

# Методические указания (рекомендации) для студентов по изучению дисциплины «Информатика и программирование»

## 1. Общие положения

Целями освоения дисциплины «Информатика и программирование» являются:

- формирование основ научного мировоззрения, представлений об информатике как о фундаментальной науке и универсальном языке естественнонаучных дисциплин.
- формирование способности у студентов работы с информацией, умений и навыков применения методов информатики для решения профессиональных задач с использованием персонального компьютера.
- воспитание у студентов информационной культуры, отчетливого представления о роли этой науки в современном обществе;
- формирование у студентов системы знаний в области теории и практики применения информационных технологий в сфере управления профессиональной деятельностью.
- дисциплина нацелена на развитие у студентов навыков практической работы, которые востребуются напрямую и немедленно, сразу после включения молодого специалиста в профессиональную деятельность.

Более подробно цели курса представлены в рабочей программе (РП) дисциплины.

В курсе важное место занимают вопросы изучения модели инфокоммуникаций, структуры компьютерных сетей, базовые принципы и методы их построения, а также состояние, основные направления и тенденции их развития. Кроме этого, студенты должны приобрести навыки анализа преимуществ и недостатков локальных и глобальных сетей.

Организация обучения дисциплины включает в себя следующие основные компоненты:

- лекции,
- лабораторные работы,
- самостоятельная работа,
- зачет или экзамен.

### **Рекомендации:**

Ознакомиться с содержанием курса по рабочей программе дисциплины.

Выписать из рабочей программы:

- список рекомендованной литературы;
- наименования лекционных разделов курса;
- названия лабораторных работ;
- теоретические вопросы к зачету.

## 2. Лекционные занятия (теоретический курс)

Лекция является ведущей формой учебных занятий. На лекциях активная роль принадлежит преподавателю, задача которого сводится к тому, чтобы в отведенное время раскрыть содержание учебных вопросов и дать схему ответа на узловые проблемы темы.

### Рекомендации:

Перед очередной лекцией необходимо просмотреть по конспекту материал предыдущей лекции. При затруднениях в восприятии материала надо обратиться к основным и дополнительным литературным источникам (**раздел 6 РП**). Если разобраться в материале не удалось, следует обратиться к лектору по графику его консультаций или на лабораторных занятиях.

## 3. Лабораторные работы

После лекций по каждой теме дисциплины предлагается ряд лабораторных работ. На лабораторных занятиях задания выполняются на компьютерах.

Лабораторные работы должны выполняться самостоятельно. В противном случае студент не приобретет необходимых знаний и может оказаться неподготовленным к защите работы, а, в конечном итоге, к зачету и экзамену.

### Структура лабораторной работы.

Начинается лабораторная работа с титульного листа. Титульный лист лабораторной работы должен содержать:

- наименование дисциплины;
- название лабораторной работы;
- номер группы;
- фамилию, имя, отчество студента;
- фамилию, имя отчество преподавателя.

Отчет о лабораторной работе должен содержать следующие основные части:

- а) постановка задачи,
- б) результаты выполнения работы,
- в) выводы по работе.

Отчет о лабораторной работе выполняется на листах формата А4 в текстовом процессоре Microsoft Word. Работа выполняется на листах с одной стороны. Лабораторная работа должна быть выполнена в срок, определенный графиком изучения дисциплины. По согласованию с преподавателем отчет может быть представлен в электронном виде.

Защита лабораторных работ принимается преподавателем, проводившим лабораторные занятия в группе или читающим лекции по данному курсу.

Отметка «зачтено» выставляется в случае успешного выполнения задания, правильного оформления, положительного ответа на все контрольные вопросы, задаваемые преподавателем.

#### **Рекомендации:**

- руководствоваться графиком лабораторных работ РП;
- накануне перед очередной работой необходимо по конспекту или в методических указаниях к лабораторной работе просмотреть теоретический материал;
- на лабораторном занятии необходимо проанализировать окончательные результаты и убедиться в их достоверности;
- обратить внимание на оформление отчета, в котором должны присутствовать: цель работы, результаты работы и выводы;
- при подготовке отчета руководствоваться примерами, приведенными в методических указаниях к данной лабораторной работе.

### **4. Образовательные технологии**

Реализация программы предусматривает использование образовательных технологий, направленных на формирование элементов компетенций, в обеспечении которых участвует дисциплина «Информатика и информационно-коммуникационные технологии». В процессе обучения используются следующие образовательные технологии:

1. Информационно-коммуникативные технологии, позволяющие овладевать и свободно оперировать большим запасом знаний путем самостоятельного изучения профессиональной литературы, применения новых информационных технологий, включая использование технических и электронных средств получения информации.
2. Проблемно-ориентированные технологии, направленные на формирование и развитие проблемного мышления, мыслительной активности, способности видеть и формулировать проблемы, выбирать средства для их решения.
3. Практико-ориентированные технологии, направленные на формирование системы профессиональных практических умений и навыков, позволяющих качественно осуществлять профессиональную деятельность.
4. Личностно-ориентированные технологии, обеспечивающие в ходе учебного процесса учет различных способностей обучаемых, создание необходимых условий для развития их индивидуальных способностей, развитие активности личности учебном процессе.

### **5. Самостоятельная работа студентов**

Самостоятельная работа студентов осуществляется по разделам и темам курса, по которым не предусмотрены аудиторские занятия, либо требуется

дополнительно проработать и проанализировать рассматриваемый преподавателем материал в объеме запланированных часов. Задания по самостоятельной работе включают в себя:

- конспектирование первоисточников и другой учебной литературы;
- проработку учебного материала (по конспектам лекций, учебной и научной литературе);
- поиск и обзор научных публикаций и электронных источников информации;
- выполнение лабораторных работ;
- работа с тестами и вопросами для самопроверки.

Для самостоятельного изучения теоретического материала рекомендуются печатные и электронные ресурсы, включенные в список литературы РП.

### **Рекомендации:**

- руководствоваться графиком самостоятельной работы РП;
- выполнять все плановые задания, выдаваемые преподавателем для самостоятельного выполнения, и разбирать на консультациях неясные вопросы;
- особое внимание следует обращать на определение основных понятий дисциплины;
- после изучения определенной темы и выполнения лабораторной работы студенту рекомендуется воспроизвести по памяти определения, проверяя себя всякий раз по учебнику или конспекту лекций;
- изучая материал по учебнику или конспекту лекций, следует переходить к следующему вопросу в том случае, когда хорошо усвоен предыдущий вопрос;
- подготовку к зачету необходимо проводить по перечню теоретических вопросов, представленным в файле «Оценочные материалы.doc» УМК дисциплины;
- при подготовке к зачету параллельно прорабатываете соответствующие теоретические и практические разделы курса, все неясные моменты фиксируйте и выносите на плановую консультацию.

## **6. Текущий контроль**

Контроль успеваемости и качества подготовки студентов включает текущий контроль успеваемости, рубежный контроль и промежуточную аттестацию.

Текущий контроль успеваемости проводится в ходе всех видов учебных занятий в форме устного опроса при защите лабораторных работ.

Промежуточная аттестация имеет целью определить степень достижения учебных целей по дисциплине и проводится в форме защиты лабораторных работ и ответов на теоретические вопросы.

Рубежный контроль имеет целью установить качество усвоения учебного материала по определенным темам учебной дисциплины. Проводится в форме устного опроса по темам и написания контрольных работ.

Результаты, полученные при выполнении студентами в течение семестра заданий, оцениваются по шкале:

**1 рейтинг:** максимальный балл-20.

нижний порог -10; верхний порог-20.

**2 рейтинг:** максимальный балл -30

нижний порог -15; верхний порог-30.

## 7. Экзамен

Завершающим этапом изучения курса является экзамен. Допуск к экзамену студент получает после того, как выполнены и защищены лабораторные работы. Список теоретических вопросов, выносимых на экзамен, представлен в файле «Оценочные материалы.doc» УМК дисциплины. Если какие-то вопросы вынесены преподавателем на самостоятельное изучение, следует обратиться к учебной литературе, рекомендованной преподавателем в качестве источника сведений.

Для успешной сдачи экзамена студент должен продемонстрировать знание основных теоретических положений дисциплины и показать свои навыки применения теории при решении конкретных практических задач.

Экзамен оценивается от 0 – до 50 баллов.

По суммарным результатам экзамена и рейтингов выставляется оценка.

В аудитории при подготовке к экзамену нельзя пользоваться какими-либо материалами. В случае затруднений следует обратиться к преподавателю, принимающему экзамен. Студенту, замеченному в списывании, экзамен не засчитывается.

# Конспект лекций к изучению дисциплины «Информатика и информационно-коммуникационные технологии»

## Понятие информации. Общая характеристика процессов сбора, передачи, обработки и накопления информации

### *Лекция 1.1. Основные задачи информатики*

#### **1.1.1. Место информатики в современном научном знании**

В жизни современного человека слово «информация» очень распространено и часто используется в разных контекстах. Информацией обычно называют любые сведения, которые кого-либо интересуют, которые мы получаем или передаем. Например, мы говорим: «В результате научных исследований нами получена информация о ...» или «Из книг мы информированы о ..., «По телевидению была передана информация о ...».

Фундаментальной чертой цивилизации является рост производства и потребления информации во всех отраслях человеческой деятельности. Вся жизнь человека, так или иначе, связана с получением, накоплением, обработкой, передачей информации. Практически в любой деятельности человека: в процессе чтения книг, просмотра телепередач, в разговоре мы получаем и обрабатываем информацию.

Начиная с XVII в. объем научной информации удваивался, примерно, каждые 20 лет, в настоящее время он удваивается в 5 – 6 лет и тенденция ускорения сохраняется. Одной из важнейших проблем человечества является лавинообразный рост потока информации в любой отрасли жизнедеятельности. Подсчитано, что современный специалист должен тратить около 80% своего рабочего времени, чтобы уследить за всеми новыми работами в его области деятельности.

Увеличение объема используемой человеком информации и растущий спрос на нее обусловили появление отрасли знаний, связанной с автоматизацией обработки информации – информатики.

#### **1.1.2. Предмет и структура информатики**

Термин информатики получил распространение с середины 80-х г.г. прошлого века. Inform – «информация» и matic – «наука о ...». Т.о., информатика – это наука об информации. В англоязычных странах используется термин Computer Science – наука о компьютерах.

Информатика – это наука, изучающая:

- Методы реализации информационных процессов средствами вычислительной техники (СВТ);
- Состав, структуру, общие принципы функционирования;
- Принципы управления СВТ.

Из определения следует, что информатика – прикладная наука, использующая научные достижения многих наук.

Кроме того, информатика – практическая наука, которая не только занимается описательным изучением перечисленных вопросов, но и во многих случаях предлагает способы их решения. В этом смысле информатика технологична и часто смыкается с информационными технологиями.

Методы реализации информационных процессов находятся на стыке информатики с теорией информатики, статистикой, теорией кодирования, математической логикой, документоведением и т.д.

В этом разделе изучаются вопросы:

- Представление различных типов данных (числа, символы, текст, звук, графика, видео и т.д.) в виде, удобном для обработки СВТ.(кодирование данных);
- Форматы представления данных (одни и те же данные могут быть представлены разными способами);
- Структуры данных, т.е. способы хранения, с целью удобного доступа к данным.

В изучении состава, структуры, принципов функционирования СВТ используются научные положения из электроники, автоматики, кибернетики. В целом этот раздел известен как аппаратное обеспечение (АО) информационных процессов.

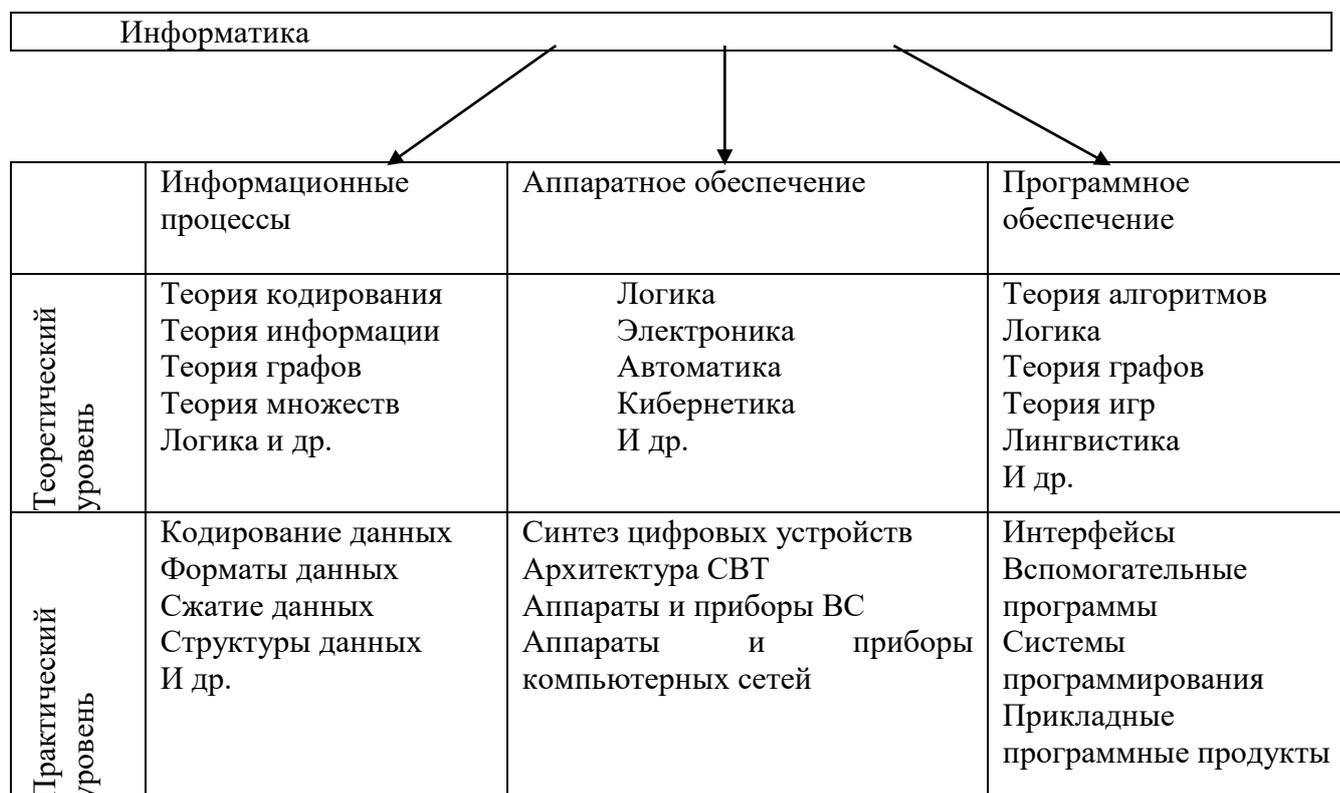
В этом разделе изучаются:

- Основы построения элементов цифровых устройств;
- Основные принципы функционирования цифровых вычислительных устройств;
- Архитектура СВТ – основные принципы функционирования систем, предназначенных для автоматической обработки данных;
- Приборы и аппараты, составляющие аппаратную конфигурацию вычислительных систем;
- Приборы и аппараты, составляющие аппаратную конфигурацию компьютерных сетей.

В разработке методов управления СВТ используют научные положения из теории алгоритмов, логики, теории графов, лингвистики, теории игр. Этот раздел информатики известен как программное обеспечение СВТ. В этом разделе изучается:

- Средства взаимодействия аппаратного и программного обеспечения;
- Средства взаимодействия человека с аппаратным и программным обеспечением, объединяемые понятием интерфейс;
- Программное обеспечение СВТ (ПО).

Обобщая сказанное, можно предложить следующую структурную схему:



Третья составляющая информатики – ПО неоднородна и имеет сложную структуру, включающую несколько уровней: базовый, системный, служебный, инструментальный, прикладной.

Самый низкий уровень ПО представляет базовое программное обеспечение. Оно отвечает за взаимодействие с базовыми аппаратными средствами. Как правило, базовые программные средства непосредственно входят в состав базового оборудования и хранятся в специальных микросхемах, называемых постоянными запоминающими устройствами (ПЗУ – Read Only Memory, ROM). Программы и данные записываются («прошиваются») в микросхемы ПЗУ на этапе производства и не могут быть изменены в процессе эксплуатации.

В последнее время вместо микросхем ПЗУ применяют перепрограммируемые запоминающие устройства (ППЗУ), позволяющие изменять базовые программные средства во время эксплуатации. Изменение содержания ППЗУ можно выполнять как непосредственно в составе вычислительной системы (такая технология называется *флэш-технологией*), так и вне ее, на специальных устройствах, называемых *программаторами*.

Комплект программ, находящихся в ПЗУ образуют систему ввода/вывода (BIOS). Основное назначение программ этого пакета – проверка состава и работоспособности компьютерной системы и обеспечение взаимодействия с клавиатурой, монитором, жестким диском и дисководом гибких дисков и CD-ROM.

Программы, входящие в BIOS позволяют наблюдать на экране диагностические сообщения, сопровождающие запуск компьютера, а также вмешиваться в ход запуска с помощью клавиатуры.

На следующем уровне находятся комплексы программ, осуществляющих интерфейсные функции (посреднические между человеком и компьютером, аппаратным и программным обеспечением, между одновременно работающими программами), т.е. распределение различных ресурсов компьютера. Программы этого уровня называются *системными*. Любые пользовательские программы запускаются под управлением системных программ, называемых *операционными системами*.

Следующий уровень – служебное ПО. Программы этого уровня, называемые утилитами, выполняют различные вспомогательные функции. Это могут быть ремонтные или диагностические программы, используемые при обслуживании различных устройств, тестовые программы, представляющие комплекс программ технического обслуживания, архиваторы, антивирусы и т.п. Служебные программы, как правило, работают под управлением операционной системы. В некоторых классификациях системный и служебный уровни объединяются в один класс – системного ПО.

Инструментальное ПО представляет комплекс программ для создания других программ. Программы инструментального ПО управляются системными программами.

Прикладное программное обеспечение – самый большой по объему класс программ. Это программы конечного пользователя, с помощью которых на данном рабочем месте выполняются конкретные задания. Спектр этих заданий очень широк – от производственных до творческих и развлекательно - обучающих. ППО также управляется системными программами, и имеет более высокий уровень.

В настоящее время программные продукты многих фирм стали объединять в себе программные элементы из разных классов. Например, ОС Windows, являясь комплексом системных программ, в своем составе содержит блок служебных программ (дефрагментация, проверка, очистка диска и др.), а также текстовый процессор Word Pad, графический редактор Paint, принадлежащих к классу прикладных программ.

### 1.1.3. Цели и задачи курса "Информатика"

**Главная функция** информатики – разработка методов и средств преобразования информации и их использовании технологического процесса переработки информации.

**Задачи информатики** состоят в следующем:

Исследование информационных процессов любой природы;

Разработка информационной техники и создание новейших технологий переработки информации на базе полученных результатов исследований информационных процессов;

Решение научных и инженерных проблем создания, внедрения и обеспечения эффективного использования компьютерной техники и технологии во всех сферах общественной жизни.

Информатика существует не сама по себе, а является комплексной научно - технической дисциплиной, призванной создавать новые информационные техники и технологии для решения проблем в других областях. Особенно следует выделить в информатике методы математического моделирования и методы распознавания образов, практическая реализация которых стала возможной благодаря достижениям компьютерной техники.

Комплекс индустрии информатики станет ведущим в информационном обществе. Тенденция ко все большей информатизации общества в существенной степени зависит от прогресса информатики как единства науки, техники и производства.

Вопросы для самоконтроля

1. Как появилась и развивалась наука «Информатика»?
2. Опишите структуру информатики.
3. Каковы главная функция и задачи информатики?

## ***Лекция 1.2. Сигналы. Данные. Информация***

### **1.2.1. Понятия данные и информация**

Термин «информация» используется во многих науках и во многих сферах человеческой деятельности. Он происходит от латинского слова «informatio», «сведения, разъяснения, изложение». С точки зрения науки «Информатики», «информация» является первичным, а, следовательно, неопределенным понятием. Так же как понятия «точка» в математике, «тело» в механике, «поле» в физике.

Понятию «информация» не возможно дать строгое определение, но можно описать его через проявляемые свойства.

Как известно, в материальном мире все физические объекты, окружающие нас, являются либо телами, либо полями. Физические объекты, взаимодействуя друг с другом, порождают сигналы различных типов. В общем случае любой сигнал – это изменяющийся во времени физический процесс. Такой процесс может иметь различные характеристики. Та из них, которая используется для представления данных, называется параметром сигнала. Если параметр сигнала принимает ряд последовательных, конечных во времени, значений, то сигнал называется дискретным. Если параметр сигнал – непрерывная во времени функция, то сигнал называется непрерывным.

Сигналы, в свою очередь, могут порождать в физических телах изменения свойств. Это явление называется регистрацией сигналов. Сигналы, зарегистрированные на материальном носителе, называются **данными**.

Существует большое количество физических методов регистрации сигналов на материальных носителях. Это могут быть механические воздействия, перемещения, изменения формы или магнитных, электрических, оптических параметров, химического состава, кристаллической структуры. В соответствии с методами регистрации, данные могут храниться и транспортироваться на различных носителях.

Наиболее часто используемый и привычный носитель – бумага; сигналы регистрируются путем изменения ее оптических свойств. Сигналы могут быть зарегистрированы и путем изменения магнитных свойств полимерной ленты с нанесенным ферромагнитным покрытием, как это делается в магнитофонных записях, и путем изменения химических свойств в фотографии.

Данные несут информацию о событии, но не являются самой информацией, так как одни и те же данные могут восприниматься (отображаться или интерпретироваться) по-

разному. Например, текст, написанный на русском языке (т.е. данные), даст различную информацию человеку, знающему алфавит и язык, и человеку, не знающему их.

Чтобы получить информацию, имея данные, к ним необходимо применить методы, которые преобразуют данные в понятия, воспринимаемые человеческим сознанием. Методы, в свою очередь, тоже различны. Например, человек, знающий язык и алфавит, применяет адекватный метод, читая русский текст. Соответственно, человек, не знающий русского языка и алфавита, применяет неадекватный метод, пытаясь понять русский текст. Таким образом, можно считать, что информация – это продукт взаимодействия данных и адекватных методов. Или иными словами: информация – это используемые данные. Информация не является статическим объектом, она появляется и существует в момент слияния методов и данных, а все прочее время она находится в форме данных. Момент слияния данных и методов называется информационным процессом (рис. 1.1).

Человек воспринимает первичные данные различными органами чувств (их у нас пять – зрение, слух, осязание, обоняние, вкус), и на их основе сознанием могут быть построены вторичные абстрактные (смысловые, семантические) данные.



Рис. 1.1. Формирование информации

Т.о. первичная информация может существовать в виде рисунков, фотографий, звуковых, вкусовых ощущений, запахов. Вторичная – в виде чисел, символов, текстов, чертежей, радиоволн, магнитных записей.

## 1.2.2. Свойства информации

Понятие «информация» имеет большое количество разнообразных свойств, но для дисциплины информатика наиболее важными являются следующие свойства:

1. **Дуализм** характеризуется двойственностью информации. С одной стороны, информация объективна в силу объективности данных. С другой стороны – субъективна, в силу субъективности применяемых методов. Иными словами, методы могут вносить в большей или меньшей степени субъективный фактор и т.о. влиять на информацию в целом. Например, два человека читают одну и ту же книгу и получают подчас весьма разную информацию, хотя прочитанный текст, т.е. данные, были одинаковы. Более объективная информация применяет методы с меньшим субъективным элементом.

2. **Полнота** информации характеризует степень достаточности данных для принятия решения или создания новых данных на основе имеющихся. Неполный набор данных оставляет большую долю неопределенности, т.е. большое число вариантов выбора, а это потребует применения дополнительных методов, например, экспертных оценок, бросания жребия и т.п. Избыточный набор данных затрудняет доступ к нужным данным, создает повышенный информационный шум, что также вызывает необходимость дополнительных методов, например, фильтрацию, сортировку. И неполный и избыточный наборы данных затрудняют получение информации и принятие адекватного решения.

3. **Достоверность** – это свойство, характеризующее степень соответствия информации реальному объекту с необходимой точностью. Измеряется достоверность информации *доверительной вероятностью* необходимой точности, т.е. вероятностью того, что отображаемое информацией значение параметра отличается от истинного значения этого параметра в пределах необходимой точности.

4. **Адекватность** информации выражает степень соответствия создаваемого на основе информации образа реальному объекту, процессу, явлению. Адекватность информации может выражаться в трех формах:

- *Синтаксическая* адекватность отображает формально-структурные характеристики информации и не затрагивает ее смыслового содержания. На синтаксическом уровне учитываются тип носителя и способ представления информации, скорость передачи и обработки, размеры кодов представления информации, надежность и точность преобразования этих кодов и т.п. Информацию, рассматриваемую только с синтаксических позиций, обычно называют данными, т.к. при этом не имеет значения смысловая сторона. Эта форма способствует восприятию внешних, структурных характеристик;
- *Семантическая* (смысловая) адекватность определяет степень соответствия образа объекта и самого объекта и предполагает учет смыслового содержания информации. Эта форма служит для формирования понятий и представлений, выявления смысла, содержания информации и ее обобщения;
- *Прагматическая* (потребительская) адекватность отражает отношение информации и ее потребителя, соответствие информации цели управления, которая на ее основе реализуется. Прагматические свойства информации проявляются только при наличии единства информации (объекта), пользователя и цели управления. Прагматический аспект рассмотрения связан с ценностью, полезностью использования информации при выработке потребителем решения для достижения своей цели.

5. **Доступность** информации, Информация должно быть доступна восприятию пользователя. Доступность обеспечивается выполнением соответствующих процедур ее получения и преобразования. Например, в ИС информация преобразовывается к доступной и удобной для восприятия пользователя форме.

6. **Актуальность информации.** Информация существует во времени, т.к. существуют во времени все информационные процессы. Информация, актуальная сегодня, может стать совершенно ненужной по истечении некоторого времени. Например, программа телепередач на нынешнюю неделю будет неактуальна для многих телезрителей на следующей неделе.

### 1.2.3. Понятие количества информации

Свойство полноты информации предполагает, что имеется возможность измерять количество информации. Какое количество информации содержится в книге или популярной песенке? Что содержит больше информации роман «Война и мир» или сообщение, полученное от товарища? Ответы на подобные вопросы не просты и не однозначны, т.к. во всякой информации присутствует субъективная компонента. А возможно ли вообще объективно измерить количество информации?

Важнейшим результатом теории информации является вывод о том, что в определенных, весьма широких условиях, можно, пренебрегая качественными особенностями информации, выразить ее количество числом, а следовательно, сравнивать количество информации, содержащейся в различных группах данных.

Количеством информации называют числовую характеристику информации, отражающую степень неопределенности, которая исчезает после получения информации.

Пример, Дома осенним утром старушка предположила, что могут быть осадки, а могут и не быть, а если будут, то в форме снега или в форме дождя, т.е. «бабушка надвое сказала – то ли будет, то ли нет, то ли дождик, то ли снег». Затем, выглянув в окно, увидела пасмурное небо и с большой вероятностью предположила – осадки будут, то есть, получив информацию, снизила количество вариантов выбора. Далее, взглянув на термометр, она увидела, что температура отрицательная, значить осадки следует ожидать в виде снега. Т.о., получив последние данные о температуре, бабушка получила полную информацию о предстоящей погоде и исключила все, кроме одного, варианты выбора.

Данный пример показывает, что понятия «информация», «неопределенность», «возможность выбора» тесно связаны. Полученная информация уменьшает число

возможных вариантов выбора (т.е. неопределенность), а полная информация не оставляет вариантов вообще.

За единицу информации принимается один **бит** (binary digit – двоичная цифра). Это количество информации, при котором неопределенность, т.е. количество вариантов выбора, уменьшается вдвое или, другими словами, это ответ на вопрос, требующий односложного разрешения – да или нет.

Бит – слишком мелкая единица измерения информации. На практике, чаще применяются более крупные единицы, например, байт – последовательность из восьми бит. Именно восемь бит, или один байт, используется для того, чтобы закодировать символы алфавита, клавиши клавиатуры компьютера. Один байт также является минимальной единицей адресуемой памяти компьютера, т.е. обратиться в память можно к байту, а не биту.

Широко используются еще более крупные производные единицы информации:

1 байт = 8 бит

1 Кбайт = 1024 байт

1 Мбайт = 1024 Кбайт

1 Гбайт = 1024 Мбайт

1 Тбайт = 1024 Гбайт.

Вопросы для самоконтроля

1. Что называется информацией?
2. В чем различие информации и данных?
3. Что такое адекватность информации, и в каких формах она проявляется?
4. Какие показатели качества информации вы знаете?
5. Как и чем измеряется информация?

## ***Лекция 1.3. Информационный процесс. Основные операции с данными***

### **1.3.1. Общая характеристика процессов сбора, передачи, обработки и накопления информации**

Получение информации тесно связано с информационными процессами. Рассмотрим их виды.

Сбор данных – это деятельность субъекта по накоплению данных с целью обеспечения достаточной полноты. Соединяясь с адекватными методами, данные рожают информацию, способную помочь в принятии решения. Например, интересуясь ценой товара, его потребительскими свойствами, мы собираем информацию для того, чтобы принять решение: покупать или не покупать его.

Передача данных – это процесс обмена данными. Предполагается, что существует источник информации, канал связи, приемник информации, и между ними приняты соглашения о порядке обмена данными, эти соглашения называются протоколами обмена. Например, в обычной беседе между людьми негласно принимается соглашение, не перебивать друг друга во время разговора.

Хранение данных – это поддержание данных в форме, постоянно готовой к выдаче их потребителю. Одни и те же данные могут быть востребованы не однажды, поэтому разрабатывается способ их хранения (обычно на материальных носителях) и методы доступа к ним по запросу потребителя.

Обработка данных – это процесс преобразования информации от исходной ее формы до определенного результата. Сбор, накопление, хранение информации часто не являются конечной целью информационного процесса. Чаще первичные данные привлекаются для решения какой-либо проблемы, затем они преобразуются шаг за шагом в соответствии с алгоритмом решения задачи до получения выходных данных, которые после анализа пользователем предоставляют необходимую информацию.

## Лекция 1.4. Системы счисления

### 1.4.1. Понятие системы счисления, как способа представления чисел

**Система счисления** – принятый способ записи чисел и сопоставления этим записям реальных значений. Все системы можно разделить на два класса: **позиционные** и **непозиционные**. Для записи чисел в различных системах счисления используется некоторое количество отличных друг от друга знаков. Число таких знаков в позиционной системе счисления называется **основанием системы счисления**.

