значений поля «ОбщийИтог». На экране должна быть отображена строка Групповая операция в бланке запроса (Бланк запроса. Бланк, предназначенный для определения запроса или фильтра в режиме конструктора запроса или в окне расширенного фильтра. В предыдущих версиях использовался термин бланк запроса по образцу (QBE).), а в ячейке Групповая операция для данного поля должно быть установлено значение Выражение.</p> <p>Если для данного поля в ячейке свойства Формат поля (Format) установлено значение Процентный, не включайте выражение *100</p>
КоличествоТоваров: Nz([НаСкладе], 0) + Nz([Ожидается], 0)	<p>Отображение в поле «КоличествоТоваров» общего количества товаров как заказанных, так и находящихся на складе.</p> <p>Если в выражении используются арифметические операторы (+, -, *, /), а одно из полей имеет значение Null (Null. Значение, которое можно ввести в поле или использовать в выражениях и запросах для указания отсутствующих или неизвестных данных. В Visual Basic ключевое слово Null указывает значение Null. Некоторые поля, такие, как поля первичного ключа, не могут содержать значения Null.), результатом всего выражения тоже будет значение Null. Если какие-либо записи в одном из использованных в выражении полей могут содержать значение Null, то это значение Null можно преобразовать в ноль с помощью функции Nz</p>

Выражение	Результаты отбора
<b>Использование статистических функций SQL или статистических функций по подмножествам</b>	
Count(*)	Подсчет с помощью функции Count числа записей в запросе, включая записи, содержащие поля со значением <u>Null</u> (Null. Значение, которое можно ввести в поле или использовать в выражениях и запросах для указания отсутствующих или неизвестных данных. В Visual Basic ключевое слово Null указывает значение Null. Некоторые поля, такие, как поля первичного ключа, не могут содержать значения Null.) (пустые поля)
ДоляФрахта: Sum([Фрахт])/Sum([ОбщийИтог]) * 100	Отображение в поле «ДоляФрахта» процента на оплату фрахта для каждого итога. Для этого сумма всех значений поля «Фрахт» делится на сумму всех значений поля «ОбщийИтог». На экране должна быть отображена строка Групповая операция в <u>бланке запроса</u> (Бланк запроса. Бланк, предназначенный для определения запроса или фильтра в режиме конструктора запроса или в окне расширенного фильтра. В предыдущих версиях использовался термин бланк запроса по образцу (QBE).), а в ячейке Групповая операция для данного поля должно быть установлено значение Выражение. Если для данного поля в ячейке свойства Формат поля (Format) установлено значение Процентный, не включайте выражение *100
СредняяСкидка: DAvg("Скидки", "Заказы")	Отображение в поле «СредняяСкидка» значения средней скидки, предоставляемой по всем заказам, объединенным в итоговый запрос
<b>Работа со значениями Null</b>	
ТекущаяСтрана: If(IsNull([Страна]))	(пустое поле)
ВремяДоставки: If(IsNull([ДатаРазмещения]) — [ДатаИсполнения])	Проверьте отсутствующую дату
ПродажиЗаПолгода: Nz([Продажи1кв],0) + Nz([Продажи2кв],0)	Отображение в поле «ПродажиЗаПолгода» суммы значений полей по продажам за первый и второй кварталы

Таблица П2.2. Примеры выражений, используемых в запросах на обновление записей

Выражение	Результаты выполнения
"Продавец"	Изменяет текстовое значение на «Продавец»
#10.08.99#	Изменяет значение даты на 10-авг-99
Да	Изменяет в логическом поле значения «Нет» на «Да»
"PN" & [ИнвНомер]	Добавляет строку «PN» к началу каждого инвентарного номера
[Цена] * [Количество]	Вычисляет произведение значений полей «Цена» и «Количество»
[СтоимостьДоставки] * 1.5	Увеличивает стоимость доставки на 50 %
DSum("[Количество] * [Цена]", "Заказано", "[КодТовара]=" & [КодТовара])	Если значение поля «КодТовара» в текущей таблице совпадает со значением поля «КодТовара» в таблице «Заказано», обновляет итоговые значения продаж, рассчитываемые как произведение полей «Цена» и «Количество»
Right([Индекс], 5)	Обрезает левые знаки в поле, оставляя пять правых знаков
IIf(IsNull([Цена]), 0, [Цена])	Преобразует значения <u>Null</u> (Null. Значение, которое можно ввести в поле или использовать в выражениях и запросах для указания отсутствующих или неизвестных данных. В Visual Basic ключевое слово Null указывает значение Null. Некоторые поля, такие, как поля первичного ключа, не могут содержать значения Null.) в поле «Цена» в нули (0)