Таблица 1.

#### Некоторые системы счисления

Основание	Система счисления	Знаки
2	Двоичная	0,1
3	Троичная	0,1,2
4	Четвертичная	0,1,2,3
5	Пятеричная	0,1,2,3,4
...	...	...
8	Восьмеричная	0,1,2,3,4,5,6,7
10	Десятичная	0,1,2,3,4,5,6,7,8,9
12	Двенадцатеричная	0,1,2,3,4,5,6,7,8,9,A,B
16	Шестнадцатеричная	0,1,2,3,4,5,6,7,8,9,A ,B,C,D,E,F

В позиционной системе счисления число может быть представлено в виде суммы произведения коэффициентов на степени основания системы счисления:

$$A_n A_{n-1} A_{n-2} \dots A_1 A_0, A_{-1} A_{-2} \dots = A_n * B^n + A_{n-1} * B^{n-1} + A_{n-2} * B^{n-2} + \dots + A_1 * B^1 + A_0 * B^0 + A_{-1} * B^{-1} + A_{-2} * B^{-2} + \dots$$

(здесь знак «точка» отделяет целую часть от дробной; знак «звездочка» используется для обозначения операции умножения;  $B$  – основание системы счисления;  $A$  – коэффициенты). Таким образом, значение каждого знака в числе зависит от позиции, которую занимает знак в записи числа. Именно поэтому такие системы счисления называют позиционными (система счисления обозначается индексом в скобках справа от числа).

Примеры:

$$23,43_{(10)} = 2 * 10^1 + 3 * 10^0 + 4 * 10^{-1} + 3 * 10^{-2} = 23,43_{(10)}$$

$$692_{(10)} = 6 * 10^2 + 9 * 10^1 + 2 * 10^0 = 692_{(10)}$$

$$1101_{(2)} = 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 13_{(10)}$$

$$112_{(3)} = 1 * 3^2 + 1 * 3^1 + 2 * 3^0 = 13_{(10)}$$

$$341,5_{(8)} = 3 * 8^2 + 4 * 8^1 + 1 * 8^0 + 5 * 8^{-1} = 225,625_{(10)}$$

$$A1F,4_{(16)} = A * 16^2 + 1 * 16^1 + F * 16^0 + 4 * 16^{-1} = 2591,25_{(10)}$$

Кроме позиционных систем счисления существуют такие, в которых значение знака не зависит от того места, которое он занимает в числе. Такие системы счисления называются непозиционными. Наиболее известным примером непозиционной системы является римская. В этой системе используется 7 знаков (I, V, X, L, C, D, M), которые соответствуют следующим величинам:

I (1), V (5), X (10), L (50), C (100), D (500), M (1000).

### 1.4.2. 10-ая, 2-ая, 8-ая и 16-ая системы как примеры позиционных систем

При работе с компьютером приходится параллельно использовать несколько позиционных систем счисления (чаще всего двоичную – т.к. внутреннее представление

информации в компьютере является двоичным, десятичную – т.к. мы привыкли к этой системе со школы и шестнадцатеричную – для сокращения записи чисел). Поэтому большое практическое значение имеют процедуры перевода чисел из одной системы счисления в другую. Обратите внимание, что во всех приведенных примерах результат является десятичным числом. Таким образом, представляя число в виде суммы произведений коэффициентов на степени основания, можно перевести число из любой системы счисления в десятичную.

### Перевод десятичных чисел в систему с основанием В

Целая и дробная части десятичного числа переводятся порознь.

Чтобы перевести целую часть числа из десятичной системы счисления в систему с основанием В, необходимо разделить ее на В. Остаток даст младший разряд числа. Полученное при этом частное необходимо вновь разделить на В – остаток даст следующий разряд числа и т.д. Деление производят до тех пор, пока частное от деления не станет меньше основания В. Последнее частное, будет самым старшим разрядом числа.

Примеры:	Остаток
25:2=12	(1)
12:2=6	(0)
6:2=3	(0)
3:2=1	(1)

Значения получившихся остатков, взятые в обратной последовательности, образуют искомое двоичное число.

Таким образом,  $25_{(10)} = 11001_{(2)}$

$125_{(10)} \rightarrow X_{(8)} = 175_{(8)}$

125:8=15	(5)
15:8=1	(7)

$1234_{(10)} \rightarrow X_{(16)} = 4DC_{(16)}$

1234:16=77	(12) C
77:16=4	(13) D

Для перевода дробной части ее необходимо умножить на В. Целая часть полученного произведения будет первым (после запятой) знаком. Дробную же часть произведения необходимо вновь умножить на В. Целая часть полученного числа будет следующим знаком после запятой и т.д. до нужного количества знаков после запятой.

Например:

$0,73 \cdot 2 = 1,46$  (целая часть 1),  
 $0,46 \cdot 2 = 0,92$  (целая часть 0),  
 $0,92 \cdot 2 = 1,84$  (целая часть 1),  
 $0,84 \cdot 2 = 1,68$  (целая часть 1) и т.д.

В итоге  $0,73_{(10)} = 0,1011\dots_{(2)}$ .

С точки зрения изучения принципов представления информации, в компьютере системы счисления представляют большой интерес. Хотя компьютер «знает» только двоичную систему счисления, часто с целью уменьшения количества записываемых на бумаге или вводимых с клавиатуры знаков бывает удобнее пользоваться восьмеричными или шестнадцатеричными числами. С практической точки зрения представляет интерес процедура взаимного преобразования двоичных, восьмеричных и шестнадцатеричных чисел. Для этого используется таблица соответствия чисел в различных системах счисления (Таблица 2).

Таблица 2.

**Соответствие чисел в различных системах счисления**

Десятичная	Восьмеричная	Шестнадцатеричная	Двоичная
0	0	0	0
1	1	1	1
2	2	2	10
3	3	3	11
4	4	4	100
5	5	5	101
6	6	6	110
7	7	7	111
8	10	8	1000
9	11	9	1001
10	12	A	1010
11	13	B	1011
12	14	C	1100
13	15	D	1101
14	16	E	1110
15	17	F	1111

Для перевода целого двоичного числа в восьмеричное необходимо разбить его справа налево на группы по 3 цифры (двоичные триады). Самая левая группа может содержать менее трех двоичных цифр. Затем каждой группе поставить в соответствие ее восьмеричный эквивалент. Например:

$$11011001_{(2)} = 11\ 011\ 001_{(2)} = 331_{(8)}.$$

Перевод целого двоичного числа в шестнадцатеричное производится аналогично путем разбиения данного числа на группы по четыре цифры (двоичные тетрады):

$$1100011011001_{(2)} = 1\ 1000\ 1101\ 1001_{(2)} = 18D9_{(16)}.$$

Для перевода дробных частей двоичных чисел в восьмеричную или шестнадцатеричную системы аналогичное разбиение на триады или тетрады производится от точки вправо (с дополнением недостающих последних цифр нулями):

$$0,1100011101_{(2)} = 0,110\ 001\ 110\ 100_{(2)} = 0,6164_{(8)}$$

$$0,1100011101_{(10)} = 0,1100\ 0111\ 0100_{(2)} = 0,C74_{(16)}.$$

Перевод восьмеричных (шестнадцатеричных) чисел в двоичные производится обратным путем – сопоставлением каждому знаку числа соответствующей тройки (четверки) двоичных цифр.

Перевод восьмеричных в шестнадцатеричные числа и наоборот производится с использованием двоичной или десятичной системы в качестве промежуточной.

**1.4.3. Арифметические операции над двоичными числами**

Над числами, записанными в любой системе счисления можно производить различные арифметические операции. Так, для сложения и умножения двоичных чисел необходимо использовать следующие таблицы:

**Таблицы сложения и умножения в двоичной системе**

+	0	1
0	0	1
1	1	10

$$\begin{array}{r|l} * & 0 \ 1 \\ \hline 0 & 0 \ 0 \\ 1 & 0 \ 1 \end{array}$$

Заметим, что при двоичном сложении  $1 + 1$  возникает перенос единицы в старший разряд – точь-в-точь как в десятичной арифметике:

$$\begin{array}{r} 1001 \\ + \ 11 \\ \hline = 1100 \end{array} \qquad \begin{array}{r} 1001 \\ * \ 11 \\ \hline 1001 \\ + 1001 \\ \hline = 11011 \end{array}$$

### **Вопросы для самоконтроля**

1. Что понимается под системой счисления?
2. Какие системы счисления вы знаете?
3. В чем отличие позиционной системы счисления от непозиционной ?
4. Какие системы счисления используются в вычислительной технике, и в чем заключается преимущество их использования?
5. Что называется основанием системы счисления?
6. Каковы способы перевода чисел из одной системы счисления в другую?
7. Как выглядят таблицы сложения и умножения в двоичной системе счисления?

## ***Лекция 1.5. Единицы представления, измерения и хранения данных***

### **1.5.1. Представление чисел в двоичном коде**

Представление чисел в памяти компьютера имеет специфическую особенность, связанную с тем, что в памяти компьютера они должны располагаться в байтах – минимальных по размеру адресуемых ячейках памяти. Вся память компьютера разбита на отдельные участки из 8 бит (т.е. на байты). Байты имеют номера: 0, 1, 2, ..., называемые адресами. Два соседних участка: 0 и 1, 2 и 3 и т.д. образуют ячейку памяти ЭВМ. Т.е. одна ячейка памяти может хранить два байта или 16 битов информации. Для содержимого одной ячейки используют название – «машинное слово» или просто «слово». Очевидно, адресом числа считается адрес первого байта, т.е. четные числа, начиная с 0, 2, 4, 6 и т.д. В байте может содержаться произвольный код из восьми двоичных разрядов, и задача представления состоит в том, чтобы указать правила, как в одном или нескольких байтах записать число.

Числа могут быть целые точные, дробные точные, рациональные, иррациональные, дробные приближенные, положительные и отрицательные. Числа могут быть «карликами» (например, масса атома), «гигантами» (например, масса земли), реальными (например, количество студентов в группе, рост, возраст). И каждое из чисел потребует для оптимального представления в памяти свое количество байтов.

Единого оптимального представления для действительных чисел создать невозможно. Поэтому множества чисел разделили на типы (например, целые в диапазоне от ... до ..., приближенные с плавающей точкой с количеством значащих цифр ... и т.д.). Для каждого в отдельности типа создается собственный способ представления.

Целые числа. Целые положительные числа от 0 до 255 можно представить непосредственно в двоичной системе счисления (двоичном коде). Такие числа будут занимать один байт в памяти компьютера.

Числа	Двоичный код
0	00000000
1	00000001
2	00000010
3	00000011
...	.....
255	11111111



Знак  
мантиссы

Первый разряд представления используется для записи знака мантиссы. За ним – группа разрядов, определяющих порядок, а остальные разряды определяют абсолютную величину мантиссы. Размеры обеих групп разрядов фиксируются. Т.к. порядок может быть положительным или отрицательным, нужно решить проблему его знака. Величина порядка представляется с избытком, т.е., вместо истинного значения порядка хранится число, называемое характеристикой (или смещенным порядком). Для получения характеристики надо к порядку прибавить смещение. Например, при использовании для хранения порядка восьми бит и значений от  $-128$  до  $+127$  используется смещение 128. Тогда для представления порядка будут использоваться значения от 0 до 255, т.е. только неотрицательные числа.

Т.к. мантисса нормализованного числа всегда равна 1, некоторые схемы представления ее лишь подразумевают, используя лишний разряд для повышения точности представления мантиссы.

Использование смещенной формы позволяет производить операции над порядками, как над без знаковыми числами, что упрощает операции сравнения, сложения и вычитания порядков. А также упрощает операцию сравнения самих нормализованных чисел. Чем больше разрядов отводится под запись мантиссы, тем выше точность представления числа. Чем больше разрядов занимает порядок, тем шире диапазон от наименьшего, отличного от нуля числа до наибольшего числа, представимого в компьютере при заданном формате.

Как и в случае целых чисел, в программных системах могут использоваться несколько типов данных, реализующих модель с плавающей точкой. Например, в языке СИ применяются три типа данных с разной «длиной». Шестнадцатиразрядные компиляторы для IBM-совместимых ПК реализуют эти типы следующим образом:

- Float – 4 байта, из них 23 разряда мантиссы и 8 битов порядка  
(от  $3,4 \cdot 10^{-38}$  до  $3,4 \cdot 10^{38}$ , обеспечивает точность с 7 значащими цифрами);
- Double – 8 байтов, из них 52 разряда мантиссы и 11 битов порядка  
(от  $1,7 \cdot 10^{-308}$  до  $1,7 \cdot 10^{308}$ , обеспечивает точность с 15 знаками);
- Long double – 10 байтов, из них 65 разрядов мантиссы и 14 битов порядка  
(от  $3,4 \cdot 10^{-4932}$ , обеспечивает точность с 19 знаками).

Понятие типа данных. Мы уже говорили, что минимально адресуемой единицей памяти является байт, но представление числа требует большего объема. Такие числа займут группу байт, а адресом числа будет адрес первого байта группы. Следовательно, произвольно взятый из памяти байт ничего не скажет о том, частью какого информационного объекта он является – целого числа, числа с плавающей точкой или командой. Отсюда можно сделать вывод, что кроме задачи представления данных в двоичном коде, параллельно решается обратная задача – интерпретации кодов, т.е. как из кодов восстановить первоначальные данные.

Для представления основных видов информации (числа целые, числа с плавающей точкой, символы, звук и т.д.) в системах программирования используют типы данных. Каждый тип данных определяет логическую структуру представления и интерпретации для соответствующих данных.

## 1.5.2. Представление символьных и текстовых данных

Тесты являются важнейшим источником информации. Именно такой характер имеют экономические, плановые, учетные данные, представленные на естественном или искусственном языке. Каждый язык использует свою знаковую систему, основанную на

алфавите. Письменность можно рассматривать как метод представления на материальных носителях знаков звуковой системы разговорного языка.

Рассмотрим, как кодируются символы, элементы текстов, текстовые документы.

Символы. Двоичное кодирование символьных данных производится заданием кодовых таблиц, согласно которым каждому символу ставят в соответствие одно – или двухбайтовый код. Помимо этого, кодовая таблица ставит в соответствие кодам клавиши на клавиатуре и начертание символа на экране монитора.

Кодовые таблицы разрабатывались в разных странах в разные времена. Поэтому, как правило, в одном языке существуют несколько кодовых таблиц, что осложняет обратную задачу интерпретации кодов.

Наиболее популярная таблица ASCII разработана институтом стандартизации США в 1981 г. Ее использовали, в частности, программные продукты, работающие под управлением ОС MS DOS. Для представления одного символа используется один байт (8 бит), т.е. кодовая таблица описывает  $2^8 = 256$  различных кодов.

Коды с 0 до 127 составляют базовую таблицу; коды со 128 по 255 – расширенную (дополнительную) таблицу.

В основной таблице располагаются управляющие команды для принтеров (коды 0 – 31 «перевод строки», «возврат каретки», им не соответствуют символы), затем спец. символы, знаки арифметических действий и знаки препинания, цифры, латинские буквы – прописные и строчные.

Дополнительная таблица отдана национальным алфавитам, символам псевдографики (с помощью которых форматируются таблицы).

Позднее, при разработке ОС Windows, была создана кодовая таблица Windows – 1251, в которой базовая таблица осталась прежней, а расширенная изменилась. В целом, существование в нашей стране нескольких кодовых таблиц порождает задачу межсистемного преобразования данных.

Во многих странах Азии 256 кодов явно не хватает. В 1991 г. производители программных продуктов (Microsoft, IBM, Apple) и стандартизаторы пришли к соглашению о выработке единого стандарта ISO 10646-1 (он же Unicode 3.0). Код построен по 31-битной схеме, но используются только два байта для кодирования одного символа. Два байта (16 бит) создают  $2^{16} = 65536$  кодов, которые описывают цифры, буквы латинского и многих национальных алфавитов, спец. Символы, знаки арифметических операций и т.д. Все текстовые документы в этой кодировке вдвое длиннее, что сначала задерживало ее внедрение, но современный уровень технических средств дает такую возможность. В настоящее время, текстовый процессор Word, начиная с версии Word 8.0 (Microsoft Office 97), использует шрифты Unicode 3.0.

Текстовые строки. Текстовая (символьная) строка – это конечная последовательность символов. Это может быть осмысленный текст или произвольный набор, короткое слово или целая книга. Длина символьной строки – это количество символов в ней. Записывается в память символьная строка двумя способами: либо число, обозначающее длину текста, затем текст, либо текст, а затем разделитель строк.

Текстовые документы. Текстовые документы используются для хранения и обмена данными в ИС, но сплошной, не разбитый на логические фрагменты текст воспринимается тяжело. Структурирование текста достигается форматированием – специфическим расположением текста при подготовке его к печати. Для анализа структуры текста были разработаны языки разметки, которые вместе с основным текстом записывают в текстовом формате текстовые метки (маркеры и теги), используемые для обозначения частей документа.

#### 1.5.4. Представление звуковых данных

Звук – это упругая продольная волна в воздушной среде. Чтобы ее представить в виде, читаемом компьютером, необходимо выполнить следующие преобразования (рис. 1.5.1).

Звуковой сигнал преобразовать в электрический аналог звука с помощью микрофона. Электрический аналог получается в непрерывной форме и непригоден для обработки на цифровом компьютере. Чтобы перевести сигнал в цифровой код, надо пропустить его через аналого-цифровой преобразователь (АЦП). При воспроизведении происходит обратное преобразование – цифро – аналоговое (ЦАП). АЦП и ЦАП находятся в звуковой карте компьютера.

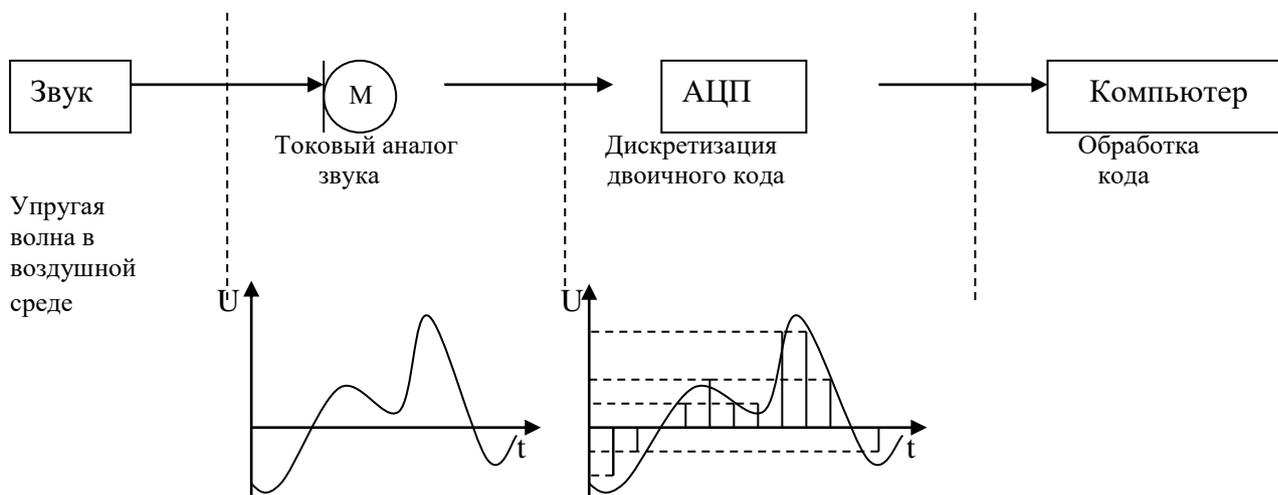


Рис. 1.5.1. Схема обработки звукового сигнала

Во время оцифровки сигнал дискретизируется по времени и по уровню. Дискретизация по времени выполняется следующим образом: весь период времени  $T$  разбивается на малые интервалы времени  $\Delta t$ , точками  $t_1, t_2, \dots, t_n$ . Предполагается, что в течение интервала  $\Delta t$  уровень сигнала изменяется незначительно и может с некоторым допущением считаться постоянным. Величина  $\nu = \frac{1}{\Delta t}$  называется частотой дискретизации.

Она измеряется в герцах. (Гц) – количество измерений в течение сек.

Дискретизация по уровню называется квантованием и выполняется так: область измерения сигнала от самого малого значения  $X_{\min}$  до самого большого  $X_{\max}$  разбивается на  $N$  равных квантов, промежутков величиной  $\Delta x = (X_{\max} - X_{\min}) / N$  точками  $x_1, x_2, \dots, x_n, x_i = X_{\min} + \Delta x * (i - 1)$ . Каждый квант связывается с его порядковым номером, т.е. целым числом, которое легко может быть представлено в двоичной С.С. Если сигнал после дискретизации по времени попадает в промежуток  $x_{i-1} \leq x \leq x_i$ , то ему в соответствие ставится код  $i$ .

Теорема Найквиста утверждает, что, если сигнал оцифрован с частотой  $\nu$ , то высшая «слышимая» частота будет не более  $\nu/2$ . Число уровней подбирается так, чтобы звук не имел высокого уровня шума и «электронного» оттенка звучания. Число уровней берется как  $2^n$ . Чтобы измерение занимало целое число байт, выбирают  $n=8$  или  $n=16$ , т.е. один или два байта.

Высокое качество воспроизведения получается в формате лазерного аудиодиска при следующих параметрах оцифровки: частота дискретизации – 44,1 Гц, квантование – 16 бит, т.е.  $\Delta x = (X_{\max} - X_{\min}) / 2^{16}$ . Т.о. 1 сек стерео звука займет  $2 \text{ байт} * 44100 \text{ байт/с} * 2 \text{ кан.} * 1 \text{ с} = 176400 \text{ байт}$  дисковой памяти. При этом качество звука получается очень высоким.

Для телефонных переговоров удовлетворительное качество получается при частоте дискретизации 8 кГц и частоте квантования 255 уровней, т.е. 1 байт, при этом 1 сек звуковой записи, займет на диске  $1 \text{ байт} * 8000 \text{ байт/с} * 1 \text{ с} = 8000 \text{ байт}$ .

### 1.5.7. Представление графических данных. Понятие растровой и векторной графики

Изображения представляются двумя способами.

1. Графические объекты создаются как совокупности линий, векторов – называется векторной графикой.
2. Графические объекты формируются в виде множества точек (пикселей) разных цветов и разных яркостей, распределенных по строкам и столбцам – называется растровой графикой.

Модель RGB. Чтобы оцифровать цвет, его надо измерить. Немецкий ученый Гросман сформулировал три закона смешения цветов:

1. Закон трехмерности – любой цвет может быть представлен комбинацией трех основных цветов;
2. Закон непрерывности – к любому цвету можно подобрать бесконечно близкий;
3. Закон аддитивности – цвет смеси зависит только от цвета составляющих.

За основные три цвета приняты красный (Red), зеленый (Green), синий (Blue). В модели RGB любой цвет получается в результате сложения основных цветов. Каждый составляющий цвет при этом характеризуется своей яркостью, поэтому модель называется аддитивной. Эта схема применяется для создания графических объектов в устройствах, излучающих свет – телевизорах, мониторах.

Модель CMYK. В полиграфических системах напечатанный на бумаге графический объект сам не излучает световых волн. Изображение формируется на основе отраженной волны от окрашенных поверхностей. Окрашенные поверхности на которые падает белый свет (т.е. сумма всех цветов), должны поглотить (т.е. вычесть) все составляющие цвета, кроме того, цвет которого мы видим. Цвет поверхности можно получить красителями, которые поглощают, а не излучают. Например, если мы видим зеленое дерево, то это означает, что из падающего белого цвета (т.е. суммы красного, зеленого, синего), поглощены синий и красный, а зеленый отражен. Цвета красителей должны быть дополняющими: голубой Cyan = B + G), дополняющий красного; пурпурный (Magenta = R + G), дополняющий зеленого; желтый (Yellow = R + B), дополняющий синего. Т.к. цветовые красители по отражающим свойствам не одинаковы, то для повышения контрастности применяется еще черный (Black). Модель CMYK названа по первым буквам слов Cyan, Magenta, Yellow и последней букве Black. Т.к. цвета вычитаются, модель называется субтрактивной.

Оцифровка изображения. При оцифровке изображение с помощью объектива проецируется на светочувствительную матрицу  $m$  строк и  $n$  столбцов, называемую растром. Каждый элемент матрицы – мельчайшая точка, при цветном изображении состоящая из трех светочувствительных (т.е. регистрирующих яркость) датчиков красного, зеленого, желтого цвета. Далее оцифровывается яркость каждой точки по каждому цвету последовательно по всем строкам растра.

Если для кодирования яркости каждой точки использовать по одному байту (8 бит) на каждый из трех цветов ( $3 * 8 = 24$  бита), то система обеспечит представление  $2^{24} \approx 16,7$  млн. распознаваемых цветов, что близко цветовосприятию человеческого зрения. Режим представления цветной графики двоичным кодом из 24 разрядов называется полноцветным или True Color. Графические данные, также как и звуковые, занимают очень большие объемы на носителях. Например, если экран монитора имеет растр  $800 \times 600$  точек, изображение, представленного в режиме True Color, займет  $800 \times 600 \times 3 = 1440000$  байт. Если не требуется высокое качество отображения цвета, применяют режим High Color, который кодирует одну точку растра 2 байтами (16 разрядов дают  $2^{16} \approx 65,5$  тысячи цветов).

Режим, который при кодировании одной точки растра использует один байт, называется индексным, в нем различаются 256 цветов. Этого не достаточно, чтобы передать весь диапазон цветов. Код каждой точки при этом выражает собственно не цвет, а некоторый номер цвета (индекс) из таблицы цветов, которая называется палитрой. Палитра

прикладывается к файлам с графическими данными и используется при воспроизведении изображения.

Графическими форматами файлов являются .TIFF, .GIF, .JPG, .BMP, JPEG.

### 1.5.8. Структуры данных

Автоматизированная обработка больших объемов данных становится проще, если данные упорядочены (структурированы). Применяются следующие структуры данных:

Линейная структура данных (список) – это упорядоченная структура, в которой адрес данного однозначно определяется его номером (индексом). Пример: список учебной группы, дома, стоящие на одной улице.

В списках, как известно, новый элемент начинается с новой строки. Если элементы располагаются в строчку, то вводится разделительный знак между элементами. Поиск осуществляется по разделителям (например, чтобы найти десятый элемент, надо отсчитать девять разделителей).

Если элементы списка одной длины, структура называется вектором данных, разделители не требуются. При длине одного элемента  $d$ , зная номер элемента  $n$ , его начало определяется соотношением  $d(n-1)$ .

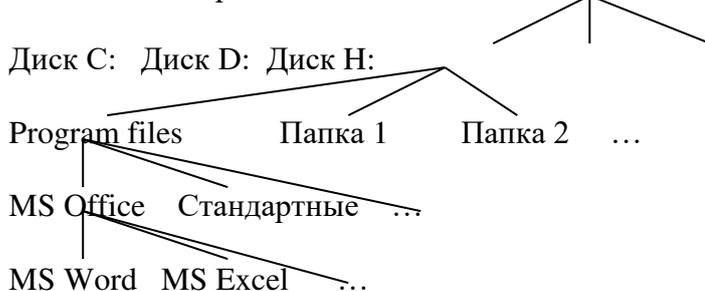
Табличная структура данных – это упорядоченная структура, в которой адрес данного однозначно определяется двумя числами – номером строки и номером столбца, на пересечении которых находится ячейка с исходным элементом.

Если элементы располагаются в строчку, вводятся два разделительных знака – между элементами строки и между строками. Поиск, аналогично линейной структуре осуществляется по разделителям.

Если элементы таблицы одной длины, структура называется матрицей данных, разделители в ней не требуются. При длине одного элемента  $d$ , зная номер строки  $m$  и номер столбца  $n$ , а также строк и столбцов  $M, N$ , найдем адрес его начала:  
 $d[N(m-1) + (n-1)]$ .

Таблица может быть трехмерной, тогда три числа характеризуют положение элемента, а может быть  $n$  – мерной.

Мой компьютер



Иерархическая структура – это структура, в которой адрес каждого элемента определяется путем (маршрутом доступа), идущим от вершины структуры к данному элементу. Иерархическую структуру образуют, например, почтовые адреса, файловая система компьютера:

Линейная и табличная структуры более просты, чем иерархическая, но, если в линейной структуре появляется новый элемент, то упорядоченность сбивается. Например, если в списке студентов появляется новый человек, расположенный по алфавиту список нарушается.

В иерархической структуре введение нового элемента не нарушает структуры дерева. Недостатком является трудоемкость записи адреса и сложность упорядочивания.

H:\Stud\tsi11Ivanov\Text\my\_file.doc

Россия \ Ростовская область \ Ростов – на – Дону \ ул. Б. Садовая \ дом 1 кв. 11

### 1.5.9. Хранение данных

При хранении данных решаются две задачи:

- как сохранить данные;
- как обеспечить быстрый удобный доступ к ним.

В компьютерных технологиях единицей хранения данных является объект переменной длины, называемый файлом. Файл – это поименованная область на внешнем носителе, содержащая данные определенной длины, обладающая собственным именем.

Современные файлы могут содержать данные различных типов. Например, в текстовом файле могут содержаться графические вставки, таблицы и пр.

Имя файла фактически несет в себе адресные функции в иерархических структурах. Имя файла может иметь расширение, в котором хранятся сведения о типе данных. При автоматической обработке по типу файла может запускаться приложение (программа), работающее с ним.

Для доступа к файлам, которых на жестком диске может быть сотни тысяч, используется определенное ПО, предназначенное для централизованного управления данными.

Системы управления данными называются файловыми системами. Именно файловая система решает задачи распределения внешней памяти, отображение имен файлов в соответствующие адреса во внешней памяти и обеспечение доступа к данным. Благодаря этому, работа с файлами во многом стала напоминать работу с обычными документами.

Информация любого типа хранится в виде файлов, выступающих в роли логически завершенных совокупностей данных.

Для пользователя файл является основным и неделимым элементом хранения данных, который можно найти, изменить, удалить, сохранить или переслать, но только целиком. С физической точки зрения, файл – это всего лишь последовательность байтов. Способ использования или отображения этой последовательности определяется типом файла – текстовый, звуковой, исполняемый модуль программы и т.п.

Для хранения различных видов информации необходимо использовать по-разному устроенные файлы. Способ организации данных (структура файла) называется форматом.