## Краткий словарь терминов Интернет

**IP-адрес** — 32-х битное двоичное число, присваиваемое каждому компьютеру для того, чтобы отличить его от всех остальных в Интернете.

**URL [Ю Эр Эль]** — Universal Resource Locator, URL — это адрес любого ресурса в Интернете вместе с указанием того, с помощью какого протокола следует к нему обращаться.

**World Wide Web (WWW)** — это распределенная информационная система с гиперсвязями, существующая на технической базе всемирной компьютерной сети Интернет.

**Абонентская система** — совокупность абонента и станции.

**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.

**Станция** — аппаратура, которая выполняет функции, связанные с передачей и приемом информации.

**Архитектура вычислительной сети** — это описание ее общей модели.

**Гиперссылка** — это некоторое ключевое слово или объект в документе, с которым связан указатель для перехода на другую страницу, в другой документ.

**Гипертекст** — множество отдельных документов, связанных гиперссылками.

**Домен** — это произвольная группа серверов, использующих единую базу учетных записей пользователей и политику защиты.

**Доменное имя (domain name)** — уникальный идентификатор, который назначается определенному IP-адресу.

**Интернет** — всемирная информационная компьютерная сеть, которая объединяет в единое целое множество компьютерных сетей, работающих по единым правилам.

- Интернет-телефония** — голосовое общение через Сеть в режиме on-line.
- Клиент** — задача, рабочая станция или пользователь компьютерной сети.
- Коммуникационная среда** — линии связи или пространство, в котором распространяются электрические сигналы, и аппаратура передачи данных. Эта среда обеспечивает взаимодействия абонентов.
- Маршрутизатор, или роутер** — устройство, соединяющее сети разного типа, но использующее одну операционную систему.
- Мост** — устройство, соединяющее сети разных топологий, но работающие под управлением однотипных сетевых операционных систем.
- Опорная сеть Интернет** — совокупность компьютерных узлов, соединенных линиями связи (оптоволоконными или спутниковыми) с высокой пропускной способностью и помехозащищенностью.
- Открытая система** — система, взаимодействующая с другими системами в соответствии с принятыми стандартами.
- Поисковые серверы** — выделенные хост-компьютеры, в которых размещаются базы данных ресурсов Интернет. По способу наполнения поисковые серверы делятся на каталоги и поисковые машины.
- Каталоги** — иерархические базы данных о документах сети, пополнение которых осуществляется человеком.
- Поисковые машины** — иерархические базы данных о документах сети, пополнение которых осуществляется специальными программами. Поисковые машины устроены гораздо сложнее каталогов. В структуре поисковой машины выделяют обычно три составляющие:
- специальную программу, осуществляющую сканирование Сети, именуемую *поисковым роботом*;
  - базу данных документов, называемую *индексом*;
  - интерфейсную часть, предназначенную для взаимодействия с пользователем.

- Провайдер** (поставщик услуг Интернет) — организация, обеспечивающая подключение пользователей к Интернет.
- Сервер** — компьютер, подключенный к сети и обеспечивающий ее пользователей определенными услугами.
- Сетевой протокол** — совокупность правил, в соответствии с которыми осуществляется взаимодействие устройств, и последовательность выполнения действий, необходимых для обмена сообщениями.
- Сигнал** — переносчик сообщения, представляет собой физический процесс или явление, обеспечивающее перенос сообщения в коммуникационной среде. Сообщение преобразуется в сигнал станцией.
- Служба Telnet** — сервис, обеспечивающий взаимодействие с удаленным компьютером. Он позволяет превратить компьютер пользователя в удаленный терминал другого компьютера.
- Сообщение** — цифровые данные, предназначенные для распространения (передачи или приема) в коммуникационной среде.
- Телеконференция** — это система обмена электронными сообщениями на определенную тему между абонентами сети (в режиме отложенного общения — offline).
- Узел опорной сети** — как правило, несколько мощных компьютеров, находящихся в состоянии постоянного подключения.
- Форумы прямого общения** — IRC (Internet Relay Chat) — это система обмена электронными сообщениями в режиме реального времени — on-line.
- Шлюзы** — устройства, позволяющие организовать обмен данными между двумя сетями, использующими различные протоколы взаимодействия.
- Электронная почта** — это система обмена электронными сообщениями в компьютерных сетях в режиме отложенного общения — offline.

# Указатель

---

---

## D

---

DNS-сервер 367

## U

---

URL 380

## W

---

Web-сайт 378

Web-сервер 378

Web-страница 378

## A

---

Абзац 119

Абонент сети 26, 320

Абонентская система 26, 320

Абсолютные ссылки 173

Автозамена 129

Администратор базы данных 261

Алгоритм 41, 42

Анализ 198

Аннуитет 194

Аргументы функции 175

Архитектура ЭВМ 70

## Б

---

База данных 41, 253

Базовая система ввода-вывода 99

Базовое программное  
обеспечение 91

Безопасность информации 392

Браузер 378

Буфер памяти 107

## В

---

Вложенный документ 147

Выдача 266

Высказывание 31

Вычисляемое поле 309

## Г

---

Гарнитура шрифта 120

Гиперссылка 378

Гипертекст 378

Гистограмма 184

Главный документ 147

Глобальные сети 336

Групповой запрос 305

## Д

---

Данные 13

Детерминированность алгоритма 42

Диаграмма 199

Дискретность алгоритма 42

Домен 362

Доменное имя 363

Доступность информации 392

## З

---

Заголовок 141

Закладка 143

Запись 32, 281

Запрос 298

— на выборку 301

— на добавление 312

— на изменение 312

— на обновление записей 312

— на создание таблицы 314

— на удаление 313

Зараженная программа 403

Зараженный диск 403

## И

---

Интерпретатор 53

Интерфейс 325

Инфологическая модель 282  
 Информатизация 10  
 Информатика 4  
 Информационная система 260  
 Информационные процессы 14  
 — ресурсы 8  
 Информационный продукт 10  
 Информация 4  
 Исправление 140  
 Источник данных 149  
 Исходный модуль программы 53

**К**

Категория данных 208  
 Кластер 102  
 Клиент 343  
 Ключевое поле 281  
 Ключевые слова 266  
 Код 13  
 Кодирование 13  
 Колонтитул 120  
 Команда алгоритма 42  
 Коммуникационная среда 26, 320  
 Компилятор 53  
 Компьютерная сеть 319  
 Компьютерный вирус 402  
 Конечность алгоритма 43  
 Константа 29  
 Контроллер 71  
 Конфигурация 86  
 Конфиденциальность  
 информации 392  
 Корневой каталог 104  
 Кредит 193  
 Курсор 163

**Л**

Легенда 208  
 Локальные сети 335

**М**

Маркеры данных 207  
 Массив 31  
 Массовость алгоритма 43  
 Мастер функций 177

Модель 256  
 Мост 353

**Н**

Надписи 138  
 Надстройка 176  
 Названия 138  
 Непечатаемые символы 117  
 Нормализация отношений 278  
 Носитель информации 19

**О**

Обеспечение информационной  
 безопасности 406  
 Область отображения задач 115  
 Область уведомлений 115  
 Обработка информации 13, 23  
 Одноранговая сеть 344  
 Окно 115  
 Операционная оболочка 92  
 — система 91, 94  
 Опорная сеть Интернет 357  
 Организационные (административ-  
 ные) меры защиты 407  
 Основной документ 148  
 Ось диаграммы 208  
 Относительные ссылки 173  
 Отношение 40, 258, 281  
 Отчет 292, 296