Файловая система – это часть ОС компьютера и поэтому всегда несет на себе отпечаток свойств конкретной ОС. Некоторые форматы файлов стандартизированы и должны поддерживаться любой ОС и работающими в ней приложениями (например, графические файлы .GIF или JPEG). Наряду с этим всегда имеются форматы, специфичные только для данной системы (например, исполняемые файлы MS DOS или Windows). Есть форматы, разработанные для конкретных приложений, работающих под управлением данной ОС (например, формат .xls, используемый MS Excel). В некоторых случаях при разработке приложений программистам приходится создавать новые форматы.

Структура файла может быть простой. Например, текст может сохраняться в виде последовательности байтов, прямо соответствующих формату ASCII. Но в большинстве случаев вместе с данными приходится сохранять и дополнительную информацию. Например, особенности форматирования текста (размеры символов, шрифты и т.п.). Процесс форматирования можно рассматривать, как процедуру придания некоторых свойств фрагменту текста. Поэтому, для сохранения форматирования нужно иметь два типа кодов, для обозначения блока текста, к которому применяется форматирование, и для указания свойства (типа форматирования).

Если система служебных кодов определена, то для сохранения текста требуется вставка в него управляющих символов. Например, надо сохранить таблицу из шести чисел, имеющую две строки и три столбца (2 x 3):

4	1	8
2	5	11

Для хранения чисел используются ячейки фиксированного размера (например, два байта). Поэтому, если в файл записано 6 чисел, то при чтении данных из него нужно извлечь шесть раз по 2 байта. Т.к. память ПК линейна, то возникает задача, как сохранить не только сами числа, но и структуру таблицы. Очевидно, что если таблица запоминается построчно, то основным параметром, определяемым ее вид, является количество чисел в строке (т.е. количество столбцов). Поэтому, договариваемся, что первое число в файле – это длина строки таблицы, затем записывается количество строк, а далее построчно сохраняются числовые элементы таблицы. Тип данных, представляющий их, также должен быть оговорен. Число строк необходимо запомнить, чтобы знать, где остановиться при чтении информации. Таблица может быть записана в файл в виде следующей последовательности:

3	2	4	1	8	2	5	11
---	---	---	---	---	---	---	----

Сформулированными правилами должна будет пользоваться не только программа, сохраняющая таблицу, но и любая другая, которой потребуется прочесть данные из таблицы, т.к. формат файла определяет способ правильной интерпретации хранимых данных. Размещение в начале файла блока служебной информации часто используется в многочисленных форматах, например, в файлах баз данных или графических данных.

Заголовок (служебная информация)	Собственно сохраняемые данные
-------------------------------------	-------------------------------

Часто заголовок файла включает идентификатор формата файла. Программы, предназначенные для просмотра файлов определенного типа, начинают работу с чтения служебной информации и проверки возможности восприятия формата файла.

Современные программные системы позволяют одновременно включать в файл данные разных видов, а это требует разработки очень сложных форматов. Придумать простой формат, который позволил бы хранить множество видов данных вместе, невозможно. Поэтому, например, для хранения документов в MS Office строятся так называемые структурированные хранилища – фактически файловые системы, спрятанные в одном файле. Наличие разных форматов для хранения данных одного и того же типа затрудняет переносимость их из среды одного приложения (программы) в среду другого. Проблема обычно решается использованием специальных программ, называемых конверторами.

Вопросы для самоконтроля

1. Как представляются числа в памяти компьютера?
2. Как представляются символьные и текстовые данные?
3. Как представляются звуковые данные?
4. Как представляются графические данные?
5. Какими способами формируются графические изображения?
6. Что вы можете рассказать о моделях RGB и CMYK?
7. Что вы можете сказать об особенностях хранения данных различных типов?

## ***Лекция 1.6. Понятие о методах сжатия информации***

### **1.6.1. Два типа методов сжатия данных и области их применения**

С моделями представления информации тесно связана еще одна проблема – сжатие информации.

При архивировании и передаче по каналам связи объем информации является основным параметром. Поэтому модели представления дополняются процедурами сжатия, т.е. плотной упаковкой информации.

Применяются два типа алгоритмов сжатия: сжатие с изменением структуры данных (оно происходит без потери данных) и сжатие с частичной потерей данных. Алгоритмы первого типа предусматривают две операции: сжатие информации для хранения или передачи и восстановление данных точно в исходном виде, когда их требуется использовать. Такой тип сжатия применяется для текстов (алгоритмы Хафмана, Keyword encoding и

Лемпеля – Зива). Алгоритмы второго типа не позволяют полностью восстановить оригинал и применяются для сжатия графики или звука; для текстовых, числовых данных или программ они не применяются.

Вопросы для самоконтроля

1. С какой целью применяют сжатие данных?
2. Какие типы алгоритмов сжатия вы знаете?
3. Какие методы сжатия можно использовать для текстовых файлов, а какие нельзя?
4. Какие методы сжатия можно использовать для графических файлов?

## ***Лекция 1.7. Основные этапы развития вычислительной техники***

### **1.7.1. История развития средств ВТ**

Слово «компьютер» означает «вычислитель», т.е. устройство для вычислений. Потребность в автоматизации обработки данных, в том числе и вычислений, возникла очень давно. Многие тысячи лет назад для счета использовались пальцы, счетные палочки, камешки, веревки с узелками. Более 1500 лет назад для вычислений стали использовать абак (разновидностью которого являются русские счеты).

В 1642 г. Блэз Паскаль изобрел устройство, механически выполняющее сложение чисел, а в 1673 г. Готфрид Вильгельм Лейбниц сконструировал арифмометр, позволяющий механически выполнять четыре арифметических действия. Начиная с XIX в. арифмометры получили очень широкое применение. На них выполняли даже артиллерийской стрельбы. Существовала и специальная профессия – счетчик – человек, работающий с арифмометром, быстро и точно соблюдающий определенную последовательность инструкций (такая последовательность инструкций получила название «программа»). Но многие расчеты производились медленно – даже десятки счетчиков должны были работать несколько недель и месяцев. Причина проста – при таких расчетах выбор выполняемых действий и запись результатов производились человеком, а скорость его работы весьма ограничена.

В первой половине XIX в. английский математик Чарльз Бэббидж попытался построить универсальное вычислительное устройство – **аналитическую машину** - которая должна была выполнять вычисления без участия человека. Для этого она должна была уметь исполнять программы, вводимые с помощью перфокарт (карт из плотной бумаги с информацией, наносимой с помощью отверстий, которые придумал Жакард, они в то время уже широко употреблялись в ткацких станках), и иметь «склад» для запоминания данных и промежуточных результатов (память). Бэббидж не смог довести до конца работу по созданию аналитической машины – она оказалась слишком сложной для техники того времени. Однако, он разработал все основные идеи. В 1943 г. американец Говард Эйкен с помощью работ Бэббиджа на основе электромеханических реле смог построить на одном из предприятий фирмы IBM такую машину под названием «Марк-1». Еще раньше идеи Бэббиджа были переоткрыты немецким инженером Конрадом Цузе, который в 1941 г. построил аналогичную машину.

К тому времени, потребность в автоматизации вычислений (в том числе и для военных нужд – баллистики, криптографии) уже стала настолько велика, что над созданием машин подобного типа одновременно работало несколько групп исследователей. Начиная с 1943 г. группа специалистов под руководством Джона Мочли и Преспера Экерта в США начала конструировать машину уже на основе электронных ламп, а не реле. Их машина, названная ENIAC, работала в тысячу раз быстрее, чем «Марк-1», однако для задания программы приходилось в течение нескольких часов или даже дней подсоединять нужным образом провода. Чтобы упростить процесс задания программы, Мочли и Экерт стали конструировать машину, которая бы могла хранить программу в своей памяти. В 1945 г. к работе был привлечен знаменитый математик Джон фон Нейман, который подготовил доклад об этой машине, в котором ясно и просто сформулировал общие принципы

функционирования универсальных вычислительных устройств, и разослал его многим ученым.

Первый компьютер, в котором были воплощены принципы фон Неймана, был построен в 1949 г. английским исследователем Морисом Уилксом.

### 1.7.2. Эволюция ЭВМ

В развитии вычислительной техники принято выделять пять поколений ЭВМ.

- **ЭВМ I поколения.** ЭВМ, элементной базой которой являлись лампы. Она обладала малым быстродействием и объемом памяти, неразвитой операционной системой, программированием на машинном языке. Использовалась в 50-е годы («Урал», БЭСМ).
- **ЭВМ II поколения.** ЭВМ, элементной базой которой являются полупроводники. Она имеет изменяемый состав внешних устройств, использует языки программирования высокого уровня и принцип библиотечных программ. Наиболее большое применение нашла в 60-е годы (БЭСМ-6, М-220, «Минск-32», «Весна»).
- **ЭВМ III поколения.** ЭВМ, характерными признаками которой являются интегральная элементная база, развитая конфигурация внешних устройств с использованием стандартных средств сопряжения, высокое быстродействие и большой объем основной и внешней памяти, развитая операционная система, обеспечивающая работу в мультипрограммном режиме. Появление первых ЭВМ этого поколения относится к началу 70-х годов (ЕС ЭВМ, СМ ЭВМ, ИВМ).
- **ЭВМ IV поколения.** ЭВМ, характерными признаками которой являются элементная база на основе больших интегральных схем (БИС), виртуальная память, многопроцессорность, параллелизм выполнения операций, развитые средства диалога. Появилась в середине 80-х годов («Эльбрус-2», ПЭВМ ЕС1841, ЕС1842, ИВМ РС).
- **ЭВМ V поколения.** ЭВМ, характерными признаками которой являются использование в качестве элементной базы сверхбольших интегральных схем (СБИС), применение принципа «управления потоками данных» (в отличие от принципа фон Неймана «управление потоками команд»), использование новых решений в архитектуре вычислительной системы и принципов искусственного интеллекта.

### Вопросы для самоконтроля

1. Какие ключевые события из истории развития вычислительной техники вам известны?
2. По каким признакам ЭВМ относят к тому или иному поколению?
3. Каковы совокупные признаки ЭВМ 1-, 2-, 3-, 4-го поколений?

### Аппаратные реализации информационных процессов.

#### *Лекция 2.1. Понятие вычислительной системы*

##### 2.1.1. Компьютеры, как системы обработки информации

Изыскание средств и методов механизации и автоматизации работ – одна из основных задач технических дисциплин. Автоматизация работ с данными имеет свои особенности и отличия от автоматизации других типов работ. Для этого класса задач используют особые виды устройств, большинство из которых являются электронными приборами. Совокупность устройств, предназначенных для автоматической или автоматизированной обработки данных, называют вычислительной техникой. Конкретный набор взаимодействующих между собой устройств и программ, предназначенный для обслуживания одного рабочего участка, называют вычислительной системой. Центральным устройством большинства вычислительных систем является компьютер.

## 2.1.2. Представление информации в технических устройствах

В основу любого устройства, предназначенного для преобразования или хранения информации, должен быть положен принцип ее представления, то есть ее физический носитель.

Вычислительные устройства, использующие непрерывную форму представления информации, называются аналоговыми вычислительными машинами (АВМ). Вычислительные устройства, использующие дискретную форму представления, называются цифровыми вычислительными машинами (ЦВМ).

В настоящее время устройства, использующие непрерывный способ представления информации, вытесняются более прогрессивными цифровыми устройствами, даже из таких традиционно «аналоговых» областей, как телевидение и телефония. Развитие вычислительных систем, начавшееся преимущественно с АВМ, постепенно перешло к ЦВМ и к середине 70-гг. прошлого столетия ЦВМ полностью вытеснили АВМ.

АВМ имели блочную структуру, т.е. представляли собой систему связанных между собой базовых элементов. Связи между базовыми элементами, их состав и количество изменялись для каждой задачи. В качестве базового элемента использовался операционный усилитель (рис. 2.1.1.):

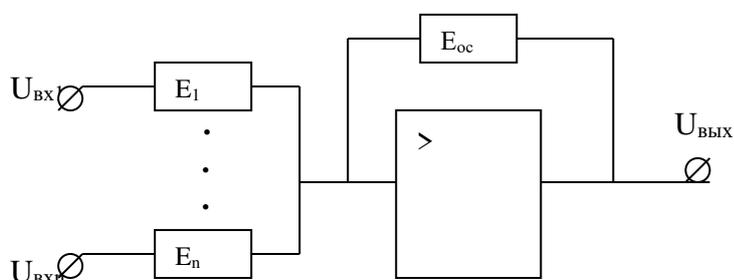


Рис. 2.1.1. Операционный усилитель

В качестве элементов используются радиоэлектронные компоненты: резисторы, конденсаторы, индуктивности. В зависимости от типов элементов, базовый элемент может производить сложение, интегрирование, дифференцирование и некоторые другие операции над входными напряжениями ( $U_{вх1}, \dots, U_{вхn}$ ), результат операции снимается в виде выходного напряжения ( $U_{вых}$ ).

ЦВМ имеют гораздо более высокую сложность аппаратной и программной реализации. Информация в них имеет определенные границы представления, т.е. точность представления информации конечна. Для расширения границ представления необходимо увеличивать аппаратную часть или увеличивать время обработки. Основными достоинствами ЦВМ являются:

- Гарантированная точность результата, зависящая только от границ представления данных;
- Универсальность – способность обрабатывать данные любыми методами, представляемыми последовательностью простых арифметических и логических операций;
- Возможность реализации большого числа известных численных, математических методов решения задач.

## 2.1.3. Физические и логические принципы работы компьютера. Базовые логические операции. Принцип выполнения программ.

Люди, далекие от техники, часто смотрят на ЭВМ и др. цифровые электронные устройства, как на нечто таинственное и непостижимое. Тем не менее, все эти устройства

работают в строгом соответствии с четкими логическими законами. Знание и понимание этих законов помогает в общении с компьютером.

В основе всех выводов компьютера лежат три основные операции: «И», «ИЛИ», «НЕ». Иногда эти операции называют «тремя китами машинной логики».

При записи логических выражений используется специальный язык, который принят в машинной логике. Основоположником мат. Логики является великий немецкий математик Готфрид Вильгельм Лейбниц. Он сделал попытку построить универсальный язык, с помощью которого споры между людьми можно было бы разрешать посредством вычислений. На заложенном Лейбницем фундаменте ирландский математик Джордж Буль построил здание новой науки – математической логики (Алгебры логики), которая в отличие от обычной алгебры оперирует не числами, а высказываниями. В честь Д. Буля логические переменные в языке программирования Паскаль в последствии назвали булевскими.

**Высказывание**- это любое утверждение, относительно которого можно сказать истинно оно или ложно, т.е. соответствует оно действительности или нет. Таким образом, по своей сути высказывания фактически являются двоичными объектами и поэтому часто истинному значению высказывания ставят в соответствие 1, а ложному - 0. Например, запись  $A=1$  означает, что высказывание  $A$  - истинно.

Высказывания могут быть простыми и сложными. Простые -соответствуют алгебраическим переменным, а сложные являются аналогом алгебраических функций. Функции могут получаться путем объединения перемещенных с помощью логических действий.

Самой простой логической операцией является операция «НЕ» (по другому её часто называют отрицанием, дополнением или инверсией и обозначают NOT X). Результат отрицания всегда противоположен значению аргумента.

Логическая операция «НЕ» является унарной, т.е. имеет всего один операнд. В отличие от «НЕ», операции «И» (AND) и «ИЛИ» (OR) являются бинарными, т.к. представляют собой результат действий над двумя логическими величинами.

Логическое «И» ещё часто называют конъюнкцией, или логическим умножением, а «ИЛИ» – дизъюнкцией, или логическим сложением. Часто в ВТ используют дополнительную логическую операцию “исключающее ИЛИ”(X XOR Y), которая отличается от «ИЛИ» только при  $X=1, Y=1$ .

X	NOT X
0	1
1	0

X	Y	X and Y	X or Y	X xor Y
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Операция «И» имеет результат “истина” только в том случае, если оба операнда истинны.

Операция «ИЛИ» “менее привередлива” к исходным данным. Она дает “истину” если значение “истина” имеет хотя бы один из операндов. В случае, когда справедливы оба аргумента одновременно, результат также истинный. В таблицах истинности указаны все возможные комбинации логических переменных X и Y, а также соответствующие им результаты операций. Таблица истинности может рассматриваться в качестве одного из способов задания логической функции.

Операции «И», «ИЛИ», «НЕ» образуют полную систему логических операций, из которых можно построить сколь угодно сложное логическое выражение.

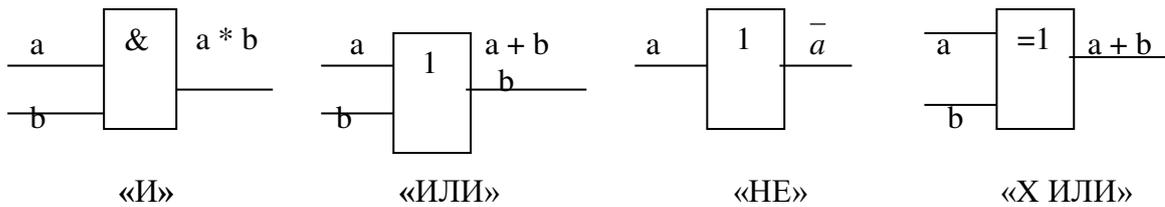
Обработка информации в ЭВМ происходит путём последовательного выполнения элементарных операций. К ним относятся: установка - запись в операционный элемент (например, регистр) двоичного кода: прием- передача (перезапись) кода из одного элемента в другой; сдвиг- изменение положения кода относительно исходного; преобразование- перекодирование; сложение- арифметическое сложение или целых двоичных чисел- некоторые другие. Для выполнения каждой из этих операций сконструированы электронные узлы, являющиеся основными узлами цифровых вычислительных машин- регистры, счетчики, сумматоры, преобразователи кодов и т.д.

В основе каждой из элементарных операций лежит некоторая последовательность логических действий. Например, операция сложения двух чисел: 3+6. Имеем:

$$\begin{array}{r} 011 \\ +110 \\ \hline 1011 \end{array}$$

На каждом элементарном шаге двум двоичным цифрам сопоставляется двоичное число (одно или двухзначное) по правилам: (0;0)  $\Rightarrow$  0, (0;1)  $\Rightarrow$  1, (1;0)  $\Rightarrow$  1, (1;1)  $\Rightarrow$  10. Таким образом, сложение цифр можно описать логической бинарной функцией. Если дополнить это логическим правилом переноса единицы в старший разряд, то сложение полностью сведется к цепочке логических операций.

Рассмотрим условные обозначения базовых логических элементов.



Простейшие логические элементы можно реализовать аппаратно. Это означает, что можно создать электронные устройства на транзисторах, резисторах и т.п, каждая из которых имеет один или два входа для подачи управляющих напряжений и один выход, напряжение на котором определяется таблицей истинности. На практике логическому "да" ("истина", или цифра 1) соответствует наличие напряжения, логическому "нет" ("ложь", или цифра 0) – его отсутствие.

В качестве примера применения логических элементов в ВТ возьмем устройство, называемое сумматором. Его назначение - нахождение суммы двух двоичных чисел. Рассмотрим простейшее устройство, являющееся звеном сумматора – полусумматор - реализующий сложение двух одноразрядных двоичных чисел, которые обозначим А и В. В результате получается двухразрядное двоичное число. Его младшую цифру обозначаем S, а старшую, которая при сложении многоразрядных чисел будет перенесена в старший разряд, через Co (от английского Carry out- "выходной перенос").

Обе цифры можно получить по следующим логическим формулам:

$$S = (\bar{A} * B) + (A * \bar{B}); Co = A * B.$$

Черта над символом обозначает операцию NOT.

Это легко проверить перебором всех четырех возможных случаев сочетания значений А и В, по таблице.

Таблица истинности для полусумматора.

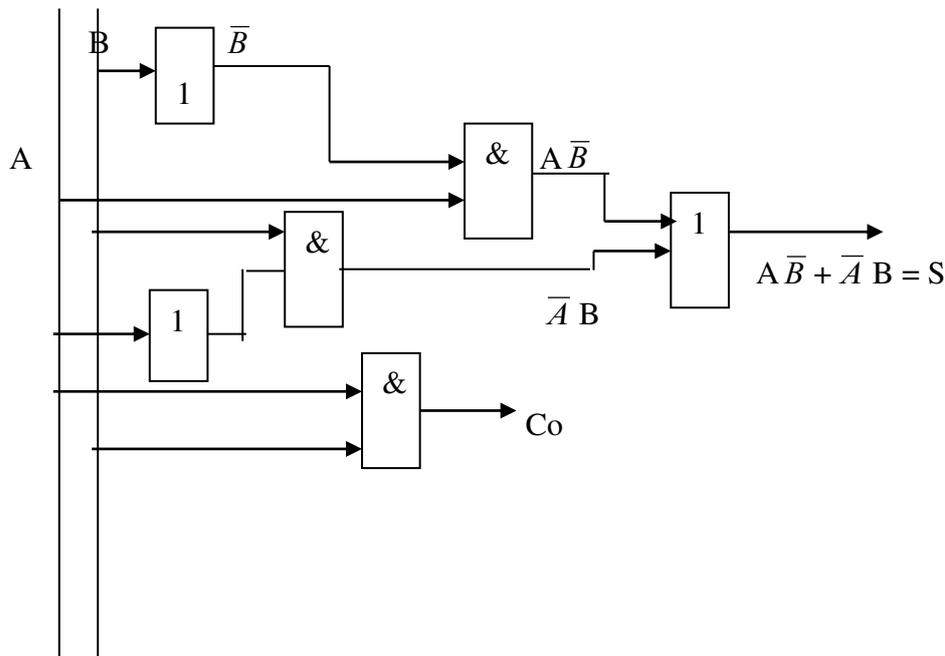
A	B	S	Co
---	---	---	----

0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

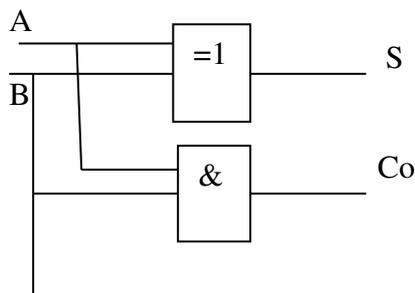
Мысленно объединим столбцы А, В и С<sub>0</sub>. Полученная таблица напоминает базовый элемент «И».

Аналогично сравнив первые три столбца А, В и S с таблицей истинности для рассмотренных логических элементов, обнаружением подходящей для наших целей элемент «исключающее ИЛИ». Таким образом, для реализации полусумматора достаточно соединить параллельно входы двух логических элементов.

### Логическая схема полусумматора.



А) с использованием лишь базовых логических элементов.



Б) с использованием «исключающего ИЛИ».

При построении функциональных узлов КС используются элементы, которые реализуют базовую систему логических функций. Одним из таких базовых наборов является набор из трех функций: дизъюнкции (логическое ИЛИ), конъюнкции (логическое И) и отрицание (логическое НЕ). На рис. 2.1.2. показаны условные обозначения и значения выходного сигнала в зависимости от входных сигналов. Ноль изображается на диаграммах низким значением сигнала, а единица – высоким. Используя эти базовые элементы, строятся все функциональные узлы ЦВМ.

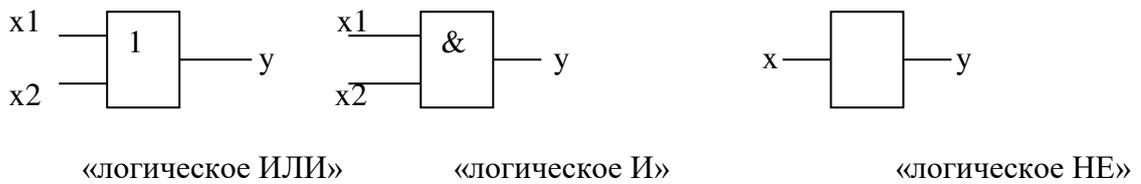


Рис. 2.1.2. Базовая система логических элементов цифровых устройств

### 2.1.3. Функциональные узлы компьютерной системы

#### 2.1.3.1. Элемент памяти

Основой любого компьютера является ячейка памяти, которая может хранить данные или команды. Основой любой ячейки памяти является функциональное устройство, триггер (или защелка), которое может по команде принять или выдать один двоичный бит, а, главное, сохранять его. Триггер строится на основе базового набора логических схем (рис. 2.1.3.).

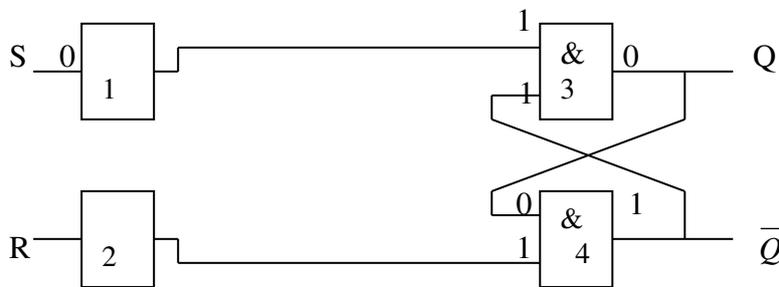


Рис. 2.1.3. Схема триггера в состоянии хранения бита информации.

1 и 2 – это два элемента «логическое НЕ», 3 и 4 – два элемента «логическое И-НЕ», которые представляют собой комбинацию логических элементов «И» и «НЕ». Такой элемент на входе выполняет операцию логического умножения, результат которой инвертируется на выходе логическим отрицанием. Триггер имеет два выхода  $Q$  и  $\bar{Q}$ . Сигнал на выходе  $Q$  соответствует значению, хранящемуся в триггере. Выход  $\bar{Q}$  используется для получения инверсного значения сигнала. Входы  $S$  и  $R$  предназначены для записи в триггер одного бита со значением 0 или 1.

Для записи в триггер 1 на вход  $S$  подается 1 (рис.2.1.4.). На выходе схемы 1 получится 0, который обеспечит на выходе схемы 3 единицу. С выхода схемы 3 единица поступит на вход схемы 4, на выходе которой значение изменится на ноль ( $1 \wedge 1 = 0$ ). Этот ноль на входе схемы 3 будет поддерживать сигнал на выходе в состоянии единицы. Теперь можно снять единичный сигнал на входе  $S$ , на выходе схемы 3 все равно будет высокий уровень, т.е. триггер сохраняет записанную в него 1. Единичный сигнал на входе 3 необходимо удерживать некоторое время, пока на выходе схемы 4 не появится нулевой сигнал. Затем на входе  $S$  вновь устанавливается нулевой сигнал, но триггер поддерживает единичный сигнал на выходе  $Q$ , т.е. сохраняет записанную в него единицу. Точно также, подав единичный сигнал на вход  $R$ , можно записать в триггер ноль.

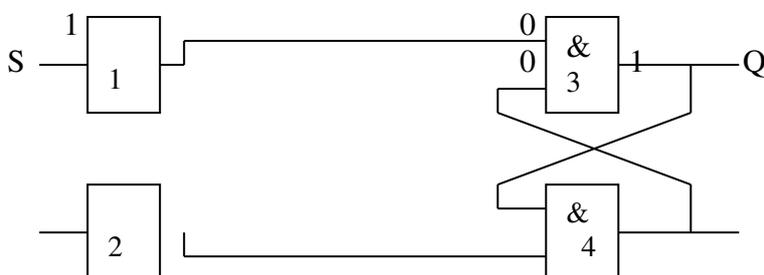




Рис. 2.1.4. Запись в триггер единицы.

### 2.1.3.2. Регистры

Триггер служит основой для построения функциональных узлов, способных хранить двоичные числа, осуществлять их синхронную параллельную передачу и запись, а также выполнять с ними некоторые специальные операции. Такие функциональные узлы называются регистрами.

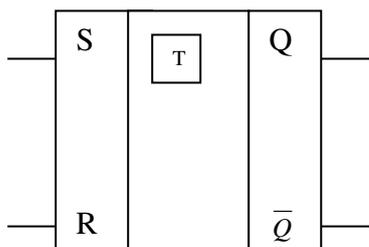


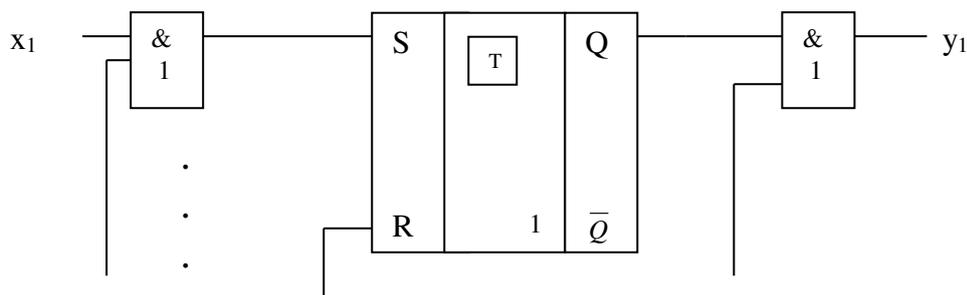
Рис. 2.1.5. Условное обозначение триггера.

Регистр – это набор триггеров, число которых определяет разрядность регистра. Разрядность регистра кратна восьми битам: 8-, 16-, 32-, 64- разрядные регистры. Кроме этого, в состав регистра входят схемы управления его работой.

Регистр содержит  $n$  триггеров, образующих  $n$  разрядов (рис. 2.1.6.). Перед записью информации регистр обнуляется подачей единичного сигнала на вход «сброс». Запись информации в регистр производится синхронно подачей единичного сигнала «Запись». Этот сигнал открывает входные вентили (схемы «логическое И»), и на тех входах  $x_1, \dots, x_n$ , где присутствует единичный сигнал, произойдет запись единицы. Чтение информации из регистра также производится синхронно, подачей сигнала «Чтение» на выходные вентили. Обычно регистры содержат дополнительные схемы, позволяющие организовать такие операции, как сдвиг информации (регистр сдвига) и подсчет поступающих единичных сигналов (регистры счетчики).

### 2.1.4. Понятие архитектуры вычислительной системы

Под архитектурой ЭВМ понимают, прежде всего, основные устройства и блоки ЭВМ, а также структуру связей между ними. С точки зрения архитектуры представляют интерес лишь те связи и принципы, которые являются наиболее общими, присущими многим конкретным реализациям вычислительных систем. Именно, то общее, что есть в строении ЭВМ, и относится к понятию архитектуры. С точки зрения архитектуры важны только те сведения о построении ЭВМ, которые могут как-то использоваться при программировании и «пользовательской» работе с ЭВМ.



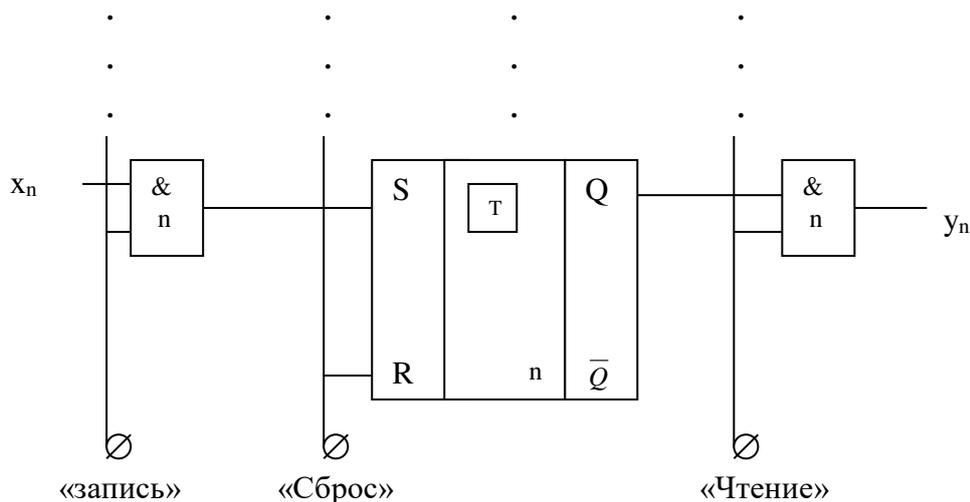


Рис. 2.1.6. Схема n-разрядного регистра хранения с синхронной записью и чтением

Наиболее общие принципы построения ЭВМ, которые относятся к архитектуре:

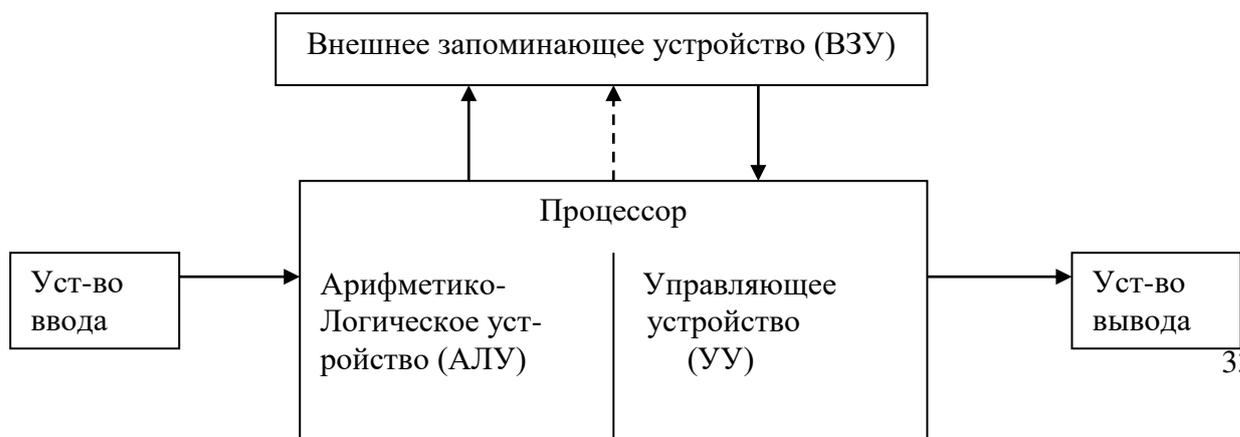
- Структура памяти ЭВМ;
- Способы доступа к памяти и внешним устройствам;
- Возможность изменения конфигурации компьютера;
- Система команд;
- Форматы данных;
- Организация интерфейса.

Обобщая вышесказанное, можно сказать: «Архитектура ЭВМ – наиболее общие принципы построения ЭВМ, реализующие управления работой и взаимодействием ее функциональных устройств».

### 2.1.5. Классическая архитектура ЭВМ и принципы фон Неймана

Основы учения об архитектуре вычислительных машин заложил великий американский ученый Джон фон Нейман. Он подключился к созданию первой ламповой ЭВМ ENIAC, когда ее конструкция была уже выбрана. Фон Нейман высказал идею принципиально новой ЭВМ. Вместе со своими коллегами Г. Голдстейном и А. Берксом фон Нейман обосновал использование двоичной системы для представления чисел (раньше все вычислительные машины хранили обрабатываемые числа в десятичном виде). Они продемонстрировали преимущества двоичной системы для технической реализации, удобство и простоту выполнения в ней арифметических и логических операций.

Еще одной, поистине революционной идеей, является предложенный фон Нейманом принцип «хранимой программы». Он первый догадался, что программа может также храниться в виде набора нулей и единиц в той же памяти, что и обрабатываемые числа. Отсутствие принципиальной разницы между программой и данными дало возможность ЭВМ самой формировать для себя программу в соответствии с результатами вычислений.



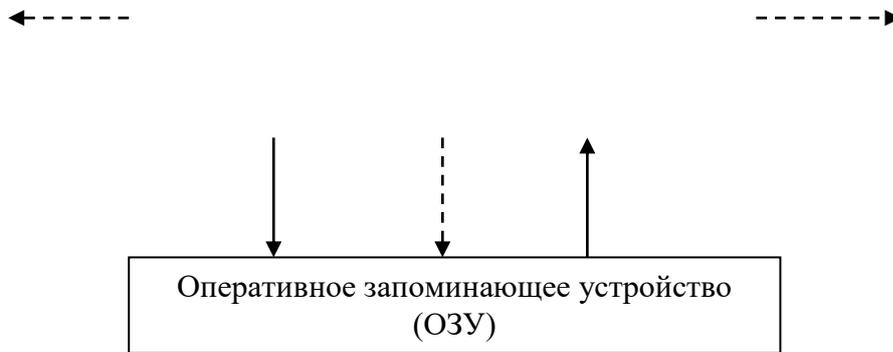


Рис.2.1.7. Архитектура ЭВМ, построенной на принципах фон Неймана. Сплошные линии со стрелками указывают направление потоков информации, пунктирные – управляющих сигналов от процессора к остальным устройствам.

Фон Нейман, кроме основополагающих принципов логического устройства ЭВМ, предложил также ее структуру, которая воспроизводилась в течение первых двух поколений ЭВМ. Основными блоками по Нейману являются устройство управления (УУ) и арифметико-логическое устройство (АЛУ) (в последствии объединенные в центральный процессор), память, внешняя память, устройства ввода и вывода (рис. 2.1.7).

В данной схеме ЭВМ происходит последовательное считывание команд из памяти и их выполнение. Номер (адрес) очередной ячейки памяти, из которой будет извлечена следующая команда программы, указывается специальным устройством – счетчиком команд в УУ. Его наличие является одним из характерных признаков рассмотренной архитектуры.

Разработанные фон Нейманом основы архитектуры вычислительных устройств оказались настолько фундаментальными, что получили название «фон-неймановской архитектуры».

## 2.1.6 Архитектуры ВС сосредоточенной обработки информации

Современный компьютер состоит из нескольких функциональных узлов: процессор, память, контроллеры устройств и т.д. Каждый узел представляет собой сложное электронное устройство, в состав которого могут входить миллионы логических элементов.

Уровни представление компьютера:

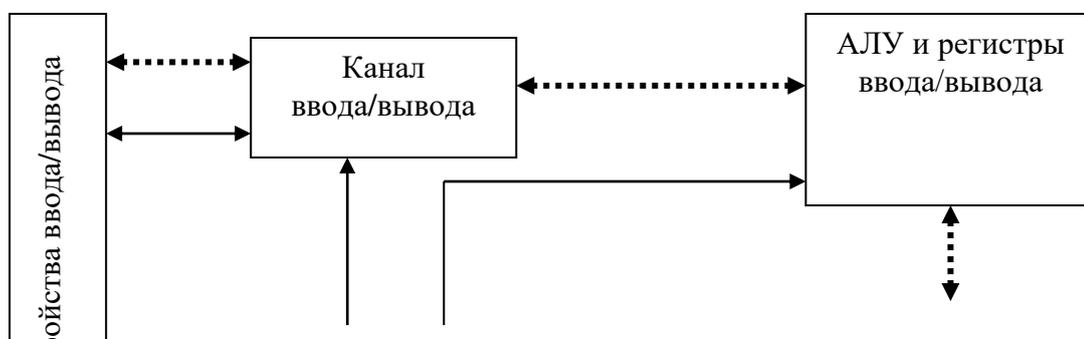
*Цифровой логический уровень* – уровень логических схем базовой системы элементов;

*Микроархитектурный уровень* – уровень организации обработки информации внутри функционального узла. Сюда относятся регистры различного назначения, устройство обработки поступающих команд, устройство преобразования данных, устройство управления;

*Командный уровень* - набор функциональных узлов и связи между ними, система команд и данных, передаваемых между устройствами.

### 2.1.6.1. Архитектуры с фиксированным набором устройств

Компьютерами с сосредоточенной обработкой называются вычислительные машины, у которых одно или несколько обрабатывающих устройств (процессоров) расположены компактно и используют для обмена информацией внутренние шины передачи данных. Компьютеры первых двух поколений имели архитектуру закрытого типа с ограниченным набором внешних устройств.



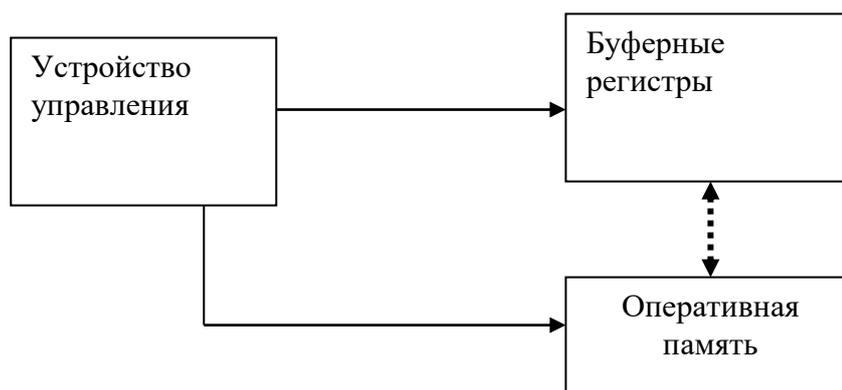


Рис. 2.1.8. Архитектура компьютера закрытого типа

Такая архитектура характерна для компьютеров, базовая система логических элементов которых построена на электронных лампах и транзисторах. Введение любого дополнительного, функционального блока был сопряжен с увеличением потребляемой мощности, занимаемой площади и резко увеличивал стоимость всей системы.

Компьютер, выполненный по этой архитектуре (рис. 2.1.8.), не имел возможности подключения дополнительных устройств, не предусмотренных разработчиком. Оперативная память хранит команды и данные исполняемых программ, АЛУ обеспечивает не только числовую обработку, но и участвует в процессе ввода-вывода информации, осуществляя ее занесение в оперативную память. Канал ввода/вывода представляет собой специализированное устройство, работающее по командам, подаваемым устройством управления. Канал допускает подключение определенного числа внешних устройств. Устройство управления обеспечивает выполнение команд программы и управляет всеми узлами системы.

Компьютеры такой архитектуры эффективны при решении чисто вычислительных задач. Они плохо приспособлены для реализации компьютерных технологий, требующих подключения дополнительных внешних устройств и высокой скорости обмена с ними информации.

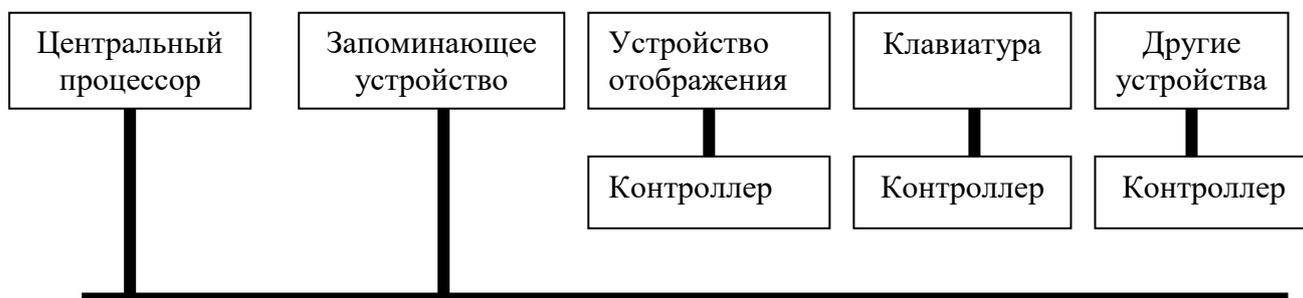
### 2.1.6.2 Вычислительные системы с открытой архитектурой

В начале 70-х г.г. фирмой DEC был предложен компьютер с архитектурой, позволяющей свободно подключать любые периферийные устройства. Главным нововведением являлось подключение всех устройств, независимо от их назначения, к *общей шине* передачи информации. Подключение осуществлялось в соответствии со *стандартом шины*.

Архитектура компьютера открытого типа, основанная на использовании общей шины, приведена на рис. 2.1.9.

Общее управление всей системой осуществляет центральный процессор. Он управляет общей шиной, выделяя время другим устройствам для обмена информацией. Запоминающее устройство хранит исполняемые программы и данные. Внешние устройства подключаются к шине через специальное устройство – контроллер. Контроллер согласовывает сигналы устройства с сигналами шины и осуществляет управление устройством по командам, поступающим от центрального процессора. Контроллер подключается к шине специальными устройствами – портами ввода-вывода. Каждый порт имеет свой номер, и обращение к нему процессора происходит, так же как к ячейке памяти, по этому номеру. Процессор имеет специальные линии управления, сигнал на которых определяет, обращается ли процессор к ячейке памяти или к порту ввода-вывода контроллера внешнего устройства. Несмотря на преимущества архитектуры с общей

шиной, она имеет серьезный недостаток. К общей шине подключаются устройства с разными объемами и скоростью обмена, в связи с чем «медленные» устройства задерживали работу «быстрых». Дальнейшее повышение производительности компьютера было найдено во введении, дополнительной, локальной шины, к которой подключались «быстрые» устройства. Архитектура с общей и локальной шинами приведена на рис. 2.1.10.



Общая шина

Рис. 2.1.9. Архитектура компьютера открытого типа

Контроллер шины анализирует адреса портов, передаваемые процессором, и передает их контроллеру, подключенному к общей или локальной шине.

Конструктивно контроллер каждого устройства размещается на общей плате с центральным процессором и запоминающим устройством или, если устройство не является

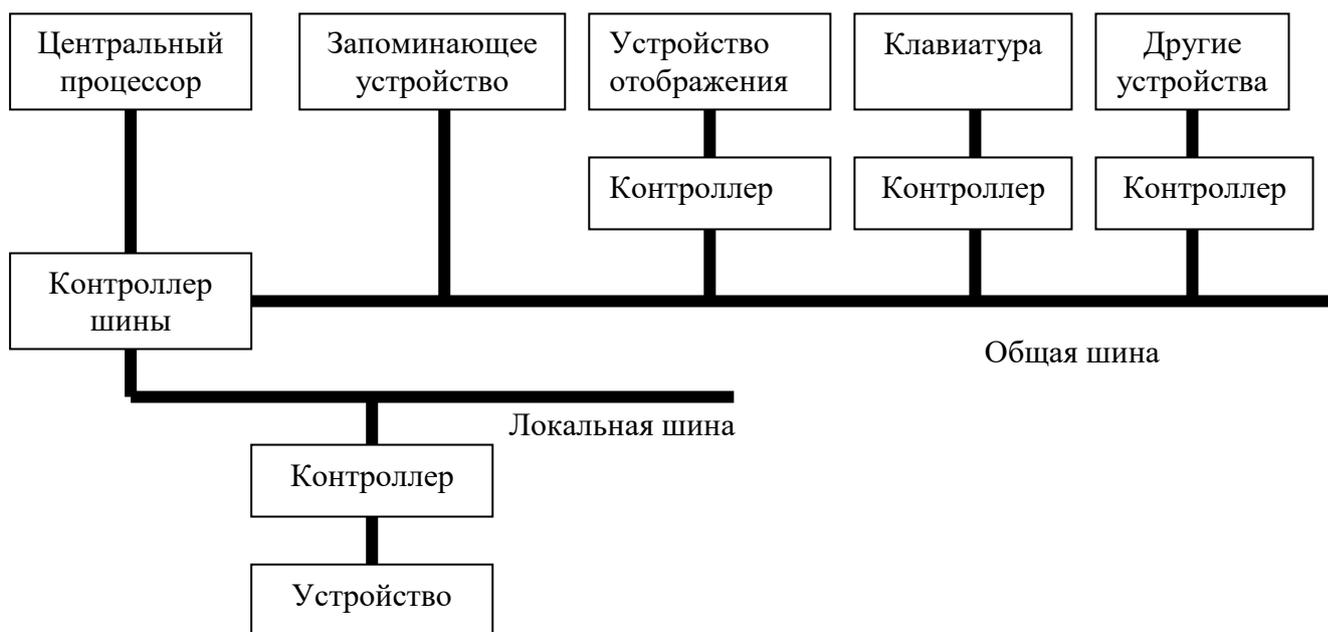
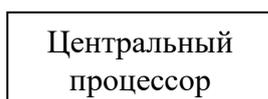


Рис. 2.1.10. Архитектура с общей и локальной шиной.

стандартно входящим в состав компьютера, на специальной плате, вставляемой в специальные разъемы на общей плате – слоты расширения. Дальнейшее развитие микроэлектроники позволило размещать несколько функциональных узлов и контроллеры стандартных устройств в одной микросхеме СБИС. Это сократило количество микросхем на общей плате и дало возможность ввести две, дополнительные, локальные шины для подключения запоминающего устройства и устройства отображения, которые имеют наибольший объем обмена с центральным процессором и между собой. Хотя архитектура компьютера осталась прежней, структура современного ПК имеет вид, представленный на рис. 2.1.11.



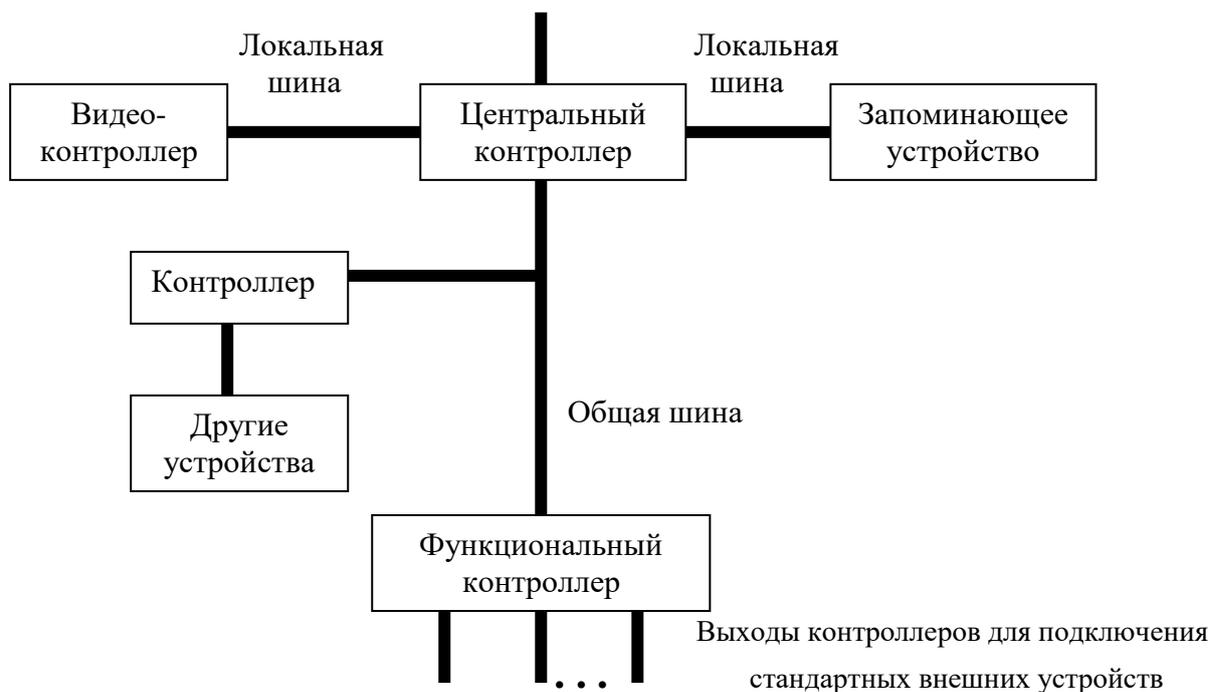


Рис. 2.1.11. Структура персонального компьютера

*Центральный контроллер* играет роль коммутатора, распределяющего потоки информации между *процессором, памятью, устройством отображения* и остальными узлами компьютера. Кроме этого в состав микросхемы центрального контроллера включены устройства, которые поддерживают работу компьютера. К ним относятся: системный таймер; устройство прямого доступа к памяти, которое обеспечивает обмен данными между внешними устройствами и памятью в периоды, когда это не требуется процессору; устройство обработки прерываний, которое обеспечивает быструю реакцию процессора на запросы внешних устройств, имеющих данные для передачи.

Функциональный контроллер – это СБИС, которая содержит контроллеры для подключения стандартных внешних устройств, таких как клавиатура, мышь, принтер, модем и т.д. Часто в состав этого контроллера входит аудио-карта, позволяющая получить на внешних динамиках высококачественный звук при прослушивании музыкальных и речевых файлов.

Для подключения специфических устройства часть общей шины, соединяющая центральный и функциональный контроллеры, имеет слоты расширения для установки плат контроллеров.

## **Лекция 2.2. Состав и назначение основных элементов персонального компьютера**

### **2.2.1. Центральный процессор**

*Центральный процессор (ЦП)* – функционально-законченное программно-управляемое устройство обработки информации, выполненное на одной или нескольких СБИС. В современных ПК разных фирм применяются процессоры двух основных архитектур:

- Полная система команд переменной длины – Complex Instruction Set Computer (CISC);
- Сокращенный набор команд фиксированной длины – Reduced Instruction Set Computer (RISC).

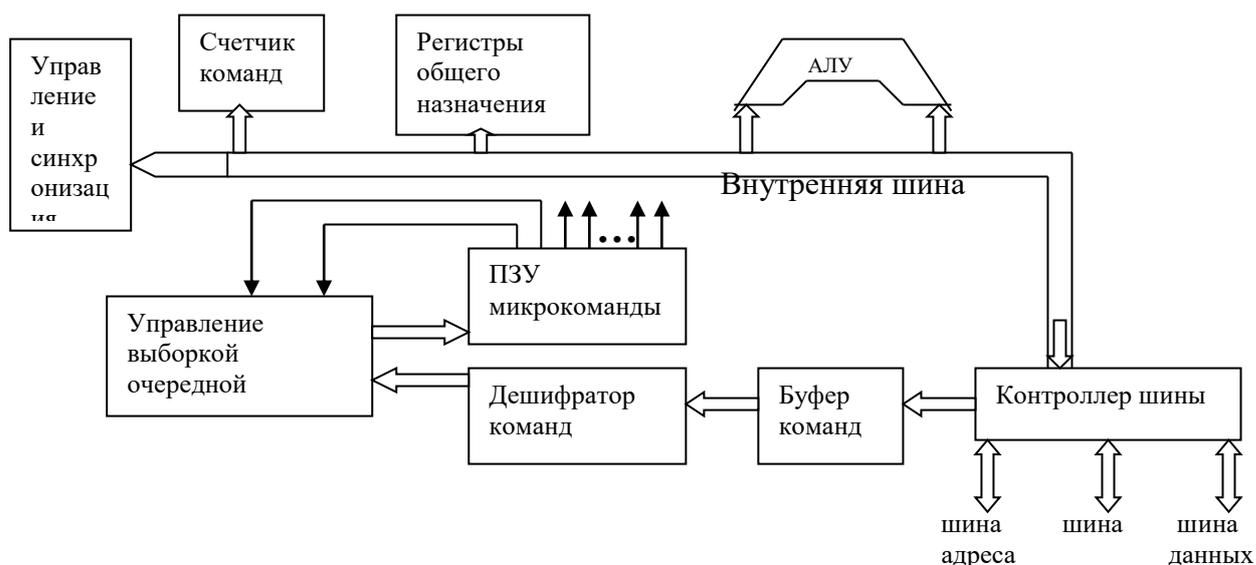
Весь ряд процессоров фирмы Intel, устанавливаемых в ПК IBM имеют архитектуру CISC, а процессоры Motorola, используемые фирмой Apple для своих ПК, имеют архитектуру RISC. Обе архитектуры имеют свои достоинства и недостатки. CISC -

процессоры имеют обширный набор команд (до 400), из которых программист может выбрать, наиболее подходящую в данном случае, команду. Недостатком является то, что большой набор команд усложняет внутреннее устройство управления процессором, увеличивает время исполнения команды на микропрограммном уровне. Команды имеют различную длину и время исполнения.

RISC – архитектура имеет ограниченный набор команд и каждая команда выполняется за один такт работы процессора. Небольшое число команд упрощает устройство управления процессором. К недостаткам можно отнести то, что если требуемой команды в наборе нет, программист вынужден реализовать ее с помощью нескольких команд из имеющегося набора, увеличивая размер программного кода.

Упрощенная схема процессора, отражающая основные особенности архитектуры микро - уровня, приведена на рис. 2.2.1. Наиболее сложным функциональным устройством процессора является устройство управления выполнением команд. Оно содержит:

- *Буфер команд*, который хранит одну или несколько очередных команд программы; читает следующие команды из запоминающего устройства, пока выполняется очередная команда, уменьшая время ее выборки из памяти;
- *Дешифратор команд* расшифровывает код операции очередной команды и преобразует его в адрес начала микропрограммы, которая реализует исполнение команды;



управления

Рис. 2.2.1. Архитектура процессора

- *Управление выборкой очередной микрокоманды* представляет собой небольшой процессор, работающий по принципу фон Неймана, имеет свой счетчик микрокоманд, который автоматически выбирает очередную микрокоманду из ПЗУ микрокоманд;
- *Постоянное запоминающее устройство (ПЗУ) микрокоманд* – это запоминающее устройство, в которое информация записывается однократно и затем может только считываться; отличительной особенностью ПЗУ является то, что записанная в него информация сохраняется сколь угодно долго и не требует постоянного питающего напряжения.

Выборка очередной микрокоманды осуществляется через определенный интервал времени, который зависит от времени выполнения предыдущей микрокоманды. Частота, с которой осуществляется выборка микрокоманд, называется тактовой частотой процессора. Тактовая частота является важной характеристикой процессора, так как определяет скорость выполнения процессором команд, и, в конечном итоге, быстродействие процессора.

Арифметико-логическое устройство (АЛУ) предназначено для выполнения арифметических и логических операций преобразования информации. Функционально АЛУ состоит из нескольких специальных регистров, полно разрядного сумматора и схем местного управления. В основе АЛУ лежит устройство, реализующее арифметическую операцию. Сложения двух целых чисел. Остальные арифметические операции реализуются с помощью представления чисел в специальном дополнительном коде. Сумматор АЛУ – это много разрядное устройство, каждый разряд которого представляет собой схему на логических элементах, выполняющих суммирование двух одноразрядных двоичных чисел с учетом переноса из предыдущего младшего разряда (полусумматор). Результатом является сумма входных величин и перенос в следующий старший разряд. Такое функциональное устройство называется одноразрядным, полным сумматором.

Регистры общего назначения (РОН) используются для временного хранения операндов исполняемой команды и результатов вычислений, а также хранят адреса ячеек памяти или портов ввода-вывода для команд, обращающихся к памяти и внешним устройствам. Если операнды хранятся в РОН, то время выполнения команды значительно сокращается.

Процессоры для современных ПК типа IBM PC выпускают многие фирмы, но наиболее распространенными являются процессоры фирмы Intel. Ее последняя разработка - процессор Pentium 4 (выпуск начат в 2001 г.). К основным особенностям архитектуры Pentium 4 относится:

- В системы команд добавлены новые команды, ориентированные на работу с видео- и аудио - потоками;
- Имеется специальный внутренний КЭШ, размером 256 Кбайт, который работает на тактовой частоте процессора, и имеет собственную шину связи с процессором, обеспечивающую скорость обмена 48 Гбайт/с;
- Внутренняя микро архитектура процессора базируется на двух параллельно работающих конвейерах команд (суперскалярная архитектура), которые исполняют сразу несколько команд в разных фазах обработки (чтение, дешифрация, загрузка операндов, исполнение), конвейеры заканчиваются двумя АЛУ, работающими на удвоенной частоте процессора для коротких арифметических и логических команд, и АЛУ для выполнения медленных команд;
- Процессор работает на частотах 1500 – 3000 МГц и содержит около 42 млн. транзисторов.

Фирма Intel поставляет упрощенные варианты процессор Pentium 4 под названием Celeron, который в два раза дешевле базового варианта. Последние модели Celeron ни в чем не уступают «старшему брату» и даже в некоторых случаях превосходят его.

### **2.2.2. Внутренние запоминающие устройства: оперативное запоминающее устройство, кэш-память, постоянное запоминающее устройство**

Другим важным функциональным узлом компьютера является запоминающее устройство, или память. Память, в которой хранятся исполняемые программы и данные, называется оперативным запоминающим устройством (ОЗУ), или RAM (Random Access Memory) - памятью со свободным доступом. ОЗУ позволяет записывать и считывать информацию из ячейки, обращаясь к ней по ее номеру или адресу. Ячейка памяти имеет стандартное число двоичных разрядов – один байт. Информация в ОЗУ сохраняется все время, пока на схемы памяти подается питание, т.е. она является энергозависимой.

Существует два вида ОЗУ, отличающиеся техническими характеристиками: динамическое ОЗУ, или DRAM (Dynamic RAM), и статическое ОЗУ, или SRAM (Static RAM). Быстродействие динамического ОЗУ на порядок ниже, чем статического. Обычно, в качестве оперативной или видеопамяти используется динамическое ОЗУ. Статическое ОЗУ

используется в качестве небольшой буферной сверхбыстродействующей памяти. В кэш – память из динамической памяти заносятся команды и данные, которые процессор будет выполнять в данный момент.