**П**

Панель задач 113  
 Передача информации 25  
 Перекрестные ссылки 138  
 Перекрестный запрос 307  
 Переменная 29  
 Пиксель 158  
 Пиктограмма 113  
 Подбор параметра 232  
 Поле 281  
 Поле слияния 149  
 Поля 133  
 Понятность алгоритма 43  
 Правовые меры защиты  
 информации 406

Предложение 119  
Предметная область 253  
Примечание 141  
Принципы фон Неймана 69  
Провайдер 358  
Программа 89  
Программное обеспечение 89  
Программный продукт 89  
Протокол 325  
Процессор 70

---

**Р**

---

Рабочая станция 343  
Рабочий стол 108  
Раздел документа 134  
Ранг 186  
Распределенная обработка  
данных 342  
Расширенный фильтр 295  
Редактирование 129  
Результативность алгоритма 42  
Релевантность 267  
Ряд данных 207

---

**С**

---

Сбор информации 14  
Сводная таблица 227  
Сервер 343  
Сервисное программное  
обеспечение 92  
Сетевая операционная система 92  
Сеть с выделенным сервером 345  
Сигнал 26  
Символ 117  
Синтаксис формулы 169  
Синтаксис функции 175  
Система команд исполнителя 42  
Системы управления базами  
данных 260  
Слово 118  
Сноска 139  
Сообщение 26  
Список 136, 212  
Справочно-правовые системы 268  
Ссылка 174  
Станция 26, 320

Стиль 135  
Страница 119  
Строка 118  
Структура данных 36  
Сущность 280

---

**Т**

---

Таблица 151  
— размещения файлов 102  
Таблицы подстановки 238  
Табличный процессор 153  
Тип данного 30  
Топологии 337  
Транслятор 52

---

**У**

---

Узел 36  
Утилиты 92

---

**Ф**

---

Файл 22, 32, 104  
— подкачки 108  
Файловая система 101  
Физические меры защиты 407  
Фильтр по выделенному 294  
Фильтрация 216  
Форма 289  
— данных 213  
Форматирование 129  
Формула 169  
Функции 175

---

**Х**

---

Хост-компьютер 357

---

**Ц**

---

Целостность информации 392

---

**Э**

---

Электронная таблица 153

---

**Я**

---

Ярлык 111, 113

# Оглавление

---

---

<b>Введение</b> . . . . .	3
<b>Глава 1. ИНФОРМАЦИЯ И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ</b> . . . . .	4
1.1. Основные понятия информатики . . . . .	4
1.2. Краткая характеристика информационных процессов . . . . .	11
<b>Глава 2. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ОБРАБОТКИ ДАННЫХ</b> . . . . .	28
2.1. Данные и действия с ними . . . . .	28
2.2. Алгоритмы и программы обработки данных . . . . .	41
<b>Глава 3. ОСНОВЫ УСТРОЙСТВА ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА</b> . . . . .	64
3.1. Принципы устройства компьютера . . . . .	64
3.2. Аппаратная реализация персонального компьютера . . . . .	72
<b>Глава 4. ПРОГРАММНЫЕ СРЕДСТВА ИНФОРМАТИЗАЦИИ</b> . . . . .	88
4.1. Классификация программного обеспечения . . . . .	89
4.2. Общие сведения об операционных системах . . . . .	94
4.3. Особенности операционной системы MS Windows . . . . .	106
<b>Глава 5. ОСНОВЫ ТЕХНОЛОГИИ ОБРАБОТКИ ТЕКСТОВ</b> . . . . .	116
5.1. Документ и его состав . . . . .	117
5.2. Процедуры обработки текста . . . . .	128
5.3. Дополнительные объекты документа . . . . .	138
5.4. Особенности работы с документами большого объема . . . . .	141
5.5. Особенности создания текстовых документов . . . . .	144
<b>Глава 6. ОСНОВЫ ФУНКЦИОНИРОВАНИЯ ТАБЛИЧНОГО ПРОЦЕССОРА MS EXCEL</b> . . . . .	151
6.1. Табличные структуры данных . . . . .	151
6.2. Принципы функционирования электронных таблиц ТП MS EXCEL . . . . .	156