Скорость работы ОЗУ ниже, чем быстродействие процессора, поэтому применяются различные методы для повышения ее производительности. Например, размещение в одном корпусе микросхемы СБИС нескольких модулей памяти с чередованием адресов.

Для уменьшения влияния времени обращения процессора к ОЗУ и увеличения производительности компьютера дополнительно устанавливается сверхбыстродействующая буферная память, выполненная на микросхемах статической памяти. Эта память называется кэш-памятью (от англ. Cache – запас). Время обращения к данным в кэш-памяти на порядок ниже, чем ОЗУ, и сравнимо со скоростью работы самого процессора. Современные процессоры имеют встроенную кэш-память, которая находится внутри процессора, кроме этого есть кэш-память и на системной плате. Чтобы их различать, она делится на уровни. На кристалле самого процессора находится кэш-память первого уровня, она имеет объем порядка 16-128 Кбайт и самую высокую скорость обмена данными. В корпусе процессора, но на отдельном кристалле находится кэш-память второго уровня, которая имеет объем порядка 256 Кбайт – 2 Мбайта. Кэш-память третьего уровня расположена на системной плате, ее объем может составлять 16 – 1000 Мбайт.

Использование процессом кэш-памяти увеличивает производительность процессора, особенно в тех случаях, когда происходит последовательное преобразование относительно небольшого числа данных, которые постоянно во время преобразования хранятся в кэш-памяти.

В одном адресном пространстве с ОЗУ находится специальная память, предназначенная для постоянного хранения таких программ, как тестирование и начальная загрузка компьютера, управление внешними устройствами. Она является энергонезависимой, т.е. сохраняет записанную информацию при отсутствии питания. Такая память называется постоянным запоминающим устройством (ПЗУ) или ROM (Read Only Memory). Постоянные запоминающие устройство можно разделить по способу записи с них информации на следующие категории:

ПЗУ, программируемые однократно. Программируются при изготовлении и не позволяют изменять записанную в них информацию.

Перепрограммируемые ПЗУ (ППЗУ). Позволяют перепрограммировать их многократно. Изменение содержания ППЗУ можно выполнять как непосредственно в составе вычислительной системы (такая технология называется *флэш - технологией*), так и вне ее, на специальных устройствах, называемых *программаторами*.

### 2.2.3. Внутренние шины

*Общая шина*, наряду с центральным процессором и запоминающим устройством, во многом определяет производительность работы компьютера, так как обеспечивает обмен информацией между функциональными узлами. Общая шина делится на три отдельные шины по типу передаваемой информации: шина адреса, шина данных, шина управления. Каждая шина характеризуется: *шириной* – числом параллельных проводников для передачи информации; *тактовой частотой* – частотой, на которой работает контроллер шины при формировании циклов передачи информации.

Шина адреса предназначена для передачи адреса ячейки памяти или порта ввода-вывода. Ширина шины адреса определяет максимальное количество ячеек, которое она может напрямую адресовать. Если ширина шины адреса  $n$ , то количество адресуемой памяти равно  $2^n$ .

Шина данных предназначена для передачи команд и данных, и ее ширина во многом определяет информационную пропускную способность общей шины. В современных компьютерах ширина шины данных составляет 32 – 64.

Шина управления включает в себя все линии, которые обеспечивают работу общей шины. Ее ширина зависит от типа шины и определяется алгоритмом ее работы или *протоколом* работы шины. Протокол работы шины состоит из нескольких циклов и выполняется контроллером шины, расположенным внутри процессора (рис. 2.2.1.), или отдельным контроллером шины (рис. 2.1.11.).

Разработчики предлагают включать в состав компьютера дополнительные шины, связывающие напрямую центральный процессор и отдельные наиболее быстродействующие устройства. Такие шины получили название локальных шин. На рис. 2.1.11 локальные шины используются для подключения к процессору запоминающего устройства и видеоконтроллера.

*Основные характеристики общих и локальных шин, применяемых в ПК фирмы IBM.*

*Общая шина PCI* применяется в настольных компьютерах, в настоящее время используется модификация PCI 2/1/ Тактовая частота контроллера этой шины 66 МГц, ширина шины адреса – 32, а шины данных – 64 разряда. Пиковая пропускная способность шины 528 Мбайт/с.

Общая шина PCMCIA применяется в переносных компьютерах класса ноутбук и имеет параметры, сравнимые с параметрами шины PCI/

Локальная шина для подключения видеоконтроллера AGP позволяет организовать непосредственную связь видеоконтроллера и оперативного запоминающего устройства. Она ориентирована на массовую передачу видеоданных. Имеет конвейерную организацию выполнения операций чтения/записи, что позволяет избежать задержек при обращении к модулям памяти. За один такт работы может передать два, четыре или восемь блоков данных, в зависимости от установленного режима работы. При установке режима параллельной передачи восьми блоков обеспечивает пиковую скорость передачи 2112 Мбайт/с.

### **2.2.3. Внешние запоминающие устройства**

В отличие от оперативного запоминающего устройства, внешние запоминающие устройства (ВЗУ) обладают большим объемом сохраняемой информации и являются энергонезависимыми. Наибольшее распространение получили дисковые ВЗУ, которые, в зависимости от типа носителя, можно разделить на магнитные, оптические и смешанные.

**Магнитные диски** используют магнитные материалы со специальными свойствами, позволяющими фиксировать два состояния. Информация на магнитные диски записывается и считывается магнитной головкой, которая перемещается радиально с фиксированным шагом, а сам диск при этом вращается вокруг своей оси. Головка считывает или записывает информацию, расположенную на концентрической окружности, которая называется *дорожкой* или *треком*. Каждая дорожка дополнительно разбивается на ряд участков – секторов. Сектор содержит минимальный блок информации, который может быть записан или считан с диска. Чтение и запись на диск осуществляется блоками, поэтому дисководы называют блочными устройствами.

*Физическая структура диска* определяется количеством дорожек и числом секторов на каждой дорожке. Она задается при форматировании диска, которое выполняется специальными программами и должна быть произведена перед первым использованием диска для записи информации.

Кроме физической структуры диска, говорят еще о *логической структуре*. Логическая структура определяется файловой системой, которая реализована на диске и зависит от операционной системы компьютера, на котором используется данный диск.

### **Организация файловой системы на магнитных носителях.**

Все современные дисковые операционные системы обеспечивают создание файловой системы, предназначенной для хранения данных на дисках и обеспечения доступа к ним. Принцип организации файловой системы – табличный. Поверхность диска рассматривается

как трехмерная матрица, измерениями которой являются номера поверхностей, цилиндра и сектора. Под цилиндром понимается совокупность всех дорожек, принадлежащих разным поверхностям и находящихся на равном удалении от оси вращения. Данные о том, в каком месте диска записан тот или иной файл, хранятся в системной области диска в специальных таблицах размещения файлов (FAT-таблицах). Нарушение FAT-таблицы приводит к невозможности воспользоваться данными, поэтому она существует в двух экземплярах, идентичность которых регулярно контролируется средствами операционной системы.

Наименьшей физической единицей хранения данных является сектор. Размер сектора равен 512 байт. Размер FAT-таблицы ограничен, поэтому для дисков, размер которых превышает 32 Мбайт, обеспечить адресацию к каждому отдельному сектору не представляется возможным. В связи с этим группы секторов условно объединяются в кластеры. Кластер является наименьшей единицей адресации к данным. Размер кластера не фиксируется и зависит от емкости диска.

Операционные системы MS DOS, OS/2, Windows 95 и Windows NT реализуют 16-разрядные поля в таблицах размещения файлов. Такая файловая система называется FAT-16. Она позволяет разместить в FAT-таблицах не более 65536 записей ( $2^{16}$ ) о местоположении единиц хранения данных и для дисков объемом от 1 до 2 Гбайт длина кластера составляет 32 Кбайт (64 сектора). Это не вполне рационально, т.к. любой файл (даже очень маленький) полностью оккупирует весь кластер, которому соответствует только одна адресная запись в FAT-таблице. Даже если файл велик и располагается в нескольких кластерах, все равно в его конце образуется остаток, нерационально расходуемый целый кластер.

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

Современные операционные системы, начиная с Windows 98, обеспечивают более совершенную организацию файловой системы – FAT-32 с 32-разрядными полями в таблицах размещения файлов. Для дисков размера до 8 Гбайт эта система обеспечивает размер кластера 4 Кбайт (8 секторов).

Дисководы магнитных дисков делятся на дисководы для сменных носителей (дискет) и дисководы жестких дисков (винчестеры), которые устанавливаются в системном блоке компьютера. Все сменные носители на дисках характеризуются диаметром или форм-фактором. Наибольшее распространение получили гибкие магнитные диски с форм-фактором 3,5 дюйма.

#### **Накопители на оптических дисках**

Оптический компакт-диск, который был предложен в 1982 г. фирмами Philips и Sony первоначально для записи звуковой информации, идеально подходил для записи цифровой информации больших объемов на сменном носителе. Объем информации, записанной на компакт-диске, составляет 600-700 Мбайт. К достоинствам можно отнести и его относительную дешевизну, высокую надежность и долговечность, нечувствительность к загрязнению и воздействию магнитных полей.

В середине 90-х гг. появились устройства, устанавливаемые непосредственно на компьютере и позволяющие производить однократную запись информации на компакт-диск. Для таких устройств выпускают специальные компакт-диски, которые получили название CD-Recordable (CD-R).

Позднее появились компакт-диски с возможностью перезаписи – CD-ReWritable (CD-RW).

Дальнейшее развитие технологии производства компакт-дисков привело к созданию дисков с высокой плотностью записи – цифровой универсальный диск Digital Versatile Disk (DVD). Объем записываемой информации на диске достигает 4,7 Гбайт. Дальнейшее увеличение объема информации обеспечивается применением двусторонних DVD.

#### **Флэш-память.**

К недостаткам дисковой памяти можно отнести наличие механических движущихся компонентов, имеющих малую надежность, и большую потребляемую мощность при записи и считывании. Появление большого числа цифровых устройств (MP3-плееры, цифровые фото- и видеокамеры, карманные компьютеры) потребовало разработки миниатюрных устройств внешней памяти, обладающих малой энергоемкостью, значительной емкостью и обеспечивали бы совместимость с персональными компьютерами. Первые промышленные образцы такой памяти появились в 1994 г. и получили название флэш-память.

Флэш-память представляет собой микросхему перепрограммируемого постоянного запоминающего устройства (ПЗУ) с неограниченным числом циклов перезаписи. Конструктивно флэш-память выполняется в виде отдельного блока, содержащего микросхему флэш-памяти и контроллер, для подключения к одному из стандартных входов компьютера. Размеры этого блока 40 x 16 x 7 мм. Флэш-память, используемая в других цифровых устройствах, имеет иные размеры и конструктивное оформление. В настоящее время объем флэш-памяти достигает нескольких Гбайт, скорость записи и считывания составляют десятки Мбайт/с.

## 2.2.4. Устройства ввода/вывода данных

### Видеотерминалы

**Видеотерминалы** предназначены для оперативного отображения текстовой и графической информации в целях визуального восприятия ее пользователей. Видеотерминал состоит из видеомонитора (дисплея) и видеоконтроллера.

Для ПК используются мониторы следующих типов:

- На основе электроннолучевой трубки (ЭЛТ);
- На основе жидкокристаллических индикаторов (ЖКИ, LCD – Liquid Crystal Display);
- Плазменные мониторы (PDP – Plasma Display Panels);
- Электролюминесцентные мониторы (FED – Field Emission Display);
- Самоизлучающие мониторы (LEP – Light Emission Plastics).

Основные характеристики мониторов:

*Размер экрана монитора* – задается величиной диагонали в дюймах. Домашние ПК оснащаются мониторами с размерами 15 или 17 дюймов, а для профессиональной работы, требующей отображения мелких деталей, используются мониторы с размерами 21 и 22 дюйма.

*Разрешающая способность* – определяется числом *пикселей* (световых точек) по горизонтали и вертикали. Стандартные значения разрешающей способности современных мониторов следующие: 800 x 600, 1024 x 768, 1800 x 1440, 2048 x 1536 и др. Значение разрешающей способности определяет качество изображения на экране.

*Рабочая частота кадровой развертки* – определяет скорость смены кадров изображения. Она влияет на утомляемость глаз при продолжительной работе на компьютере. Чем выше частота кадровой развертки, тем меньше утомляемость глаз. Частота смены кадров во многом зависит от разрешающей способности экрана: чем выше разрешающая способность, тем меньше частота смены кадров. Например, при разрешении 800 x 600 максимальная частота смены кадров может составить 120 Гц, а при разрешении 1600 x 1200 – 67 Гц.

На разрешающую способность монитора и качество изображения влияет объем *видеопамяти*. Современные видеоконтроллеры для хранения цвета каждого пикселя расходуют 4 байт памяти, для чего необходимо иметь объем видеопамяти от 32 до 128 Мбайт. Большой объем видеопамяти позволяет устанавливать более высокий режим разрешения и большее число цветов для каждого пикселя.

*Мониторы на основе ЭЛТ* используют электроннолучевые трубки, применяемые в обычных телевизионных приемниках, и устройство, формирующее на экране точки

(пиксели). Для цветных мониторов цвет точки создается смещением тех основных цветов (красного, зеленого и синего) и зависит от интенсивности каждого электронного луча. Цветной монитор может отображать до 16 млн. оттенков в каждой точке.

Мониторы на жидкокристаллических индикаторах представляют собой плоские панели. Эти мониторы используют специальную, прозрачную жидкость, которая при определенных напряжениях электростатического поля кристаллизуется, при этом изменяется ее прозрачность и коэффициент преломления световых лучей. Эти эффекты используются для формирования изображения. Эти мониторы имеют лучшую яркость и предоставляют возможность смотреть на экран даже с отклонением до 45° от вертикали.

В *плазменных мониторах* изображение формируется светом, выделяемым при газовом разряде в каждом пикселе экрана. Большими преимуществами таких мониторов являются высокая яркость и контрастность, отсутствие дрожания изображения, а также большой угол отклонения от нормали, при котором изображение сохраняет высокое качество. К недостаткам можно отнести недостаточную пока разрешающую способность и достаточно быстрое (пять лет при офисном использовании) ухудшение качества люминофора. Пока такие мониторы используют только для конференций и презентаций.

*Электролюминесцентные мониторы* состоят из двух пластин, с ортогонально нанесенными на них прозрачными проводниками. На одну из пластин нанесен слой люминофора, который начинает светиться при подаче напряжения на проводники в точке их пересечения, образуя пиксель.

Самоизлучающие мониторы используют матрицу пикселей, построенную на основе полупроводникового материала, излучающего свет при подаче на него напряжения (светодиод). На сегодняшний день имеются монохромные самоизлучающие дисплеи с желтым свечением, но они уступают по сроку службы LCD мониторам. Достоинства таких мониторов заключаются в том, что они обеспечивают 180-градусный обзор, работают при низком напряжении питания и имеют малый вес.

### **Устройства ввода информации**

**Клавиатура.** Клавиатурой называется устройство для ручного ввода информации в компьютер. Современные типы клавиатур различаются, в основном, принципом формирования сигнала при нажатии клавиши.

Среди современных типов клавиатур можно отметить беспроводную, в которой передача информации в компьютер происходит с помощью датчика инфракрасного излучения, аналогично пультам управления различной бытовой техники. Такая клавиатура позволяет работать в любом удобном для пользователя месте, не привязываясь к расположению системного блока. Можно также отметить гибкую резиновую клавиатуру, которая работает бесшумно, защищена от механических и химических разрушающих воздействий, очень тонкая и может быть свернута в виде цилиндра.

Клавиатурный процессор, который обрабатывает сигналы от клавиатуры, определяет номер клавиши, которая была нажата, так называемый скан-код, а сервисные программы операционной системы определяют, какой именно символ или команда были введены. Такой подход позволяет сопоставлять каждой клавише больше одного символа.

Манипулятор типа «мышь». В качестве дополнительных устройств для ручного ввода информации наиболее широко используются устройства графического ввода типа «мышь» и устройства для ввода информации в игровые программы – джойстики.

Манипулятор «мышь» представляет необходимое средство для работы с компьютером. Мышь представляет собой электронно-механическое устройство, внешний вид которой и принцип действия весьма разнообразны. Наиболее популярные типы мыши, применяемые в настольных компьютерах, имеют вид небольшой коробочки, сверху которой находятся две кнопки управления командами мыши и колесико скроллинга, применяемого для прокрутки информации в некоторых приложениях. На нижней части находится механическое или электронное устройство, отслеживающее перемещение мыши по поверхности. В портативных компьютерах мышь вмонтирована в его корпус и представляет

собой площадку с сенсорами, которые отслеживают движение пальца по площадке и силу его давления и перемещают курсор по экрану или, при более сильном нажатии, выполняют команду. Такие устройства получили название трекпоинты или трекпады. Выпускаются мыши, передающие информацию в компьютер по инфракрасному каналу. У таких мышей отсутствует «хвостик», связывающий ее с компьютером, из-за которого она и получила свое имя.

**Джойстик.** Манипулятор типа является основным устройством для управления многочисленными компьютерными играми. Простейший джойстик представляет собой основание с укрепленной на нем рукояткой, на которой размещены четыре кнопки и двухпозиционный курок. Функции всех кнопок и положения рукоятки программируются и для разных игр могут иметь разные действия. Джойстики обеспечивают больший контроль над игрой и значительно полнее передают реальную игровую ситуацию. Для подключения джойстика используется стандартный вход, размещаемый обычно на разъеме звуковой карты, или другой стандартный вход компьютера.

### **Устройства печати**

Существует несколько типов устройств, обеспечивающих получение твердой копии электронного документа на бумаге или другой материале. Наиболее распространены два типа таких устройств: принтеры и плоттеры.

**Печатающие устройства (принтеры)** предназначены для вывода информации на бумагу. Все принтеры могут выводить текстовую информацию, многие из них могут выводить также рисунки и графики, а некоторые принтеры могут выводить и цветные изображения. Существует несколько тысяч моделей принтеров, которые могут использоваться с IBM PC. Как правило, применяются принтеры следующих типов: матричные, струйные и лазерные, однако встречаются и другие (светодиодные, термопринтеры и т.д.).

Основной характеристикой принтера, определяющей качество бумажного документа, является разрешающая способность, измеряющаяся числом элементарных точек, которые помещаются на одном дюйме. Чем выше разрешающая способность, тем точнее воспроизводятся детали изображения. Современные принтеры обеспечивают разрешение от 200 до 2880 dpi.

Еще одной важной характеристикой является производительность принтера, которая измеряется количеством страниц, изготавливаемых принтером в минуту. Обычно производительность указывается для страниц формата А4.

*Матричные* (или точно-матричные) принтеры - наиболее распространенный, до недавнего времени тип принтеров для IBM PC. Принцип печати этих принтеров таков: печатающая головка принтера содержит вертикальный ряд тонких металлических стержней (иглолок). Головка движется вдоль печатаемой строки, а стержни в нужный момент ударяют по бумаге через красящую ленту. Это и обеспечивает формирование на бумаге символов и изображений. В дешевых моделях принтеров используется печатающая головка с 9 стержнями. Качество печати у таких принтеров посредственное, но его можно несколько улучшить с помощью печати в несколько проходов (от двух до четырех). Более качественная и быстрая печать обеспечивается принтерами с 24 печатающими иглолками (24-точечными принтерами). Бывают принтеры и с 48 иглолками, они обеспечивают еще более качественную печать. Скорость печати точно-матричных принтеров - от 60 до 10 с на страницу, печать рисунков может выполняться медленнее - до 5 мин на страницу. Разрешающая способность составляет 200 – 300 dpi. Достоинством является низкая стоимость расходных материалов (красящей ленты) и возможность печати одновременно нескольких копий документа. К недостаткам относится низкая скорость и шум при печати.

*Струйные принтеры.* В этих принтерах изображение формируется микрокаплями специальных чернил, выдуваемых на бумагу с помощью сопел. Число сопел составляет от 12 до 64. Струйные принтеры выполняют и цветную печать. Качество цветной печати получает очень высокое и практически неотличимо от типографского.

Достоинства: высокое разрешение, которое зависит от числа сопел в головке и составляет от 300 до 1200 dpi; высокая скорость печати (до 10 ppm); бесшумность работы. Основной недостаток – высокая стоимость расходных материалов и возможность засыхания красителя в сопле, что приведет к замене всего печатающего блока.

*Лазерные принтеры* обеспечивают в настоящее время наилучшее качество печати. В этих принтерах для печати используется принцип ксерографии: изображение переносится на бумагу со специального барабана, к которому электрически притягиваются частички краски. Отличие от обычного ксерокопировального аппарата состоит в том, что печатающий барабан электризуется с помощью лазера по командам из компьютера.

К достоинствам можно отнести высокую скорость печати лазерных принтеров - от 15 до 5 с на страницу ( до 40 ppm) при выводе текстов (страницы с рисунками могут выводиться значительно дольше), а также низкую себестоимость копии и бесшумность работы.

**Плоттеры.** Плоттеры, или графопостроители, используются, в основном, для вывода графической информации – чертежи, схемы диаграммы и т.п. Основное достоинство заключается в том, что они предназначены для получения изображения на бумаге большого формата, например, А1.

Плоттеры делятся два больших класса: векторные и растровые. В векторных плоттерах пишущий узел перемещается относительно бумаги сразу по вертикали и горизонтали, вычерчивая на бумаге непрерывные кривые в любом направлении. В растровых – пишущий узел перемещается относительно бумаги только в одном направлении. Изображение формируется строка за строкой из последовательности точек.

Векторные плоттеры используют для рисования узел, который называется пером. В качестве пера, используются перья с чернилами, фибровые и пластиковые стержни (фломастеры), карандашные грифели и мелки или шариковые узлы однократного и многократного действия. Перьевые плоттеры обеспечивают высокое качество изображений, но имеют невысокую скорость работы. Постепенно перьевые и шариковые узлы плоттеров вытесняются струйными узлами, аналогичными узлам струйных принтеров.

Растровые плоттеры могут иметь струйный или лазерный пишущий узел. Их основное отличие от принтеров с подобным принципом работы в ширине обрабатываемого изображения.

### **Сканеры.**

Сканеры являются наиболее распространенными устройства для решения задачи перевода бумажных документов в электронные копии. Их можно классифицировать по целому ряду признаков. Прежде всего, сканеры бывают черно-белые и цветные.

Важной характеристикой сканеров является разрешающая способность, измеряющаяся количеством различаемых точек на дюйм изображения, и составляет от 75 до 1600 dpi. Для нормального распознавания образов, в частности, текстов, вполне достаточно 300-600 dpi. Разрешение необходимо выбирать индивидуально для каждого конкретного использования сканируемого изображения.

Увеличение разрешения резко увеличивает размер файла.

Конструктивно сканеры делятся на три типа: ручные, планшетные и роликовые. Ручные сканеры имеют малые габариты и низкую стоимость и позволяют сканировать изображения любого размера, но могут возникать искажения при совмещении частей изображения.

Планшетные сканеры – наиболее распространенный тип. В них сканирующая головка движется относительно неподвижного оригинала, который помещается на прозрачное стеклянное основание. Достоинство таких сканеров заключается в том, что с их помощью можно сканировать и листовые и сброшюрованные документы (книги). Скорость сканирования составляет 2 – 10 с на страницу формата А4. К недостаткам можно отнести необходимость ручного позиционирования каждой страницы оригинала.

Роликовые сканеры используются для пакетной обработки листовых документов. В них подача очередного листа для сканирования происходит автоматически. Сканирующая головка в таких сканерах неподвижна, а лист оригинала перемещается относительно нее. К

недостаткам можно отнести проблему выравнивания листов и сложность работы с листами нестандартного размера.

## **Программные средства реализации информационных процессов**

### ***Лекция 3.1. Классификация программного обеспечения***

*Программы — это упорядоченные последовательности команд.* Конечная цель любой компьютерной программы — управление аппаратными средствами. Даже если на первый взгляд программа никак не взаимодействует с оборудованием, не требует никакого ввода данных с помощью устройств ввода и не осуществляет вывод данных на устройства вывода, все равно ее работа основана на управлении аппаратными устройствами компьютера.

Программное и аппаратное обеспечение в компьютере работают в неразрывной связи и в непрерывном взаимодействии.

Состав программного обеспечения вычислительной системы называют *программной конфигурацией*. Между программами, как и между физическими узлами и блоками существует взаимосвязь — многие программы работают, опираясь на другие программы более низкого уровня, то есть, мы можем говорить о межпрограммном *интерфейсе*. Возможность существования такого интерфейса тоже основана на существовании технических условий и протоколов взаимодействия, а на практике он обеспечивается распределением программного обеспечения на несколько взаимодействующих между собой уровней. Уровни программного обеспечения представляют собой пирамидальную конструкцию. Каждый следующий уровень опирается на программное обеспечение предшествующих уровней. Такое членение удобно для всех этапов работы с вычислительной системой, начиная с установки программ до практической эксплуатации и технического обслуживания. Обратите внимание на то, что каждый вышележащий уровень повышает функциональность всей системы. Так, например, вычислительная система с программным обеспечением базового уровня не способна выполнять большинство функций, но позволяет установить системное программное обеспечение.

#### **3.1.1. Системное программное обеспечение компьютера**

Системный уровень ПО — переходный. Программы, работающие на этом уровне, обеспечивают взаимодействие прочих программ компьютерной системы с программами базового уровня и непосредственно с аппаратным обеспечением, то есть выполняют “посреднические” функции.

От программного обеспечения этого уровня во многом зависят эксплуатационные показатели всей вычислительной системы в целом. Так, например, при подключении к вычислительной системе нового оборудования на системном уровне должна быть установлена программа, обеспечивающая для других программ взаимосвязь с этим оборудованием. Конкретные программы, отвечающие за взаимодействие с конкретными устройствами, называются *драйверами устройств* — они входят в состав программного обеспечения системного уровня.

Другой класс программ системного уровня отвечает за взаимодействие с пользователем. Именно благодаря им он получает возможность вводить данные в вычислительную систему, управлять ее работой и получать результат в удобной для себя форме. Эти программные средства называют *средствами обеспечения пользовательского интерфейса*. От них напрямую зависит удобство работы с компьютером и производительность труда на рабочем месте.

Совокупность программного обеспечения системного уровня образует *ядро операционной системы компьютера*. Если компьютер оснащен программным обеспечением системного уровня, то он уже подготовлен к установке программ более высоких уровней, к

взаимодействию программных средств с оборудованием и, самое главное, к взаимодействию с пользователем. То есть *наличие ядра операционной системы — неперемнное условие для возможности практической работы человека с вычислительной системой.*

### 3.1.2. Операционная система. Назначение и основные функции ОС.

Особое место среди программных средств всех типов занимают ОС, являясь ядром программного обеспечения.

*ОС* - комплекс программ, обеспечивающих:

- управление ресурсами, т.е. согласованную работу всех аппаратных средств компьютеров;
- управление процессами, т.е. выполнение программ, их взаимодействие с устройствами компьютера, с данными;
- пользовательский интерес, т.е. диалог пользователя с компьютером, выполнение определенных простых команд-операций по обработке информации.

ОС наиболее машинно-зависимый вид программного обеспечения, ориентированный на конкретные модели компьютеров, поскольку они напрямую управляют их устройствами, т.е. обеспечивают интерфейс между пользователем и аппаратной частью компьютера. ОС образует автономную среду, не связанную ни с одним языком программирования. Любая же прикладная программа связана с ОС и может эксплуатироваться только на тех компьютерах, где имеется аналогичная системная среда. Прикладные программные средства, разработанные в среде одной ОС, не могут быть использованы для работы в среде другой ОС, если нет конвертора, позволяющего сделать это. В таком случае говорят о программной несовместимости компьютеров.

Обычно ОС хранится на жестком диске. При включении компьютера ОС автоматически загружается с диска в оперативную память.

### 3.1.3. Эволюция ОС

ОС за все время существования компьютеров претерпели значительную эволюцию.

1. Первые операционные системы были *однопользовательскими и однозначными*. Они характеризуются невысокой эффективностью использования ресурсов компьютера из-за простоев всех, кроме одного работающего периферийного устройства. Например, при вводе данных простаивал центральный процессор, устройства вывода и внешние запоминающие устройства.
2. *Однопользовательские и многозадачные* - обеспечивают постановку заданий в очередь на выполнение, параллельное выполнение заданий, разделение ресурсов компьютера между выполняющимися заданиями. Например, одно задание может выполнять ввод данных, другое – выполняться центральным процессором, третье – выводить данные, четвертое - стоять в очереди. Такие возможности обеспечивают контроллеры внешних устройств (собственные процессоры).

При многозадачном режиме:

- в оперативной памяти находится несколько заданий пользователей;
- время работы процессора разделяется между программами, находящимися в ОП и готовыми к обслуживанию процессором;
- параллельно с работой процессора происходит обмен информацией с различными внешними устройствами.

3. Наиболее совершенны и сложны *многопользовательские многозадачные* ОС. Они предусматривают одновременное выполнение многих заданий многих пользователей, обеспечивают разделение ресурсов компьютера в соответствии с приоритетами пользователей и защиту данных каждого пользователя от несанкционированного доступа. В этом случае ОС работает в режиме разделенного времени, т.е. обслуживает многих пользователей, работающих каждый со своего терминала.

В настоящее время используется много типов различных ОС для ЭВМ различных видов. В их структуре существуют общие принципы. В составе многих ОС можно выделить ядро – некоторую часть, являющуюся основой всей системы. В состав ядра входят наиболее часто используемые модули: модуль управления системой прерываний, средства по распределению основных ресурсов – ОП и процессор. Программы, входящие в состав ядра при загрузке ОС помещаются в ОЗУ, где постоянно находятся и используются при функционировании ЭВМ (такие программы являются резидентными). Важной частью ОС является *командный процессор* – программа, отвечающая за интерпретацию и использование простейших команд пользователя, и его взаимодействие с ядром ОС. К ОС следует относить и богатый набор *утилит* – небольших программ, обслуживающих различные устройства компьютера (форматирования магнитных дисков, восстановление удаленных файлов и т.д.).

Наиболее популярные ОС MS DOS и Windows фирмы Microsoft, UNIX.

### 3.1.4. Пользовательские интерфейсы

По реализации интерфейса пользователя различают *неграфические* и *графические* операционные системы. Неграфические операционные системы реализуют *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. Для компьютеров платформы IBMPC интерфейс командной строки обеспечивается семейством операционных систем под общим названием MS-DOS (версии от MS-DOS 1.0 до MS-DOS 6.2).

Графические операционные системы реализуют более сложный тип интерфейса, в котором в качестве органа управления кроме клавиатуры может использоваться мышь или адекватное устройство позиционирования. Работа с графической операционной системой основана на взаимодействии активных и пассивных экранных элементов управления.

В качестве активного элемента управления выступает *указатель мыши* – графический объект, перемещение которого на экране синхронизировано с перемещением мыши.

В качестве пассивных элементов управления выступают графические элементы управления приложений (экранные кнопки, значки, переключатели, флажки, раскрывающиеся списки, строки меню и многое другое).

Характер взаимодействия между активными и пассивными элементами управления выбирает сам пользователь. В его распоряжении приемы наведения указателя мыши на элемент управления, щелчки кнопками мыши и другие средства.

## Лекция 3.2. ОС Windows

### 3.2.1. 32 – разрядная архитектура

Первая 32-разрядная ОС для компьютеров IBM PC - Windows 95. Она появилась в результате слияния ОС MS-DOS и ее графической оболочки Windows 3.1. Операционная система Windows ориентирована на работу 32-разрядных программ, многие ее компоненты являются 32-разрядными. Разрядность – это максимальное количество разрядов двоичного числа, над которым одновременно может выполняться машинная операция, в т.ч. и операция передачи информации; чем больше разрядность, тем, при прочих равных условиях, будет больше и производительность ПК.

Большинство программ для ОС MS DOS относилось к 16-разрядным программам, которые использовали реальный режим работы микропроцессора. Реальный режим значительно ограничивает возможности программы, т.к. в этом режиме затруднен доступ в верхние (свыше 1 Мбайта) области памяти. MS-DOS не имеет средств для поддержки 32-разрядных программ, работающих в защищенном режиме микропроцессора. Чтобы такие программы могли работать в среде MS-DOS, требуется дополнительное программное

обеспечение, расширяющее функции MS-DOS. Возможен так же вариант, когда дополнительные функции, обеспечивающие защищенный режим, включаются непосредственно в код программы, увеличивая тем самым ее объем.

ОС Windows полностью обеспечивает работу 32-разрядных программ, причем она спроектирована таким образом, что использование 32-разрядных программ в ее среде является наиболее оптимальным. В среде Windows также успешно функционируют 16-разрядные программы, но они не могут задействовать все ресурсы системы.

32-разрядные программы занимают больше оперативной и дисковой памяти, чем 16-разрядные. Но это компенсируется, во-первых, увеличением скорости работы программы, во-вторых, удешевлением всех видов памяти, в т.ч. и электронной.

### **3.2.2. Многозадачность и многопоточность**

ОС Windows является многозадачной (мультизадачной), т.е. она способна «одновременно» выполнять несколько программ. На самом деле один МП может выполнять инструкции только одной программы. Однако ОС настолько оперативно реагирует на потребности той или иной программы, что создается впечатление одновременности их работы. Например, в процессе подготовки текстового документа можно параллельно печатать содержимое какого-либо файла и проверять на вирус жесткий диск.

Многозадачность может быть кооперативной и вытесняющей. При кооперативной многозадачности ОС не занимается решением проблемы распределения процессорного времени. Распределяют его сами программы. Причем активная программа (т.е. та, с которой работают в настоящий момент) самостоятельно решает, отдавать ли процессор другой программе. Момент передачи управления здесь зависит от хода выполнения задачи. Таким моментом должен быть системный вызов, т.е. обращение к системе за какой-либо услугой (ввод или вывод на внешнее устройство и пр.). Фоновым задачам выделяется процессорное время при простое приоритетной задачи (ожидание нажатия клавиши и др.).

Кооперативная многозадачность была реализована в среде Windows 3.1. В Windows кооперативная многозадачность обеспечивается для 16-разрядных приложений, т.к. эти приложения, созданные для Windows 3.1., умеют самостоятельно распределять процессорное время.

При вытесняющей многозадачности распределением процессорного времени между программами занимается ОС. Она выделяет каждой задаче фиксированный квант времени процессора. По истечении этого кванта времени система вновь получает управление, чтобы выбрать другую задачу для ее активизации. Если задача обращается к ОС до истечения ее кванта времени, то это также служит причиной переключения задач. Такой режим многозадачности Windows реализует для 32-разрядных приложений, а также для программ, написанных для MS-DOS.

Многопоточность операционной системы означает, что работающие программы (процессы) могут разделяться на несколько частей. При этом каждая часть самостоятельно претендует на процессорное время. Это обеспечивает одновременное выполнение программой нескольких не связанных друг с другом операций. Например, в ТП вычисления в разных ячейках благодаря многопоточности могут выполняться одновременно, причем параллельно с вычислениями могут быть организованы ввод данных в ячейки, их вывод на печатающее устройство и т.п.

### **3.2.3. Графический пользовательский интерфейс**

Пользовательский интерфейс Windows использует графический режим видеомонитора. Основу нового графического интерфейса пользователя составляет хорошо продуманная система окон, располагающаяся на экране монитора и включающая множество разнородных графических объектов для управления работой компьютера. Тем самым реализуется идея создания электронного рабочего стола пользователя, на котором

размещаются электронные документы. Принятая концепция графического пользовательского интерфейса характерна для всех программных продуктов под Windows . Это обеспечивает комфортную среду работы пользователя (более подробно этот вопрос мы рассмотрим позже).

### **3.2.4. Подключение новых периферийных устройств по технологии Plug and Play**

Для подключения нового периферийного устройства в среде ОС MS-DOS пользователь должен обладать профессиональными знаниями: например, уметь написать файл конфигурации, знать структуру команды подключения необходимого драйвера.

В среде Windows эта задача решается достаточно просто. Система самостоятельно создает и изменяет файлы конфигурации, распознает конкретное техническое устройство и производит его автонастройку. Подобная технология получила название Plug and Play – «включай и работай».

Дополнительное устройство подключается специальными программами – мастерами, которые могут распознать любое устройство, поддерживающие технологию Plug and Play. Если устройство не поддерживает эту технологию, программы – мастера запрашивают дополнительную информацию в наиболее удобной для пользователя форме, благодаря чему процесс подключения устройств упрощается.

### **3.2.5. Использование виртуальной памяти**

Извечная компьютерная проблема – нехватка оперативной памяти – решается в среде Windows с помощью виртуальной (реально не существующей) памяти.

Виртуальная память – расширение адресного пространства задачи, полученное за счет использования части внешней памяти.

В оперативной памяти всегда находится часть виртуального пространства, выделяемого для решения задачи, остальная его часть располагается на дисковой памяти. Если оперативной памяти не хватает для обеспечения работы текущего (активного) приложения, то приложение или его часть, которые не используют в данный момент микропроцессор, выгружаются (вытесняются) из оперативной памяти на диск. На их место в оперативную память загружается (подкачивается) необходимый фрагмент активного приложения. Когда одному из выгруженных приложений передается управление, оно вновь загружается в оперативную память, что может привести к выгрузке на диск другого, пассивного в данный момент приложения. Т.о., программы циркулируют между диском и оперативной памятью.

Поддержка виртуальной памяти позволяет открыть большое количество приложений одновременно, но выгрузка на диск и загрузка с диска снижают производительность компьютера.

Используемая для этой цели часть внешней памяти называется файлом подкачки, а описанный процесс подкачки – свопингом. Объем файла подкачки может в несколько раз превышать объем оперативной памяти.

Файл подкачки – файл на жестком диске, используемый для организации виртуальной памяти.

Настройка виртуальной памяти производится автоматически и вручную. Вручную выполнять настройку может только квалифицированный пользователь. Доступ к настройке виртуальной памяти осуществляется через Панель управления, после открытия значка Система.

По умолчанию задан автоматический вариант настройки, т.е. Windows самостоятельно выбирает размер виртуальной памяти, в зависимости от реальной потребности текущей задачи.

### **3.2.6. Совместимость с ранее созданным программным обеспечением**

Под совместимостью с ПО понимают способность ОС исполнять программные продукты, созданные в др. ОС. В большинстве случаев ОС Windows обеспечивает такую

совместимость на IBM – подобных компьютерах не только для программ ранних версий, но и для программ операционной системы MS-DOS.

### **3.2.7. Наличие коммуникационных программных средств**

Важнейшим направлением развития Windows является включение в ее структуру широко используемых и специальных программных средств для поддержки различных коммуникаций и компьютерных сетей.

Сетевые средства ОС Windows позволяют:

- Обеспечить передачу данных между двумя соединенными кабелями компьютерами с помощью программы Direct Cable Connection;
- Организовать электронную почту в локальной и глобальной сети с помощью программы Microsoft Exchange;
- Выполнить факсимильную передачу с помощью программы Microsoft Fax;
- Обмениваться файлами с удаленным компьютером и подключиться к глобальной сети с помощью программы Hyper Terminal;
- Организовать одноранговую компьютерную сеть из нескольких компьютеров, причем компьютеры сети могут управляться как Windows 95, так и Windows 3.1.;
- Подключиться к компьютерной сети с выделенным сервером.

### **3.2.8. Наличие средств мультимедиа**

ОС Windows обеспечивает интерактивную работу с высококачественным звуком и видео при помощи специальных аппаратных и программных средств. Мультимедиа – компьютер непременно должен иметь звуковую плату, которая обеспечивает преобразование звука в компьютерную форму и обратно, и видео-плату, которая преобразует видеoinформацию в компьютерную и обратно. К звуковой плате подключаются различные акустические системы.

Для работы со звуком и видео на компьютере разработано множество программных средств. Они обеспечивают проигрывание аудио – компакт – дисков, воспроизведение, запись и редактирование звуковых файлов, просмотр видео- клипов.

### **3.2.9. Интеграция с глобальной сетью Интернет**

В число программ, реализующих интеграцию с Интернет, входят: программа Подключения к Интернет, программа электронной почты Outlook Express, программа организации телеконференций и телефонных разговоров по цифровым каналам NetMeeting, редактор Web-страниц Front Page Express.

### **3.2.10. Повышение надежности и качества управления**

Повышение надежности обеспечивают как новые, так и модернизированные старые служебные программы, которые выполняют: проверку системных файлов, очистку и проверку диска, автоматическое техническое обслуживание, проверку реестра, архивацию данных и др.

Появилась возможность поддержки нескольких мониторов для расширения рабочего стола, выполнение разных программ на разных мониторах, а также возможность выполнения многоэкранных программ или игр.

Работа с Windows облегчается за счет таких средств управления, как запуск одним щелчком, выделение значков, кнопки «Вперед» и «Назад» и удобного для настройки Главного меню.

Компьютер, оснащенный операционной системой Windows, может работать круглосуточно, выполняя многие функции самостоятельно:

- Почистить жесткие диски и удалить с них накопившийся за день «мусор»;
- Проверить состояние жестких дисков и устранить иеющиеся дефекты файловой структуры;
- Выполнить резервное копирование важнейшей информации;

Проверить состояние сжатого тома, провести его переупаковку, повысить степень сжатия редко используемых данных и программ;

Обслужить жесткие диски и провести дефрагментацию таким образом, чтобы файлы, последовательно загружающиеся при запуске программ, располагались на диске в порядке их вызова и др.

### **Лекция 3.3. Обслуживание файловой структуры**

#### **3.3.1. Функции ОС по обслуживанию файловой структуры**

Несмотря на то, что данные о местоположении файлов хранятся в табличной структуре, пользователю они представляются в виде иерархической структуры. Все необходимые преобразования берет на себя операционная система. К функции обслуживания файловой структуры относятся следующие операции, происходящие под управлением операционной системы:

- Создание файлов и присвоение им имен;
- Создание каталогов (папок) и присвоение им имен;
- Переименование файлов и каталогов (папок);
- Копирование и перемещение файлов между дисками компьютера и между каталогами (папками) одного диска;
- Удаление файлов и каталогов;
- Навигация по файловой структуре с целью доступа к заданному файлу, каталогу (папке);
- Управление атрибутами файлов.

#### **3.3.2. Создание и именование файлов**

Файл – это поименованная последовательность байтов произвольной длины. Поскольку из этого определения вытекает, что файл может иметь нулевую длину, то фактически создание файла состоит в присвоении ему имени и регистрации его в файловой системе – это одна из функций ОС. Даже когда мы создаем файл, работая в какой-то прикладной программе, в общем случае для этой операции привлекаются средства ОС.

Имя файла состоит из двух частей: собственно *имени* и *расширения имени*.

На имя отводится 8 символов, а на его расширение – 3 символа. Имя от расширения отделяется точкой. Как имя, так и расширение могут включать только алфавитно-цифровые символы латинского алфавита.

Основным недостатком «коротких» имен является их низкая содержательность. Далеко не всегда удается выразить несколькими символами характеристику файла, поэтому с появлением ОС Windows 95 было введено понятие «длинного» имени. Такое имя может содержать до 256 символов. «Длинное» имя может содержать любые символы, кроме девяти специальных: \ / : \* ? “ < > |. В имени разрешается использовать пробелы и несколько точек. Расширением имени считаются все символы, идущие после последней точки.

Наряду с «длинным» именем ОС Windows создают также и короткое имя файла – оно необходимо для возможности работы с данным файлом на рабочих местах с устаревшими операционными системами.

#### **3.3.3. Особенности Windows**

Использование «длинных» имен файлов в операционной системе Windows имеет ряд особенностей.

3. Если «длинное» имя файла включает пробелы, то в служебных операциях его надо заключать в кавычки. Рекомендуется не использовать пробелы, а заменять их символами подчеркивания.

4. В корневой папке диска (на верхнем уровне иерархической файловой структуры) нежелательно хранить файлы с длинными именами – в отличие от прочих папок в ней ограничено количество единиц хранения, причем, чем длиннее имена, тем меньше файлов можно разместить в корневой папке.
5. Кроме ограничения на длину имени файла (256 символов) существует гораздо более жесткое ограничение на длину полного имени файла (в него входит путь доступа к файлу, начиная от вершины иерархической структуры). Полное имя не может быть длиннее 260 символов.
6. Разрешается использовать символы любых алфавитов, в том числе и русского.
7. Прописные и строчные буквы не различаются операционной системой. Для нее имена Письмо.txt и письмо.txt соответствуют одному и тому же файлу. Однако, символы разных регистров отображаются ОС и прописные буквы можно использовать.
8. Программисты давно научились использовать расширение имени файла для передачи операционной системе, исполняющей программе или пользователю информации о том, к какому типу относятся данные, содержащиеся в файле, и о формате, в котором они записаны. В ранних ОС этот факт использовался мало. В современных ОС любое расширение имени файла может нести информацию для операционной системы. Системы Windows имеют средства для регистрации свойств типов файлов по расширению их имени, поэтому во многих случаях выбор расширения имени файла не является частным делом пользователя. Приложения этих систем предлагают выбрать только основную часть имени и указать тип файла, а соответствующее расширение имени приписывают автоматически.

#### **Вопросы для самоконтроля**

1. Что такое операционная система?
2. Перечислите основные функции операционной системы.
3. Расскажите о видах интерфейса пользователя в разных операционных системах.
4. Перечислите функции операционной системы по обслуживанию файловой структуры.
5. Объясните правила, по которым формируются короткое имя файла и длинное имя файла.

### ***Лекция 3.4. Служебное программное обеспечение***

#### **3.4.1. Общая характеристика программ служебного уровня**

Программное обеспечение этого уровня взаимодействует как с программами базового уровня, так и с программами системного уровня. Основное назначение служебных программ (их также называют *утилитами*) состоит в автоматизации работ по проверке, наладке и настройке компьютерной системы. Во многих случаях они используются для расширения или улучшения функций системных программ. Некоторые служебные программы (как правило, это программы обслуживания) изначально включаются в состав операционной системы, но большинство служебных программ являются для операционной системы внешними и служат для расширения ее функций.

В разработке и эксплуатации служебных программ существует два альтернативных направления: *интеграция с операционной системой* и *автономное функционирование*. В первом случае служебные программы могут изменять потребительские свойства системных программ, делая их более удобными для практической работы. Во втором случае они слабо связаны с системным программным обеспечением, но предоставляют пользователю больше возможностей для персональной настройки их взаимодействия с аппаратным и программным обеспечением.

### 3.4.2. Классификация служебного программного обеспечения

**Диспетчеры файлов (файловые менеджеры).** С помощью программ данного класса выполняется большинство операций, связанных с обслуживанием файловой структуры: копирование, перемещение и переименование файлов, создание каталогов (папок), удаление файлов и каталогов, поиск файлов и навигация в файловой структуре. Базовые программные средства, предназначенные для этой цели, обычно входят в состав программ системного уровня и устанавливаются вместе с операционной системой (Например, программа «Проводник»). Для повышения удобства работы с компьютером большинство пользователей устанавливают дополнительные служебные программы.

**Средства сжатия данных (архиваторы)-** предназначены для создания архивов. Архивирование данных упрощает их хранение за счет того, что большие группы файлов и каталогов сводятся в один архивный файл. При этом повышается и эффективность использования носителя за счет того, что архивные файлы обычно имеют повышенную плотность записи информации. Архиваторы часто используют для создания резервных копий ценных данных.

**Средства просмотра и воспроизведения.** Обычно для работы с файлами данных необходимо загрузить их в “родительскую” прикладную систему, с помощью которой они были созданы. Это дает возможность просматривать документы и вносить в них изменения. Но в тех случаях, когда требуется только просмотр без редактирования, удобно использовать более простые и более универсальные средства, позволяющие просматривать документы разных типов.

В тех случаях, когда речь идет о звукозаписи или видеозаписи, вместо термина *просмотр* применяют термин воспроизведение документов.

**Средства диагностики.** Предназначены для автоматизации процессов диагностики программного и аппаратного обеспечения. Они выполняют необходимые проверки и выдают собранную информацию в удобном и наглядном виде. Их используют не только для устранения неполадок, но и для оптимизации работы компьютерной системы.

**Средства контроля (мониторинги).** Программные средства контроля иногда называют мониторинг. Они позволяют следить за процессами, происходящими в компьютерной системе. При этом возможны два подхода: наблюдение в реальном режиме времени или контроль с записью результатов в специальном протокольном файле. Первый подход обычно используют при изыскании путей для оптимизации работы вычислительной системы и повышения ее эффективности. Второй подход используют в тех случаях, когда мониторинг выполняется автоматически и (или) дистанционно. В последнем случае результаты мониторинга можно передать удаленной службе технической поддержки для установления причин конфликтов в работе программного и аппаратного обеспечения.

Средства мониторинга, работающие в режиме реального времени, особенно полезны для практического изучения приемов работы с компьютером, поскольку позволяют наглядно отображать те процессы, которые обычно скрыты от глаз пользователя.

**Мониторы установки.** Программы этой категории предназначены для контроля за установкой программного обеспечения. Необходимость в данном программном обеспечении связана с тем, что между различными категориями программного обеспечения могут устанавливаться связи. Вертикальные связи (между уровнями) являются необходимым условием функционирования всех компьютеров. Горизонтальные связи (внутри уровней) характерны для компьютеров, работающих с операционными системами, поддерживающими принцип совместного использования одних и тех же ресурсов разными программными средствами. И в тех и в других случаях при установке или удалении программного обеспечения могут происходить нарушения работоспособности прочих программ.

Мониторы установки следят за состоянием и изменением окружающей программной среды, отслеживают и протоколируют образование новых связей и позволяют восстанавливать связи, утраченные в результате удаления ранее установленных программ.

Простейшие средства управления установкой и удалением программ обычно входят в состав операционной системы и размещаются на системном уровне программного обеспечения, однако они редко бывают достаточны. Поэтому в вычислительных системах, требующих повышенной надежности, используют дополнительные служебные программы.

**Средства коммуникации (коммуникационные программы).** С появлением электронной связи и компьютерных сетей программы этого класса приобрели очень большое значение. Они позволяют устанавливать соединения с удаленными компьютерами, обслуживают передачу сообщений электронной почты, работу с телеконференциями (группами новостей), обеспечивают пересылку факсимильных сообщений и выполняют множество других операций в компьютерных сетях.

**Средства обеспечения компьютерной безопасности.** К этой весьма широкой категории относятся средства пассивной и активной защиты данных от повреждения, а также средства защиты от несанкционированного доступа, просмотра и изменения данных.

В качестве *средств пассивной защиты*, используют служебные программы, предназначенные для резервного копирования. Нередко они обладают и базовыми свойствами диспетчеров архивов (архиваторов). В качестве *средств активной защиты* применяют *антивирусное программное обеспечение*. Для защиты данных от несанкционированного доступа, их просмотра и изменения служат специальные системы, основанные на криптографии.

## **Лекция 3.5. Прикладное программное обеспечение**

### **3.5.1. Классификация прикладного программного обеспечения**

**Текстовые редакторы.** Основные функции этого класса прикладных программ заключаются в вводе и редактировании текстовых данных. Дополнительные функции состоят в автоматизации процессов ввода и редактирования. Для операций ввода, вывода и сохранения данных, текстовые редакторы вызывают и используют системное программное обеспечение. Это характерно и для всех прочих видов прикладных программ.

**Текстовые процессоры.** Основное отличие текстовых процессоров от текстовых редакторов в том, что они позволяют не только вводить и редактировать текст, но и *форматировать* его, то есть оформлять. Соответственно, к основным средствам текстовых процессоров относятся средства обеспечения взаимодействия текста, графики, таблиц и других объектов, составляющих итоговый документ, а к дополнительным — средства автоматизации процесса форматирования.

Современный стиль работы с документами подразумевает два альтернативных подхода — работу с бумажными документами и работу с электронными документами (по безбумажной технологии). Поэтому, говоря о форматировании документов средствами текстовых процессоров, надо иметь в виду два принципиально разных направления - форматирование документов, предназначенных для печати, и форматирование электронных документов, предназначенных для отображения на экране. Приемы и методы в этих случаях существенно различаются. Соответственно, различаются и текстовые процессоры, хотя многие из них успешно сочетают оба подхода.

**Графические редакторы.** Это обширный класс программ, предназначенных для создания и (или) обработки графических изображений. В данном классе различают следующие *категории*: *растровые редакторы*, *векторные редакторы* и программные средства для создания и обработки трехмерной графики (*3D - редакторы*).

*Растровые редакторы* применяют в тех случаях, когда графический объект представлен в виде комбинации точек, образующих растр и обладающих свойствами яркости и цвета. Такой подход эффективен в тех случаях, когда графическое изображение имеет много полутонов и информация о цвете элементов, составляющих объект, важнее, чем информация об их форме. Это характерно для фотографических и полиграфических

изображений. Растровые редакторы широко применяются для обработки изображений, их ретуши, создания фотоэффектов и художественных композиций (коллажей).

Возможности создания новых изображений средствами растровых редакторов ограничены и не всегда удобны. В большинстве случаев художники предпочитают пользоваться традиционными инструментами, после чего вводят рисунок в компьютер с помощью специальных аппаратных средств (*сканеров*) и завершают работу с помощью растрового редактора путем применения спецэффектов.

*Векторные редакторы* отличаются от растровых способом представления данных об изображении. Элементарным объектом векторного изображения является не точка, а линия. Такой подход характерен для чертежно-графических работ, в которых форма линий имеет большее значение, чем информация о цвете отдельных точек, составляющих ее. В векторных редакторах каждая линия рассматривается как математическая кривая третьего порядка и, соответственно, представляется не комбинацией точек, а математической формулой (в компьютере хранятся числовые коэффициенты этой формулы). Такое представление намного компактнее, чем растровое, соответственно данные занимают много меньше места, однако построение любого объекта выполняется не простым отображением точек на экране, а сопровождается непрерывным пересчетом параметров кривой в координаты экранного или печатного изображения. Соответственно, работа с векторной графикой требует более производительных вычислительных систем.

Из элементарных объектов (линий) создаются простейшие геометрические объекты (примитивы) из которых, в свою очередь, составляются законченные композиции. Художественная иллюстрация, выполненная средствами векторной графики, может содержать десятки тысяч простейших объектов, взаимодействующих друг с другом.

Векторные редакторы удобны для создания изображений, но практически не используются для обработки готовых рисунков. Они нашли широкое применение в рекламном бизнесе, их применяют для оформления обложек полиграфических изданий и всюду, где стиль художественной работы близок к чертежному.

*Редакторы трехмерной графики* используют для создания трехмерных композиций. Они имеют две характерные особенности. Во-первых, они позволяют гибко управлять взаимодействием свойств поверхности изображаемых объектов со свойствами источников освещения и, во-вторых, позволяют создавать трехмерную анимацию. Поэтому редакторы трехмерной графики нередко называют также *3D-аниматорами*.

**Системы управления базами данных.** Базами данных называют огромные массивы данных, организованных в табличные структуры. Основными функциями систем управления базами данных являются:

- создание пустой (незаполненной) структуры базы данных;
- предоставление средств ее заполнения или импорта данных из таблиц другой базы;
- обеспечение возможности доступа к данным, а также предоставление средств поиска и фильтрации.

Многие системы управления базами данных дополнительно предоставляют возможности проведения простейшего анализа данных и их обработки. В результате возможно создание новых таблиц баз данных на основе имеющихся. В связи с широким распространением сетевых технологий к современным системам управления базами данных предъявляется также требование возможности работы с удаленными и распределенными ресурсами, находящимися на серверах всемирной компьютерной сети.

**Электронные таблицы.** Электронные таблицы предоставляют комплексные средства для хранения различных типов данных и их обработки. В некоторой степени они аналогичны системам управления базами данных, но основной акцент смещен не на хранение массивов данных и обеспечение к ним доступа, а на преобразование данных, причем в соответствии с их внутренним содержанием.

В отличие от баз данных, которые обычно содержат широкий спектр типов данных (от числовых и текстовых до мультимедийных), для электронных таблиц характерна повышенная сосредоточенность на числовых данных. Зато электронные таблицы предоставляют более широкий спектр методов для работы с данными числового типа.

Основное свойство электронных таблиц состоит в том, что при изменении содержания любых ячеек таблицы может происходить автоматическое изменение содержания во всех прочих ячейках, связанных с измененными соотношением, заданным математическими или логическими выражениями (формулами). Простота и удобство работы с электронными таблицами снискали им широкое применение в сфере бухгалтерского учета, в качестве универсальных инструментов анализа финансовых, сырьевых и товарных рынков, доступных средств обработки результатов технических испытаний, то есть всюду, где необходимо автоматизировать регулярно повторяющиеся вычисления достаточно больших объемов числовых данных.

**Системы автоматизированного проектирования (САД-системы)** - предназначены для автоматизации проектно-конструкторских работ. Применяются в машиностроении, приборостроении, архитектуре. Кроме чертежно-графических работ эти системы позволяют проводить простейшие расчеты (например, расчеты прочности деталей) и выбор готовых конструктивных элементов из обширных баз данных.

Отличительная особенность САД-систем состоит в автоматическом обеспечении на всех этапах проектирования технических условий, норм и правил, что освобождает конструктора (или архитектора) от работ нетворческого характера. Например, в машиностроении САД-системы способны на базе сборочного чертежа изделия автоматически выполнить рабочие чертежи деталей, подготовить необходимую технологическую документацию с указанием последовательности переходов механической обработки, назначить необходимые инструменты, станочные и контрольные приспособления, а также подготовить управляющие программы для станков с числовым программным управлением (ЧПУ), промышленных роботов и гибких автоматизированных линий. Сегодня системы автоматизированного проектирования являются необходимым компонентом, без которого теряется эффективность реализации гибких производственных систем (ГПС) и автоматизированных систем управления технологическими процессами (АСУТП).

**Настольные издательские системы.** Назначение программ этого класса состоит в автоматизации процесса верстки полиграфических изданий. Этот класс программного обеспечения занимает промежуточное положение между текстовыми процессорами и системами автоматизированного проектирования.

Теоретически текстовые процессоры предоставляют средства для внедрения в текстовый документ объектов другой природы, например объектов векторной и растровой графики, а также позволяют управлять взаимодействием между параметрами текста и параметрами внедренных объектов. Однако на практике для изготовления полиграфической продукции эти средства либо функционально недостаточны с точки зрения требований полиграфии, либо недостаточно удобны для производительной работы. От текстовых процессоров настольные издательские системы отличаются расширенными средствами управления взаимодействием текста с параметрами страницы и с графическими объектами. С другой стороны, они отличаются пониженными функциональными возможностями по автоматизации ввода и редактирования текста. Типичный прием использования настольных издательских систем состоит в том, что их применяют к документам, прошедшим предварительную обработку в текстовых процессорах и графических редакторах.

**Экспертные системы.** Предназначены для анализа данных, содержащихся в *базах знаний*, и выдачи рекомендаций по запросу пользователя. Такие системы применяют в тех случаях, когда исходные данные хорошо формализуются, но для принятия решения требуются обширные специальные знания. Характерными областями использования экспертных систем являются юриспруденция, медицина, фармакология, химия. По совокупности признаков заболевания медицинские экспертные системы помогают

установить диагноз и назначить лекарства, дозировку и программу лечебного курса. По совокупности признаков события юридические экспертные системы могут дать правовую оценку и предложить порядок действий, как для обвиняющей стороны, так и для защищающейся.

Характерной особенностью экспертных систем является их способность к *саморазвитию*. Исходные данные хранятся в базе знаний в виде *фактов*, между которыми с помощью специалистов-экспертов устанавливается определенная система *отношений*. Если на этапе тестирования экспертной системы устанавливается, что она дает некорректные рекомендации и заключения по конкретным вопросам или не может дать их вообще, это означает либо отсутствие важных фактов в ее базе, либо нарушения в логической системе отношений. И том и в другом случае экспертная система сама может сгенерировать достаточный набор запросов к эксперту и автоматически повысить свое качество.

С использованием экспертных систем связана особая область научно-технической деятельности, называемая *инженерией знаний*. Инженеры знаний — это специалисты особой квалификации, выступающие в качестве промежуточного звена между разработчиками экспертной системы (программистами) и ведущими специалистами в конкретных областях науки и техники (экспертами).

**Редакторы HTML (Web-редакторы).** Это особый класс редакторов, объединяющих в себе свойства текстовых и графических редакторов. Они предназначены для создания и редактирования, так называемых *Web-документов (Web-страниц Интернета)*. Web-документы — это электронные документы, при подготовке которых следует учитывать ряд особенностей, связанных с приемом/передачей информации в Интернете.

Теоретически для создания Web-документов можно использовать обычные текстовые редакторы и процессоры, а также некоторые из графических редакторов векторной графики, но Web-редакторы обладают рядом полезных функций, повышающих производительность труда Web-дизайнеров. Программы этого класса можно также эффективно использовать для подготовки электронных документов и мультимедийных изданий.