6.3. Понятие функции в электронных таблицах . . . . .	175
6.4. Принципы вычислений с помощью функций . . . . .	179
6.5. Анализ данных в ЭТ . . . . .	198
6.6. Списки и их использование для анализа данных . . . . .	212
6.7. Решение задач путем подбора параметра . . . . .	231
6.8. Использование таблиц подстановки для анализа данных . . . . .	238
6.9. Решение оптимизационных задач . . . . .	241
<b>Глава 7. ОСНОВЫ ФУНКЦИОНИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ . . . . .</b>	<b>253</b>
7.1. Базы данных и системы управления ими . . . . .	253
7.2. Состав и классификация информационных систем . . . . .	260
7.3. Справочно-правовые системы . . . . .	268
<b>Глава 8. РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ И СИСТЕМЫ УПРАВЛЕНИЯ ИМИ . . . . .</b>	<b>277</b>
8.1. Основы построения реляционных баз данных . . . . .	277
8.2. Состав и возможности СУБД MS ACCESS . . . . .	284
8.3. Анализ данных в MS ACCESS . . . . .	294
<b>Глава 9. ОСНОВЫ ПОСТРОЕНИЯ КОМПЬЮТЕРНЫХ СЕТЕЙ . . . . .</b>	<b>317</b>
9.1. Принципы функционирования компьютерных сетей . . . . .	318
9.2. Локальные вычислительные сети . . . . .	336
<b>Глава 10. ГЛОБАЛЬНАЯ КОМПЬЮТЕРНАЯ СЕТЬ ИНТЕРНЕТ . . . . .</b>	<b>355</b>
10.1. Основы функционирования Интернет . . . . .	355
10.2. Основные службы Интернет . . . . .	369
<b>Глава 11. ОСНОВЫ ЗАЩИТЫ ИНФОРМАЦИИ . . . . .</b>	<b>389</b>
11.1. Угрозы безопасности информационных систем . . . . .	389
11.2. Противодействие угрозам безопасности информационных систем . . . . .	406
<b>Литература . . . . .</b>	<b>420</b>
<b>Приложение 1. Функции табличного процессора MS Excel . . . . .</b>	<b>421</b>
<b>Приложение 2. Примеры выражений, используемых в запросах . . . . .</b>	<b>436</b>
<b>Приложение 3. Краткий словарь терминов Интернет . . . . .</b>	<b>440</b>
<b>Указатель . . . . .</b>	<b>443</b>



**Агальцов Виктор Петрович  
Титов Виктор Михайлович**  
**Информатика для экономистов**

Учебник

Редактор *А. В. Волковицкая*  
Корректор *А. В. Алешина*  
Компьютерная верстка *И. В. Кондратьевой*  
Оформление серии *Т. В. Иваншиной*

ЛР № 071629 от 20.04.98  
Издательский Дом «ФОРУМ»  
101990, Москва — Центр, Колпачный пер., д. 9а  
Тел./факс: (495) 625-39-27  
E-mail: forum-books@mail.ru

ЛР № 070824 от 21.01.93  
Издательский Дом «ИНФРА-М»  
127282, Москва, Полярная ул., д. 31в  
Тел.: (495) 380-05-40  
Факс: (495) 363-92-12  
E-mail: books@infra-m.ru  
Http://www.infra-m.ru

***По вопросам приобретения книг обращайтесь:***

*Отдел продаж «ИНФРА-М»*  
127282, Москва, ул. Полярная, д. 31в  
Тел.: (495) 363-42-60  
Факс: (495) 363-92-12  
E-mail: books@infra-m.ru

*Центр комплектования библиотек*  
119019, Москва, ул. Моховая, д. 16  
(Российская государственная библиотека, кор. К)  
Тел.: (495) 695-93-15