**Браузеры (обозреватели, средства просмотра Web).** К этой категории относятся программные средства, предназначенные для просмотра электронных документов, выполненных в формате *HTML* (документы этого формата используются в качестве Web-документов). Современные браузеры воспроизводят не только текст и графику. Они могут воспроизводить музыку, человеческую речь, обеспечивать прослушивание радиопередач в Интернете, просмотр видеоконференций, работу со службами электронной почты, с системой телеконференций (групп новостей) и многое другое.

**Интегрированные системы делопроизводства.** Представляют собой программные средства автоматизации рабочего места руководителя. К основным функциям подобных систем относятся функции создания, редактирования и форматирования простейших документов, централизация функций электронной почты, факсимильной и телефонной связи, диспетчеризация и мониторинг документооборота предприятия, координация деятельности подразделений, оптимизация административно-хозяйственной деятельности и поставка по запросу оперативной и справочной информации.

**Бухгалтерские системы.** Это специализированные системы, сочетающие в себе функции текстовых и табличных редакторов, электронных таблиц и систем управления базами данных. Предназначены для автоматизации подготовки первичных бухгалтерских документов предприятия и их учета, для ведения счетов плана бухгалтерского учета, а также для автоматической подготовки регулярных отчетов по итогам производственной, хозяйственной и финансовой деятельности в форме, принятой для предоставления в налоговые органы, внебюджетные фонды и органы статистического учета. Несмотря на то что теоретически все функции, характерные для бухгалтерских систем, можно исполнять и другими вышеперечисленными программными средствами, использование бухгалтерских систем удобно благодаря интеграции разных средств в одной системе.

При решении о внедрении на предприятии автоматизированной системы бухгалтерского учета необходимо учитывать необходимость наличия в ней средств адаптации при изменении нормативно-правовой базы. В связи с тем, что в данной области нормативно-правовая база в России отличается крайней нестабильностью и подвержена частым изменениям, возможность гибкой перенастройки системы является обязательной функцией, хотя это требует от пользователей системы повышенной квалификации.

**Финансовые аналитические системы.** Программы этого класса используются в банковских и биржевых структурах. Они позволяют контролировать и прогнозировать ситуацию на финансовых, товарных и сырьевых рынках, производить анализ текущих событий, готовить сводки и отчеты.

**Геоинформационные системы (ГИС).** Предназначены для автоматизации картографических и геодезических работ на основе информации, полученной топографическими или аэрокосмическими методами.

**Системы видеомонтажа.** Предназначены для цифровой обработки видеоматериалов, их монтажа, создания видеоэффектов, устранения дефектов, наложения звука, титров и субтитров.

Отдельные категории прикладных программных средств, обладающие своими развитыми внутренними системами классификации, представляют *обучающие, развивающее, справочные и развлекательные* системы и программы. Характерной особенностью этих классов программного обеспечения являются повышенные требования к мультимедийной составляющей (использование музыкальных композиций, средств графической анимации и видеоматериалов).

### 3.5.2. Понятие об информационном и математическом обеспечении вычислительных систем

Наряду с аппаратным и программным обеспечением средств вычислительной техники в некоторых случаях целесообразно рассматривать *информационное обеспечение*, под которым понимают совокупность программ и предварительно подготовленных данных, необходимых для работы данных программ.

Рассмотрим, например, систему автоматической проверки орфографии в редактируемом тексте. Ее работа заключается в том, что лексические единицы исходного текста сравниваются с заранее заготовленным эталонным массивом данных (словарем). В данном случае для успешной работы системы необходимо иметь кроме аппаратного и программного обеспечения специальные наборы словарей, подключаемые извне. Это пример *информационного обеспечения* вычислительной техники.

В специализированных компьютерных системах (бортовых компьютерах автомобилей, судов, ракет, самолетов, космических летательных аппаратов и т. п.) совокупность программного и информационного обеспечения называют *математическим обеспечением*. Как правило, оно “жестко” записывается в микросхемы ПЗУ и может быть изменено только путем замены ПЗУ или его перепрограммирования на специальном оборудовании.

#### Вопросы для самоконтроля

1. В чем вы видите диалектический характер связи между *программным обеспечением* и *аппаратным*?
2. Назовите четыре основных уровня программного обеспечения. Каков порядок их взаимодействия?
3. К какому классу относятся программные средства, встроенные в видеомаягнитофон, программируемую стиральную машину, СВЧ - печь?
4. В чем преимущества и недостатки выполнения офисных работ (например, копировально-множительных) аппаратными и программными средствами?
5. Какие категории программного обеспечения могут быть использованы в работе малого предприятия, и для каких целей?

6. Какие виды работ, характерные для крупного промышленного предприятия (например, машиностроительного завода), могут быть автоматизированы с помощью компьютеров? Какие категории программных средств для этого необходимы?
7. Назовите основные категории программного обеспечения, относящиеся к классу графических редакторов. В чем состоит принципиальная разница между этими категориями?
8. Что общего и в чем различие между понятиями *программное обеспечение* и *информационное обеспечение* средств вычислительной техники?

## **Лекция 3.6. Программное обеспечение обработки текстовых данных**

### **3.6.1. Типы программ для подготовки текстовых документов**

Многообразие программных продуктов, предназначенных для создания и оформления текстовых документов можно условно разбить на *текстовые редакторы* и *текстовые процессоры*.

#### **3.6.1.1. Текстовые редакторы**

**Текстовые редакторы** – это программы для создания и редактирования текстовых документов. Редактирование текста – это комплекс операций по внутренней (смысловой) и внешней (оформительской) работе над текстом. Текст можно «кроить», т.е. вырезать из него куски, «склеивать» их, вставлять части из других текстов, менять их местами и пр. Можно изменять расположение текста на странице, формат шрифта отдельных символов, слов и целых абзацев, вставлять в текст иллюстрации (рисунки, графики, схемы и пр.).

Текстовые редакторы обеспечивают основные возможности по подготовке небольших и несложных документов. Сюда входят следующие простые операции:

- Ввод алфавитно-цифровой информации;
- Перемещение по набранному тексту;
- Вставка или удаление символов.

А также более сложные операции по работе с блоками (фрагментами) текста:

- Выделение блока;
- Удаление блока;
- Копирование, перемещение, вставка блока;
- Дополнительные удобства (возможность поиска фрагмента, поиск с заменой, печать документа и т.д.).

Большинство текстовых редакторов ориентированы на работу с чисто текстовыми файлами, среди которых могут быть тексты программ, написанные на различных языках, конфигурационные файлы, файлы настройки и др. Ярким представителем таких программных продуктов является MultiEdit (начиная с версии 5.0). Этот текстовый редактор имеет мощную систему контекстной замены, встроенный язык макрокоманд на уровне Visual Basic, средства поддержки внутренней среды, средства помощи при наборе ключевых слов.

**Программа Блокнот (NotePad)** - текстовый редактор, встраиваемый в операционные системы серии Windows (разработка Microsoft).

Блокнот предназначен для создания небольших, объемом до 50 килобайт, неформатированных текстов. Создаваемые файлы имеют формат \*.txt в кодовой таблице Windows 1251. Скромные возможности определяют область применения программы как редактора для создания коротких записок, текстов управляющих и системных файлов.

#### **3.6.1.2. Текстовые процессоры**

**Текстовые процессоры** – общее название программных средств, предназначенных для создания, редактирования и форматирования простых и комплексных текстовых

документов. В настоящее время в России наибольшее распространение имеет текстовый процессор Microsoft Word.

### 3.6.2. Основные версии текстового процессора MS Word

Первоначальные версии текстового процессора MS Word относятся к 80-м годам, и соответственно, к ОС MS-DOS. Последней версией процессора для неграфической среды была версия MS Word 5.0. Она позволяла создавать, редактировать и распечатывать форматированные текстовые документы. Поскольку ОС MS-DOS не является графической, экранное изображение не могло соответствовать печатному, и операции форматирования документа выполнялись в известной степени «вслепую». Однако возможность просмотра документа в «натуральном» виде реализовывалась специальным режимом предварительного просмотра, который сохранялся и в современных версиях программы, хотя и не имеет уже решающего значения.

Основным преимуществом текстового процессора Word 5.0, отличающим эту программу от конкурентных продуктов, была возможность встраивания в текст графических объектов, правда, без взаимодействия текста и графики (обтекания графических изображений текстом). Сегодня текстовым процессором Word 5.0 продолжают пользоваться при работе на устаревшем оборудовании (IBM PC AT 286).

Принцип соответствия экранного изображения печатному (WYSIWYG) впервые был реализован в следующей версии программы, которая называлась Microsoft Word for Windows (Word 6.0). Благодаря этому принципу значительно упростились и стали наглядными приемы форматирования документов. Будучи приложением Windows 3.1, программа получила возможность использовать системный буфер обмена, а пользователи мощное и удобное средство для создания комплексных документов.

Следующая версия называлась Microsoft Word 95 (Word 7.0), ориентированная на графическую ОС Windows 95. Основным достижением этой версии стало то, что после нее текстовый процессор использовался уже как отдельное приложение. В состав мощного офисного пакета Microsoft Office входит несколько приложений (с каждой новой версией пакета этот состав расширяется), и на процессор возлагаются дополнительные функции интеграции прочих приложений. Он занимает центральное положение в системе и позволяет организовать эффективный обмен данными между составляющими приложениями, что позволяет в значительной степени автоматизировать разработку офисных документов разной содержательности и сложности.

Еще одно важное нововведение седьмой версии - управление взаимодействием текста со встроенными объектами, что значительно расширяет возможности форматирования документа. Особый успех этой версии в России завоевали встроенные средства поддержки русского языка (автоматическая проверка орфографии и грамматики).

Восьмая версия программы Microsoft Word 97 (Word 8.0), вошедшая в состав пакета Microsoft Office 97, внесла относительно мало практически полезных изменений для повседневной офисной работы. Например, она жестко ориентирована на использование шрифтов UNICODE, что затруднило обмен данными с большинством приложений, и создала проблемы печати материалов на большинстве печатающих устройств.

Дополнительные средства оформления текстовых документов этой версии, имели практическое значение только при разработке электронных (экранных) документов. Возможность сохранения документов в «электронных» форматах HTML и PDF, рассчитанная на публикацию документов в Интернете, осталась не до конца проработанной и не вошла в практику web-дизайнеров. Начиная с этой версии текстовый процессор Microsoft Word можно рассматривать как средство автоматизации авторской деятельности. При использовании этой программы следует четко определить целевой объект – документ электронный или печатный. Для разных типов документов используют разные средства, приемы и методы. Применение неадекватных средств значительно усложняет последующую работу с документами. В итоге, в качестве средства разработки электронных документов

Word 97 не заменил web – редакторы, а в качестве средства создания печатных документов внес ряд неудобств по сравнению с предыдущей версией Word 95.

Последующая версия текстового процессора Microsoft Word 2000 (Word 9.0), входящая в состав пакета Microsoft Office 2000. В ней устранены основные недостатки предыдущей версии, заметно улучшена система управления и введены мощные средства поддержки сетевых режимов работы. Предполагается, что основным стилем производственной работы с текстовым процессором Word 2000 должна стать совместная деятельность рабочих групп над общими проектами в рамках корпоративных сетей.

### 3.6.3. Текстовый процессор MS Word 2000

#### Рабочее окно процессора MS Word 2000

Основные элементы управления рабочего окна процессора: строка меню, панель инструментов, рабочее поле и строка состояния, включающая индикаторы. Начиная с процессора MS Word 2000, панель инструментов является настраиваемой.

#### Режимы отображения документов

Начиная с шестой версии MS Word поддерживающей несколько режимов представления документов. В обычном режиме представляется только содержательная часть документа без реквизитных элементов оформления, относящихся не к тексту, а к печатным страницам (колонтитулы, колонцифры, подстраничные сноски и т.п.). Этот режим удобен на ранних этапах разработки документа (ввод текста, редактирование, рецензирование), а также во всех случаях, когда содержательная часть документа имеет более высокое значение, чем внешнее представление. В этом режиме операции с объемными документами проходят быстро, что важно при работе на мало производительных компьютерах.

В режиме web – документа экранное представление не совпадает с печатным. Это характерно для электронных публикаций в Word Wide Web (всемирная паутина), т.к. заранее не известно, каким средством просмотра и на каком оборудовании будет отображаться документ. Для электронных документов понятие печатной страницы не имеет смысла, поскольку назначенные параметры страницы не учитываются, а форматирование документа на экране является относительным. В этом режиме разрабатываются электронные публикации.

В режиме разметки экранное представление документа может полностью соответствовать печатному, вплоть до назначенных параметров печатной страницы. Этот режим удобен для большинства работ, связанных с форматированием текста, предназначенного для печати.

В режиме структуры документ отображается с утрированным выделением его структуры. Режим полезен в тех случаях, когда разработку документа начинают с создания плана содержания. Он отличается тем, что при его включении на панели инструментов автоматически открывается вспомогательная панель *Структура*, элементы управления которой позволяют править структуру документа.

#### Специальные средства ввода текста

Остановимся на особенностях текстового процессора Microsoft Word 2000, позволяющих автоматизировать ввод текста.

**Средства отмены и возврата действий.** Все операции ввода, редактирования и форматирования текста протоколируются текстовым процессором, и потому необходимое количество последних действий можно отменить. После отмены ряда действий существует возможность вернуться к состоянию, предшествовавшему отмене.

**Расширенный буфер обмена.** При компиляции документа путем использования фрагментов текста, взятых из разных первоисточников, удобно пользоваться расширенным буфером обмена.

**Автотекст.** Автотекст - это режим автоматического ввода фрагментов текста. Он

представлен двумя функциями: автозавершением и собственно автотекстом. Их принцип действия состоит в следующем.

Текстовый процессор хранит словарь автотекста, состоящий из слов и фраз, встречающихся в документах достаточно часто. При вводе первых четырех символов словарного элемента на экране появляется всплывающая подсказка с полным текстом слова или фразы. Если это то, что имел в виду пользователь, он завершает ввод всего фрагмента нажатием клавиши ENTER - так работает функция автозавершения. Однако пользователь может самостоятельно выбрать необходимый элемент текста из списка с иерархической структурой - это функция автотекста.

**Использование средства автозамены при вводе.** Последние версии текстового процессора Microsoft Word позволяют эффективно сократить объем вводимого текста за счет использования средства Автозамена. Оно позволяет заменить ввод длинных последовательностей символов произвольным (желательно коротким) сочетанием других символов. Например, если в тексте очень часто встречается словосочетание «диалоговое окно», его можно заменить коротким сочетанием «.до». Соответственно вместо «диалоговых окон» использовать «.дн», а вместо «диалогового окна» — «.да». Точка перед символами стоит специально, чтобы отличать их от двухбуквенных предлогов или союзов, таких как «да».

Настройку средства Автозамена выполняют в диалоговом окне Сервис → Автозамена. Для этого надо установить флажок **Заменять при вводе**, ввести заменяемую комбинацию в поле **Заменить**, а замещающую комбинацию в поле **На**, после чего пополнить список автозамены щелчком на кнопке **Добавить**.

Средство автоматической замены символов при вводе используется также для ввода специальных символов. Например, выполнив соответствующие настройки, можно вводить греческие буквы  $\pi$  и  $\rho$  обычным русским текстом: «пи» или «ро».

**Ввод специальных и произвольных символов.** При вводе текста часто существует необходимость ввода специальных символов, не имеющих соответствующей клавиши в раскладке клавиатуры, а также произвольных символов, раскладка для которых неизвестна. Основным средством для ввода специальных и произвольных символов, а также для закрепления их за избранными клавишами является диалоговое окно **Символ** (Вставка → Символ). Данное диалоговое окно имеет две вкладки: **Символы** и **Специальные символы**.

На вкладке **Специальные символы** присутствует список специальных символов, таких как «длинное» («полиграфическое») тире, «копирайт», «торговая марка» и других. Для вставки такого символа достаточно щелкнуть на кнопке **Вставить**. Вместе с тем, для большинства специальных символов существуют клавиатурные комбинации - они приведены в списке, и их стоит запомнить. На первых порах, пока навык их ввода не закреплен, это окно используют для получения справки. В том же окне имеются кнопки **Автозамена** и **Клавиша**, позволяющие либо выполнять ввод специальных символов обычными символами и автоматически производить замену, либо закрепить специальный символ за избранной комбинацией клавиш.

На вкладке **Символы** представлены элементы управления для ввода произвольных символов любых символьных наборов. Центральное положение в окне занимает таблица символов текущего набора. Выбор шрифта выполняют в раскрывающемся списке **Шрифт**. Если шрифт относится к категории универсальных шрифтов UNICODE, то для него имеется и возможность выбора символьного набора в соответствующем раскрывающемся списке **Набор**.

Если символ надо вставить только один раз, достаточно щелкнуть на командной кнопке **Вставить**. Если предполагается многократное использование данного символа, за ним можно закрепить постоянную комбинацию клавиш (кнопка **Клавиша**) или создать элемент для списка Автозамена с помощью одноименной кнопки.

## Специальные средства редактирования текста

Специальные средства редактирования, характерные для текстового процессора Microsoft Word, на примере последней версии Microsoft Word 2000.

**Режимы вставки и замены символов.** Текстовый процессор предоставляет возможность выбора между двумя режимами редактирования текста: режимом вставки и режимом замены. В режиме вставки вводимый текст «раздвигает» существующий текст, а в режиме замены новые символы замещают символы предшествующего текста, находившиеся в точке ввода. Режим вставки применяют при разработке основных содержательных блоков текстовых документов, а режим замены - при редактировании стандартных форм и стандартных элементов (колоннотитулов, реквизитных элементов в письмах, служебных записках, бланках).

Текущий режим правки текста индицируется на экране индикатором **Замена**. В режиме замены включен индикатор ЗАМ в строке состояния окна программы, в противном случае он выключен. Двойной щелчок на этом индикаторе позволяет переключать режимы. Настройка режима правки выполняется на вкладке **Правка** диалогового окна **Параметры** (Сервис → Параметры → Правка).

Если установлены флажки **Режим замены** и **Использовать клавишу INS для вставки**, правка осуществляется в режиме замены символов. Если оба эти флажка сброшены, то режим можно выбирать с помощью клавиши INSERT. Если флажок **Режим замены** сброшен, а флажок **Использовать клавишу INS для вставки** установлен, то правка осуществляется в режиме вставки.

**Использование Тезауруса.** Тезаурус представляет собой словарь смысловых синонимов. При подготовке технической документации особую роль играют смысловые синонимы к используемым глаголам. Для выделенного слова тезаурус удобно вызывать через пункт **Синонимы** контекстного меню. Однако этот прием срабатывает далеко не для всех слов (преимущественно для глаголов в неопределенной форме). Общий прием вызова тезауруса состоит в использовании команды строки меню **Сервис** → **Язык** → **Тезаурус**.

Окно Тезаурус имеет две панели. Его интересная особенность состоит в том, что в то время как на левой панели отображаются синонимы выделенного слова, на правой панели могут отображаться синонимы к выбранному синониму, то есть поиск синонима является как бы двухуровневым. Заменяющий синоним можно выбирать как на левой панели, так и на правой. Замена производится щелчком на командной кнопке **Заменить**. Кроме синонимов в некоторых случаях тезаурус позволяет находить антонимы слов и связанные (как правило, однокоренные) слова.

**Средства автоматизации проверки правописания.** Средства автоматизации проверки правописания включают средства проверки орфографии и грамматики. Текстовый процессор позволяет реализовать два режима проверки правописания - автоматический и командный.

В автоматическом режиме слова, содержащие орфографические ошибки, подчеркиваются красным цветом, а выражения, содержащие грамматические ошибки, - зеленым. Для того чтобы узнать характер ошибки, надо щелкнуть правой кнопкой мыши на помеченном фрагменте. В зависимости от характера ошибки контекстное меню содержит пункт **Орфография** или **Грамматика**. С их помощью открывается диалоговое окно, в котором имеются элементы управления для получения более точной справки о том, какое правило нарушено, и предложены варианты исправления предполагаемой ошибки.

Встроенное автоматическое средство проверки правописания является, по существу, экспертной системой и допускает настройку. Так, например, если рекомендации экспертной системы неточны или неприемлемы, от них можно отказаться командой **Пропустить** (обычно такое бывает при проверке грамматики). Если же слово отмечено как орфографическая ошибка только потому, что оно отсутствует в словаре системы автоматической проверки (например, слово броузер), то его можно добавить в словарь.

Встроенный словарь системы проверки правописания не подлежит правке. Все дополнения и изменения вносятся в специальный подключаемый пользовательский словарь. Каждый пользователь может создать несколько специализированных пользовательских словарей, ориентированных на различные области знаний (автомобильное дело, машиностроение, вычислительная техника и т. п.). Подключение нужного словаря для работы с конкретным документом выполняется выбором словарного файла в раскрывающемся списке **Вспомогательные словари** на вкладке **Сервис** → **Параметры** → **Правописание**. Постепенно наполняясь конкретным содержанием, вспомогательные словари пользователя становятся мощным средством повышения производительности его труда.

В командном режиме проверка правописания выполняется независимо от установки элементов управления на вкладке **Сервис** → **Параметры** → **Правописание**. Запуск средства проверки выполняют командой **Сервис** → **Правописание**. Проверка начинается от начала документа и продолжается до появления первой ошибки. В тех случаях, когда пользователь отказывается от предлагаемых исправлений и дает команду **Пропустить**, в документе накапливается список пропускаемых слов, то есть слов и выражений, не подлежащих проверке. Для того чтобы очистить этот список и начать проверку заново, используют командную кнопку **Сервис** > **Параметры** > **Правописание** > **Повторная проверка**.

**Средства рецензирования текста.** Под рецензированием можно понимать два процесса: редактирование текста с регистрацией изменений и комментирование текста. В отличие от обычного редактирования при рецензировании текст документа изменяется не окончательно - новый вариант и старый «сосуществуют» в рамках одного документа на правах различных версий.

Основным средством рецензирования является панель **Рецензирование** (Вид → Панели управления → Рецензирование). На ней представлены четыре группы элементов управления, предназначенных для:

- создания, просмотра и удаления примечаний;
- регистрации, просмотра, принятия и отмены изменений;
- выбора цвета выделения примечаний;
- сохранения версий документа.

Для создания примечания служит кнопка **Добавить примечание**. При ее использовании последнее слово текста выделяется заданным цветом и открывается дополнительная панель для ввода текста примечания. Когда примечание создано, его можно просмотреть в форме всплывающей подсказки, если навести указатель мыши на выделенное слово. Вместе с текстом примечания отображаются сведения о том, кто его внес.

Для регистрации изменений в тексте служит кнопка **Исправления**. Все редактирование текста в режиме регистрации исправлений считается неавторским и выделяется особым методом (метод выделения можно задать на вкладке **Исправления** диалогового окна **Сервис** → **Параметры**). Прочие элементы управления данной панели позволяют выполнять переходы между исправлениями, принимать их или отвергать.

Если документ проходит многоступенчатое редактирование, часто возникает необходимость хранить его промежуточные версии. Microsoft Word 2000 позволяет хранить несколько версий документа в одном файле. Это средство отличается тем, что при сохранении нескольких версий (в отличие от нескольких копий) эффективно используется рабочее место на диске. Дело в том, что при сохранении очередной версии не происходит повторного сохранения всего документа - сохраняются только отличия текущей версии от

предшествующей. Для сохранения текущей версии используют соответствующую кнопку панели **Рецензирование**, а для загрузки одной из промежуточных версий - команду **Файл** → **Версии**.

### **Форматирование текста**

Форматирование текста осуществляется средствами меню **Формат** или панели **Форматирование**. Основные приемы форматирования включают:

- выбор и изменение гарнитуры шрифта;
- управление размером шрифта;
- управление начертанием и цветом шрифта;
- управление методом выравнивания;
- создание маркированных и нумерованных списков (в том числе многоуровневых);
- управление параметрами абзаца.

#### **3.6.4. Характеристика основных шрифтов Windows**

По способу создания изображения шрифты подразделяются на *растровые* и *векторные*. Растровый шрифт получается на основе некоторой матрицы пикселей (экранных точек), векторный – на основе координат точек, элементарных векторов, уравнений линий. Набор из четырех основных шрифтов – обычный, курсив, полужирный, полужирный курсив – в совокупности называется *гарнитурой* шрифта. Каждая гарнитура имеет свое наименование, например, Arial, Times New Roman, Tahoma и другие. Каждой гарнитуре в системной папке Windows\Fonts соответствует один файл для векторного шрифта или несколько файлов (по числу размеров) для растрового шрифта. Растровые шрифты обычно бывают моноширинными, т.е. все буквы одной высоты имеют одинаковую высоту, как в пишущей машинке. Например, русская буква «Щ» и латинская буква “I” будут иметь в написании одинаковую ширину, что не улучшает внешний вид текста. Этот недостаток исправлен в векторных шрифтах, которые обычно являются пропорциональными, т.е. ширина определяется начертанием. На печати текст выполненный пропорциональными шрифтами, выглядит лучше моноширинных, но выравнивание текста пробелами становится невозможным.

Под размером шрифта понимается его вертикальный размер, который измеряется в пунктах. Один пункт равен 1/72 дюйма – примерно 0,353 мм. Шрифт размером 10 пунктов, его называют десятый кегль, часто используется в книгах. Четырнадцатым кеглем писала пишущая машинка. Это размер шрифта часто применяется сейчас при составлении различных документов. Еще одно преимущество векторных шрифтов – это их масштабируемость, т.е. все размеры и формы символов шрифта указанной гарнитуры каждый раз пересчитываются из некоторой единой математической заготовки, которой сообщается вертикальный размер.

При выборе гарнитуры шрифта следует иметь в виду, что существует две категории шрифтов: с засечками и без засечек (рубленые). Характерными представителями первой категории являются шрифты семейства Times, а второй категории - шрифты семейства Arial. Шрифты, имеющие засечки, легче читаются в больших текстовых блоках — их рекомендуется применять для оформления основного текста.

Шрифты, не имеющие засечек, рекомендуется использовать для заголовков в технических текстах, а также для оформления дополнительных материалов (врезок, примечаний и прочего).

#### **Вопросы для самоконтроля:**

1. Назначение текстового процессора.
2. В чем отличие текстовых редакторов и текстовых процессоров?
3. Чем отличаются режимы вставки и замены?
4. Какова роль буфера промежуточного хранения?

5. Для чего нужна операция откатки?
6. Какова роль окон в организации работы с текстом?
7. Для каких целей производится выделение фрагментов текста?
8. В чем сущность режима поиска и замены?
9. Как производится проверка правописания слов?
10. Для какой цели и как используется словарь синонимов?
11. В чем сущность режима проверки синтаксиса и стиля?
12. Расскажите о возможностях текстового процессора по автоматизации технологии работы.
13. В чем сходство и различие текстового процессора и издательской системы?
14. Какие параметры шрифтов вам известны?

## **Лекция 3.7. Электронные таблицы**

### **3.7.1. Основные понятия**

#### **3.7.1.1. История появления и развития ЭТ**

Идея создания ЭТ возникла у студента Гарвардского университета (США) Дэна Бриклина в 1979. Выполняя скучные вычисления экономического характера с помощью бухгалтерской книги, он и его друг Боб Франкстон, который разбирался в программировании, разработали первую программу ЭТ, названную VisiCalc. VisiCalc скоро стала одной из наиболее успешных программ. Скоро появились ЭТ-аналоги (например, Super Calc), в которых основные идеи VisiCalc были многократно усовершенствованы.

Новый существенный шаг в развитии ЭТ-появление в 1982г. на рынке программных средств Lotus 1-2-3, разработанных для компьютеров типа IBM. Lotus – первый ТП, интегрировавший в своем составе, помимо обычных инструментов, графику и возможность работы с СУБД. Следом же на рынке появились новые ЭТ. Такие как VP Planner компании Paperback Software и Quattro Pro компании Borland International, которые предложили пользователю практически тот же набор инструментов, но по значительно более низким ценам.

Следующий шаг – появление в 1987 г. ТП Excel фирмы Microsoft. Программа предложила более простой графический интерфейс в комбинации с ниспадающими меню, значительно расширив при этом функциональные возможности пакета и повысив качество выходной информации. В настоящее время Excel занимает ведущее место на рынке, благодаря облегченному освоению программы и работы с ней, несмотря на выпуск новой версии ЭТ компании Lotus, в которой использована трехмерная таблица с улучшенными возможностями.

Имеющиеся сегодня на рынке ТП способны работать в широком круге экономических приложений и могут удовлетворить практически любого пользователя.

#### **3.7.1.2. Интерфейс табличного процессора**

Что такое электронная таблица? ЭТ – компьютерный эквивалент обычной таблицы, в клетках (ячейках) которой записаны данные различных типов: тексты, формулы, числа. Результат вычисления формулы в ячейке является изображением этой ячейки. Числовые данные и даты могут рассматриваться как частный случай формул. Для управления ЭТ используется специальный комплекс программ – **табличный процессор (ТП)**.

Главное достоинство ЭТ – это возможность мгновенного пересчета всех данных, связанных формульными зависимостями, при изменении значения любой ячейки.

#### **Строки, столбцы, ячейки и их адреса**

Рабочая область ЭТ состоит из строк и столбцов, имеющих свои имена. Имена строк – их номера. Нумерация строк начинается с 1 и заканчивается максимальным числом,

установленным для данной программы. Имена столбцов – это буквы латинского алфавита сначала от А до Z, затем от AA до AZ, BA до BZ и т. д. (для EXCEL от А до IV – 256 столбцов).

Максимальное количество строк и столбцов определяется особенностями используемой программы и объемом памяти компьютера. Современные программы дают возможность создавать ЭТ, содержащие более 1 мил. ячеек, хотя для практических целей в большинстве случаев этого не требуется.

Применение ЭТ упрощает работу с данными и позволяет получать результаты без проведения расчетов вручную или специального программирования. Наиболее широкое применение ЭТ нашли в экономических и бухгалтерских расчетах, но ЭТ можно эффективно использовать и в научно – технических задачах, например, для :

- 1) проведения однотипных расчетов над большим набором данных;
- 2) автоматизации итоговых вычислений;
- 3) решения задач путём подбора значений параметров, табулирования формул;
- 4) обработки результатов экспериментов;
- 5) проведения поиска оптимальных значений параметров;
- 6) подготовки табличных документов;
- 7) построения диаграмм и графиков по имеющимся данным.

Пересечение строки и столбца образуют *ячейку* таблицы, имеющую свой уникальный адрес. Для указания адресов ячеек в формулах используются ссылки (например, А2 или D23).

*Ячейка* – область, определяемая пересечением столбца и строки ЭТ.

*Адрес ячейки* определяется именем (номером) столбца и номером строки.

*Ссылка* - способ (формат ) указания адреса ячейки.

### **Указание блока ячеек**

В ЭТ существует понятие блока (диапазона) ячеек, также имеющего свой уникальный адрес.

**Блок ячеек** – группа последовательных ячеек. Блок ячеек может состоять из одной ячейки, строки, столбца (или его части), а также последовательности строк или столбцов (или их частей). Адрес блока ячеек задается указанием ссылок первой и последней его ячеек, разделенных символом двоеточие “ : ”.

Пример:

Адрес ячейки, образованной на пересечении столбца В и строки 3 ,будет выражаться ссылкой В3;

- Адрес блока, образованного в виде части строки 1 , будет А1:Н1;
- Адрес блока, образованного в виде части столба В, В1: В10;
- Адрес блока, образованного в виде прямоугольника, будет D4:F5.

Каждая команда ЭТ требует указания блока ячеек, в отношении которых она должна быть выполнена. Блок используемых ячеек может быть указан двумя путями: либо непосредственным набором с клавиатуры начального и конечного адресов ячеек, формирующих диапазон, либо выделением соответствующей части таблицы при помощи клавиш управления курсором или мышью (последние удобнее всего) .

Типичные установки, принимаемые по умолчанию на уровне всех ячеек таблицы являются: ширина ячейки в 9 рядов , левое выравнивание для символьных данных и основной формат для цифровых данных с выравниванием вправо.

При работе на экран выводится рабочее поле таблицы и панель управления. Панель управления обычно включает: Главное меню, вспомогательную область управления, строку ввода и строку подсказки. Расположение этих областей может быть произвольным и зависит от особенностей конкретного ТП.

Строка главного меню содержит имена меню основных режимов программы. Через них, пользователь получает доступ к ниспадающему меню, содержащему перечень входящих команд. Некоторые команды меню имеют дополнительные подменю.

Вспомогательная область управлению включает:

- строку состояния;
- панели инструментов;
- вертикальную и горизонтальную линейки прокрутки.

### 3.7.2. Типы входных данных

В каждую ячейку пользователь может ввести данные одного из следующих типов: текстовые, числовые, даты, формулы и функции и пр.

**Текстовые данные** имеют описательный характер. Они могут включать в себя алфавитные, числовые и специальные символы.

**Числовые данные** не могут содержать алфавитных и специальных символов, т.к. с ними производятся математические операции. Исключением являются десятичная точка и знак числа, стоящий перед ним.

**Даты** являются особым типом входных данных. Этот тип данных обеспечивает выполнение таких функций, как добавление к дате числа (пересчет даты вперед и назад) или вычисление разности двух дат (длительности периода). Даты имеют внутренний (например, дата может выражаться количеством дней от начала 1900 года или порядковым номером дня по Юлианскому календарю) и внешний формат, используемый для ввода и отображения дат.

**Формулы.** Видимое на экране содержимое ячейки, возможно, - результат вычислений, произведенных, по имеющейся, но не видимой в ней формуле. С помощью формул в таблицах производятся все вычисления. Результат помещается в ячейку, где формула находится.

Формула начинается со знака «=» и представляет собой совокупность математических операторов, чисел, ссылок и функций. При вычислениях с помощью формул соблюдаются принятый в математике порядок выполнения арифметических операций.

Формула состоит из операторов и операндов, расположенных в определенном порядке. В качестве операндов используются данные, а также ссылки ячеек или блоков ячеек. Операторы в формулах обозначают действия, производимые с операндами. В зависимости от используемых операторов различают арифметические (алгебраические) и логические формулы. В арифметических формулах используются операторы:

- арифметических действий: сложение (+), вычитание (-), умножение (\*), деление (/), возведение в степень (^);
- операций сравнения: равно (=), не равно (<>), больше (>), не больше (<=), меньше (<), не менее (>=).

Результатом вычисления арифметической формулы является число.

Логические формулы могут содержать указанные операции сравнения, а также специальные логические операторы:

«НЕ» - логическое отрицание «NOT»;

«И» - логическое умножение «AND»;

«ИЛИ» - логическое сложение «OR».

Логические формулы определяют, истинно или ложно выражение. Истинным выражениям присваивается численная величина 1, а ложным – 0. Таким образом, вычисление логической формулы заканчивается получением оценки «Истина» (1) или «Ложь» (0).

**Функции.** Функция представляет собой программу с уникальным именем, для которой пользователь должен задать конкретные значения аргументов, стоящих в скобках после ее имени. Функции вводятся в ячейки таблиц в составе формул либо отдельно. В ЭТ могут быть представлены следующие категории функций:

- *Математические* – выполняют различные математические операции, например, вычисление логарифмов, тригонометрических функций, преобразование радиан в градусы и т.п.
- *Статистические* – выполняют операции по вычислению параметров случайных величин или их распределений, представленных множеством чисел, например, стандартного отклонения, среднего значения, медианы и т.п.
- *Текстовые* – выполняют операции над текстовыми строками или последовательностью символов, вычисляя длину строки, преобразовывая заглавные буквы в строчные и т.п.
- *Логические* – используются для построения логических выражений, результат которых зависит от истинности проверяемого условия.
- *Финансовые* – используются в сложных финансовых расчетах, например, определение нормы дисконта, размера ежемесячных выплат для погашения кредита, определение амортизационных отчислений и другие функции.

Все функции имеют одинаковый формат записи и включают имя функции и находящийся в круглых скобках перечень аргументов, разделенных запятыми или «;».

#### **Наиболее часто встречающиеся функции**

**СУММ (Список)** – математическая функция, определяющая сумму всех числовых значений в Списке. Список может состоять из адресов ячеек и блоков, а также числовых значений. Например, СУММ(А3:Е3;230).

**СРЗНАЧ (Список)** – статистическая функция. Определяет среднее арифметическое значение всех перечисленных в Списке величин.

**ЕСЛИ (Лог\_выражение;Истина;Ложь)**– логическая функция, проверяющая на истинность заданное логическое условие. Если условие выполняется, то результатом функции является значение аргумента «Истина». Если условие не выполняется, то результатом становится значение аргумента «Ложь».

В Excel встроены более 300 функций. Использовать их позволяет мастер функций. Все функции в мастере разбиты на категории, поэтому сначала выбирают категорию, а затем функцию. Мастер выводит подсказку: шаблон для функции, который помогает понять правила ее записи, и описание ее назначения.

### **3.7.3. Автоматическое изменение относительных ссылок при копировании и перемещении формул**

#### **3.7.3.1. Буфер промежуточного хранения**

Важной особенностью многих ЭТ является буфер промежуточного хранения. Буфер используется при выполнении команд копирования и перемещения для временного хранения копируемых или перемещаемых данных, после которого они направляются по новому адресу. При удалении данных они также помещаются в буфер. Содержимое буфера сохраняется до тех пор, пока в него не будет записана новая порция данных.

---

*Буфер промежуточного хранения* – область оперативной памяти, предоставляемая в распоряжении пользователя, при помощи которой он может перенести данные из одной части таблицы в другую, из одного окна (таблицы) в другое или из одного приложения Windows в другое.

---

### 3.7.3.2. Относительная и абсолютная адресация

При копировании или перемещении формулы в другое место таблицы необходимо организовать управление формированием адресов исходных данных. Поэтому в ЭТ при написании формул используется понятие *относительной* и *абсолютной* ссылок.

**Абсолютная ссылка** - это не изменяющийся при копировании формулы адрес ячейки, содержащей исходное данные. Для указания абсолютной адресации вводится символ \$. Различают два типа абсолютной ссылки: *полная* и *частичная*.

Полная абсолютная ссылка указывается, если при копировании или перемещении адрес клетки, содержащей исходное данные, не меняется. Для этого символ \$ становится перед именем столбца и номером строки.

Частичная абсолютная ссылка указывается, если при копировании или перемещении не меняется номер строки или именование столбца. При этом символ в первом случае становится перед номером строки, а во втором - перед именованием столбца.

**Относительная ссылка** - это изменяющийся при копировании и перемещении формулы адрес ячейки, содержащей исходное данные (операнд). Изменение адреса происходит по правилу относительной ориентации клетки с исходной формулой и клеток с операндами. Форма записи относительной ссылки совпадает с обычной записью.

### 3.7.3.3. Правило относительной ориентации клетки

Формула, где в качестве операнда используются ссылки ячеек, воспринимается системой как шаблон, а ссылки ячеек в таком шаблоне - как средство указания на место положение ячеек с операндами относительно ячейки с формулой.

Рассмотрим правило относительной ориентации клетки на примере:

Клетка со ссылкой C2 содержит формулу - шаблон сложения двух чисел, находящихся в ячейках A1 и B4. Эти ссылки являются относительными и отражают ситуацию расположения исходных данных в ячейках A1 и B4 и результаты вычисления по формуле в ячейке C2. По правилу относительной ориентации клеток ссылки исходных данных воспринимаются системой не сами по себе, а так, как они расположены относительно клетки C2 ссылка A1 указывает на клетку, которая смещена относительно клетки C2 на одну клетку вверх и на две клетки влево; ссылка B4 указывает на клетку, которая смещена относительно клетки C2 на две клетки вниз и одну клетку влево.

### 3.7.3.4. Копирование формул

Особенностью ЭТ является возможность автоматического изменения ссылок при копировании и перемещении формул. Копирование содержимого одной ячейки (блока ячеек) в другую (блока ячеек) производится для упрощения ввода однотипных данных и формул. При этом осуществляется автоматическая настройка относительных ссылок операндов. Для запрета автоматической настройки адресов используют абсолютные ссылки ячеек.

Исходная формула, подлежащая копированию или перемещению, воспринимается как некий шаблон, где указывается местоположение входных данных относительно местоположения клетки с формулой.

Копируемую формулу назовем формулой - оригиналом. Скопированную формулу - формулой - копией. При копировании формул действует правило относительной ориентации клеток. Поэтому после окончания копирования относительное расположение клеток, содержащих формулу копию и исходные данные (заданные относительными ссылками), остается таким же, как в формуле - оригинале. Поясним на примере.

До копирования

После копирования

	A	B	C
1			5
2		2	

3	=C1+B2		
---	--------	--	--

	A	B	C
1			5
2		2	
3	=C1+B2	=D1+C2	=E1+D2

	A	B	C
1			5
2		2	
3	=C\$1+B\$2		

	A	B	C
1			5
2		2	
3	=C1+B\$2		

	A	B	C
1			5
2		2	
3	=C\$1+B\$2	=C\$1+B\$2	=C\$1+B\$2

	A	B	C
1			5
2		2	
3	=C1+B\$2	=C1+C\$2	=C1+D\$2

- при копировании формулы с использованием относительных ссылок происходит их автоматическая подстройка;
- Применение абсолютных ссылок запрещает автоматическую настройку адресов, и копируемая формула сохраняет свой первоначальный вид;
- С частично абсолютными ссылками.

### 3.7.3.5. Перемещение формул

В ЭТ часто перемещают данные ячейки (диапазона ячеек) в другую заданную ячейку (блок ячеек). После перемещения данных исходная ячейка окажется пустой. Это главное отличие перемещения от процесса копирования, в котором копируемая ячейка сохраняет свои данные. Перемещение формул также связано с автоматической подстройкой входящих в нее адресов операндов. При перемещении формул, так же как при их копировании, действует правило относительной ориентации клеток. Поэтому после перемещения относительное расположение клеток, содержащих перемещенную формулу и исходные данные (заданные относительными адресами), сохраняется таким же, как в формуле - оригинале.

До перемещения

	A	B	C
1	5		
2	2		
3	=A1+A2		

	A	B	C
1	5		
2	2		
3	=A1+A2		

	A	B	C
1	5	=A3*4	
2	2		
3	=A1+A2		

3			=A1+A2
---	--	--	--------

	A	B	C
1			5
2			2
3			=C1+C2

	A	B	C
1	5	=C3*4	
2	2		
3			=A1+A2

После перемещения

	A	B	C
1	5		
2	2		

### **3.6.5. Технология работы с электронными таблицами**

Хотя работа в каждой ЭТ имеет свои особенности, можно говорить о некоторой обобщенной (усредненной) технологии работы с ней. Изобразим технологию работы в виде схемы (рис. 3.7.1).

На этапе 1 формируется структура таблицы. Структура включает: определение заголовка таблицы, названий строк и столбцов, а также ввод в ячейки таблицы исходных данных, формул и функций.

На этапе 2 производится работа с данными, которая состоит в исследовании сформированной таблицы. Также исследование может быть связано с использованием определенных математических моделей (моделированием), методов одновременной работы с несколькими таблицами и методов работы с данными.

Математические модели помогают пользователю на основе имеющейся таблицы получить новую информацию решением таких задач компьютерного моделирования: «Что, если»

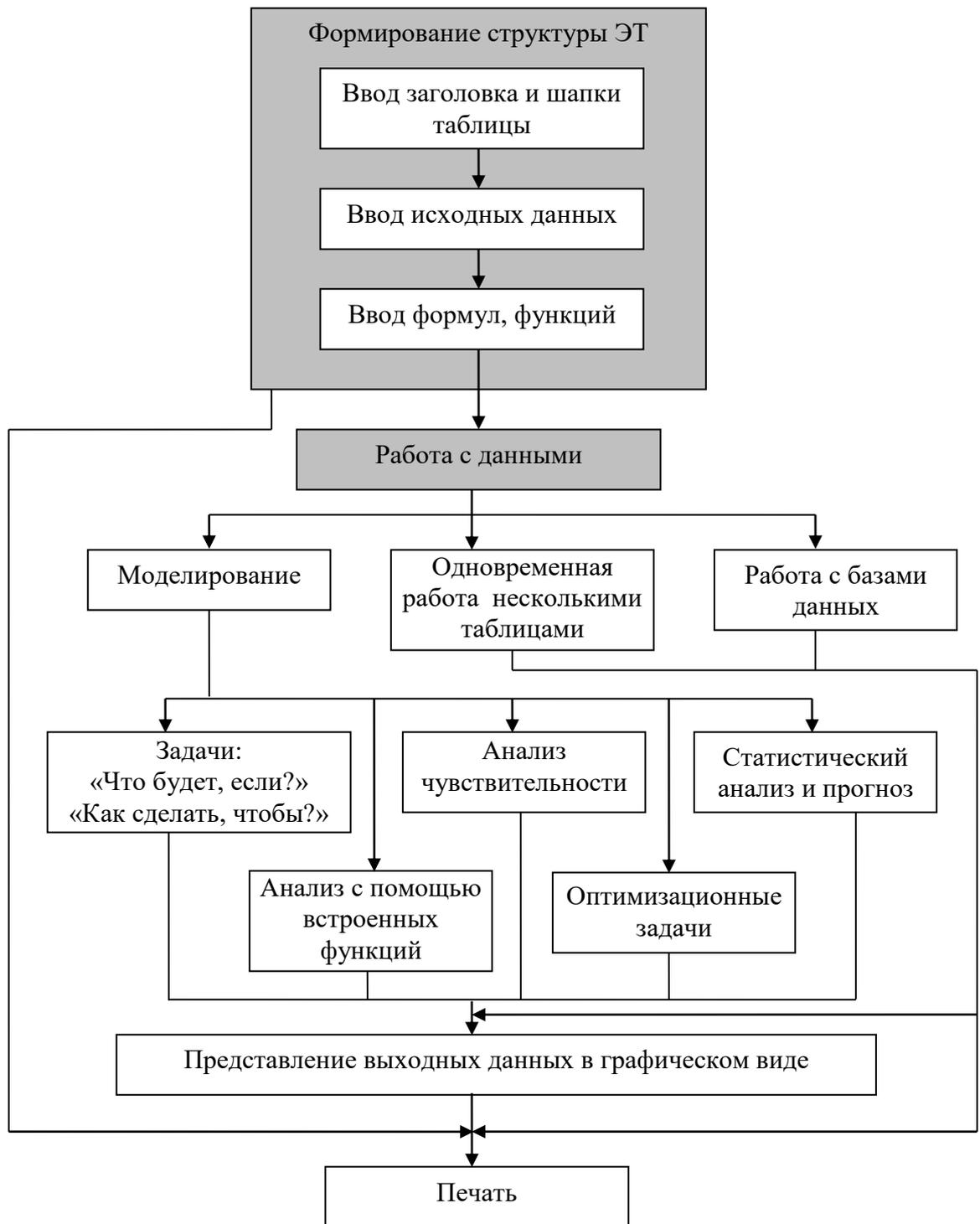


Рис. 3.7.1. Обобщенная технология работы с ЭТ

ли?», анализ чувствительности и др. Решение задач типа: «Что будет, если?» - дает возможность узнать, как изменятся выходные параметры при изменении одной или нескольких входных величин (условий). Расширением таких задач являются задачи анализа чувствительности, позволяющие определить, как изменится решение модели при изменении одной или нескольких входных величин с заданным шагом в определенном диапазоне значений. Обратной по отношению к задаче «Что будет, если?» - является задача «Как сделать, чтобы?». Она возникает в случае, если целью является достижение определенного значения модели, и ищутся значения входных параметров, обеспечивающих достижение этой цели. Различные виды анализа данных, содержащихся в исходной таблице можно проводить с использованием встроенных функций и процедур. Так, входящие в состав ЭТ

статистические функции могут использоваться в статистическом анализе или для прогноза содержащихся в таблице данных. Использование финансовых функций позволяет произвести анализ эффективности планируемых капиталовложений, рассчитать стоимость ценных бумаг или размер амортизационных отчислений. Для решения оптимизационных задач используются специальные модели математического программирования.

Часто фирма (предприятие) имеет центральный офис и несколько филиалов. В таких условиях появляется задача объединения различных документов и отчетов, приходящих из этих филиалов. Решение подобной задачи требует использования специальных многотабличных связей и программных методов для манипулирования с файлами и генерации отчетных форм. Одновременная работа с несколькими таблицами – это одна из возможностей работы с данными в электронных таблицах.

Иногда при работе с большими ЭТ требуется найти ту или иную строку (столбец) или отсортировать строки (столбцы) в нужном порядке. Для этого в ЭТ предусмотрен ограниченный программный инструментариий СУБД, позволяющий манипулировать строками и столбцами как компонентами БД.

Этап 3 технологии позволяет в графическом виде представить результаты, полученные на первом и втором этапах, и наиболее ярко их интерпретировать.

Этап 4 обеспечивает вывод результирующих данных на печать. При этом результаты могут быть распечатаны в табличном виде или в виде графических диаграмм.

### 3.7.5. Проектирование ЭТ

Проектирование ЭТ рассмотрим на примере форсирования прогноза финансовой деятельности некоторой компании за 5 лет.

Доход в любом году определяется как произведение объема продаж в натуральном выражении на продажную цену. Размер прибыли при этом определяется как разность между доходом и суммой расходов данного года.

Пользователем задаются исходные данные первого года. Данные всех последующих лет вычисляются ЭТ на основе допущений о характере их изменений в будущем.

	А	В	С	Д	Е
1					
2		2	2	2	2
		006	007	008	009
3	Объем продаж, шт.	1	1	1	1
		0000	1800	3924	6430
4	Цена	\$	\$	\$	\$
		2.00	2.10	2.21	2.32
5	Доход	\$	\$	\$	\$
		20000	24780	30702	30702
6	Расходы	\$	\$	\$	\$
		15000	15750	16537	81364
7	Прибыль	\$	\$	\$	\$
		5000	9030	14165	20676
8					
9	Прогнозные допущения				
10	Рост объема продаж	18,00%			
11	Рост цен	5,00%			

Эти данные расположены в нижнем левом углу таблицы.

Расходы и продажные цены определяются с учетом заданного роста цен, а объемы продаж – с учетом роста объемов продаж. При изменении данных допущений ЭТ должна немедленно пересчитывать значения всех прогнозируемых финансовых характеристик. Для

этого ЭТ проектируется таким образом, чтобы прогнозируемые параметры определялись по формулам, зависимым от исходных прогнозных допущений.

Проектирование ЭТ основано на использовании уже рассмотренных нами команд копирования и перемещения.

Разработка любой ЭТ начинается с постановки цели. ЭТ, проектируемая для целей финансового прогнозирования, должна всякий раз пересчитывать финансовые показатели при изменении принятых прогнозных допущений.

Создание ЭТ начинается с ввода названий столбцов и строк.

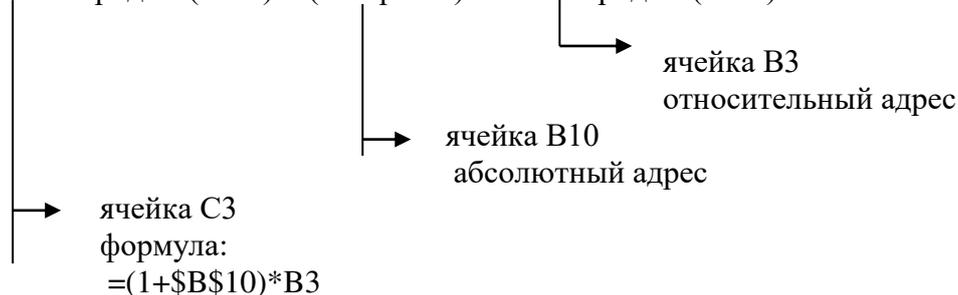
Столбцами в нашей таблице являются годы, составляющие прогнозный период, а строками – прогнозируемые финансовые показатели. Кроме того, в таблицу входят прогнозные допущения.

	A	B	C	D	E
1					
2		2006	2007	2008	2009
3	Объем продаж, шт.	10000	$=(1+\$B\$10)*B3$	...	...
4	Цена	\$2.00	$=(1+\$B\$11)*B4$		
5	Доход	$=B3*B4$	$=C3*C4$		
6	Расходы	15000	$=(1+\$B\$11)*B6$		
7	Прибыль	$=B5-B6$	$=C5-C6$		
8					
9	Прогнозные допущения				
10	Рост объема продаж	18,00%			
11	Рост цен	5,00%			

Далее вводятся данные первого года (в нашем примере 2006), которые частично задаются как числовые данные (объем продаж, продажная цена), а частично – как формулы. В ячейку B5 записываем формулу для вычисления дохода  $=B3*B4$ , а в ячейку B7 – формулу для вычисления прибыли  $=B5-B6$ . И вводим числовые данные, выражающие прогнозные допущения.

Далее вводим формулы в столбец второго года (2007). Эти формулы учитывают результаты первого года и, кроме того, отражают прогнозные допущения. Так, объем продаж в 2007 году определяется как объем продаж в 2006 году, умноженный на процент роста, указанный в прогнозных допущениях.

Объем продаж (2007) = (1+%роста)\* объем продаж (2006)



В формуле используем относительные и абсолютные адреса, что позволит скопировать ее в оставшиеся колонки. Абсолютный адрес для ячейки, содержащей процент роста объема продаж, предполагает ее обязательное использование для всех расчетов в рамках данной ЭТ. Относительный адрес ячейки, содержащей объем продаж предыдущего года, дает возможность его подстройки при копировании формулы, поскольку сохраняется логика расчета объема продаж для последующих лет.

Пересчет остальных параметров из столбца B в столбец C выполняется аналогичным образом.

Остальные столбцы (D, E, F) заполняются простым копированием формул, содержащихся в столбце C. Команда копирования, при этом автоматически подстроит содержащиеся в них относительные адреса ячеек.

Построенная ЭТ дает возможность создавать всевозможные финансовые прогнозы, изменяя прогнозные допущения. Например, изменив одно или несколько прогнозных допущений, определить, что произойдет с прибылью в 2010 году.

### **3.5.7. Объединение ЭТ**

При работе с ЭТ часто возникает необходимость их объединения.

Среди инструментов объединения ЭТ отметим:

- организацию межтабличных связей;
- консолидацию ЭТ или их частей;
- объединение файлов.

#### **Организация межтабличных связей**

Связи между таблицами осуществляются путем использования внешних ссылок (адресов ячеек), содержащих помимо имени столбца и номера строки имя файла, данные из которого используются. Например, в текущей таблице, расположенной на листе2, надо использовать данные из ячейки В3 на листе1. В нужную ячейку текущей таблицы можно записать ссылку следующим образом: Лист1!В3. Еще один пример, если надо использовать данные из ячейки С2 таблицы, расположенной на листе2, содержащемся в файле Exam.wq1, в нужную ячейку текущей таблицы можно записать внешнюю ссылку следующим образом: [Exam.wq1]Лист2!С2.

#### **Консолидация электронных таблиц или их частей**

Помимо создания межтабличных связей путем указания имен файлов, содержащий связываемые таблицы в ссылках и формулах, многие ЭТ предлагают пользователю специальный режим консолидации. Этот режим содержит необходимые команды для объединения таблиц или их частей, расположенных как на одном листе, так и на разных листах и даже в разных рабочих книгах. С помощью консолидации могут быть сведены в одной таблице, например, данные о продажах и затратах различных филиалов фирмы.

#### **Команда объединения файлов**

Многие современные табличные процессоры имеют в своем арсенале команду объединения файлов. Она имеет три форму, используемые для копирования, суммирования или вычитания данных из исходных таблиц в объединенную таблицу.

#### **Вопросы для самоконтроля**

1. Опишите структуру типового интерфейса электронной таблицы.
2. Перечислите и поясните основные типы входных данных, которые могут быть введены в ячейки электронной таблицы.
3. Что такое формула в электронной таблице и ее типы. Приведите примеры.
4. Что такое функция в электронной таблице и ее типы. Приведите примеры.
5. Поясните очередность выполнения операций в арифметических формулах.
6. Как указывается блок (диапазон) ячеек при выполнении какой-либо команды?
7. Поясните, для чего используются абсолютные и относительные адреса ячеек.
8. В чем суть правил автоматической настройки формул при выполнении операций копирования и перемещения?
9. Покажите на примерах все возможные варианты автоматического изменения адресов в формулах при выполнении операции копирования.
10. Покажите на примерах все возможные варианты автоматического изменения адресов в формулах при выполнении операции перемещения.

11. Перечислите и поясните содержание основных технологических этапов работы с электронной таблицей.
12. Какие способы объединения нескольких исходных электронных таблиц в одну вам известны?
13. Как использовать электронную таблицу для моделирования по типу решения задачи «Что будет, если?».

## **Лекция 3.8. Электронные презентации**

### **3.8.1. Программа Microsoft PowerPoint. Режимы PowerPoint**

Для подготовки презентации разработано немалое количество программных продуктов. Наиболее широкое применение получила программа Microsoft PowerPoint, вошедшая в состав пакета Microsoft Office 2000.

В PowerPoint для удобства работы над презентациями существуют разные режимы. Два основных режима, используемые в PowerPoint — обычный режим и режим сортировщика слайдов. Для быстрого переключения режимов служат кнопки в левом нижнем углу окна PowerPoint.

#### **Обычный режим**

В обычном режиме отображаются три области: область структуры, область слайда и область заметок. Эти области позволяют одновременно работать над всеми аспектами презентации. Размеры областей можно изменять, перетаскивая их границы.

**Область структуры.** Область структуры служит для организации и развертывания содержимого презентации. В ней можно вводить текст презентации и приводить в порядок пункты списка, абзацы и слайды.

**Область слайда.** В области слайда отображается текст каждого слайда с учетом форматирования. На отдельные слайды можно добавлять рисунки, фильмы, звуки, анимацию и гиперссылки.

**Область заметок.** Область заметок служит для добавления заметок докладчика или сведений для аудитории. Если в заметках должен быть рисунок, добавлять заметки следует в режиме страниц заметок.

Эти три области также присутствуют при сохранении презентации как Web-страницы. Единственное отличие состоит в том, область структуры отображается как содержание, позволяющее перемещаться по презентации.

#### **Режим сортировщика слайдов**

В режиме сортировщика слайдов на экране в миниатюре отображаются сразу все слайды презентации. Это упрощает добавление, удаление и перемещение слайдов, задание времени показа слайдов и выбор способов смены слайдов. Кроме того, можно просматривать анимацию на нескольких слайдах, выделив требуемые слайды и выбрав команду Просмотр анимации в меню Показ слайдов.

При создании презентации в любой момент можно запустить показ слайдов и просмотреть презентацию, нажав кнопку Показ слайдов.

#### **Вопросы для самоконтроля**

1. Какие режимы работы Microsoft PowerPoint вы знаете?
2. Что является основным элементом презентации?
3. Как просмотреть презентацию?
4. Как просмотреть анимация на нескольких слайда?

## Модели решения функциональных и вычислительных задач

### Лекция 4.1. Моделирование как метод познания

#### 4.1.1. Классификация задач, решаемых с помощью моделей

Задачи, которые решает человек можно разделить на две категории: вычислительные, целью которых является определение некоторой величины, и функциональные, предназначенные для создания некоего аппарата, выполняющего определенные действия – функции. Например, проектирование нового здания требует решения задачи расчета прочности фундамента, несущих опорных конструкций, расчета финансовых затрат, определение оптимального числа работников и т.д. Для повышения производительности труда строителей создано немало машин функционального назначения (т.е. решены функциональные задачи), такие, как экскаватор, бульдозер, подъемный кран и т.д.

С точки зрения информатики, решение любой задачи представляют замкнутую технологическую последовательность (рис. 4.1.):

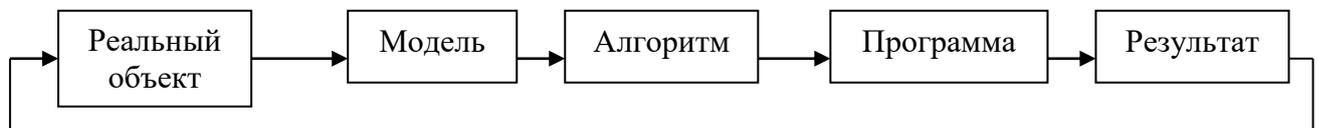


Рис. 4.1. Этапы решения задач

#### 4.1.2. Основные понятия

Каждый элемент в этом ряду играет свою особую роль.

Объектом (от латинского *Objectum* – предмет) называется все то, что противостоит субъекту в его практической и познавательной деятельности, на что направлена эта деятельность. Под объектами понимаются предметы и явления, как доступные, так и недоступные чувственному восприятию человека, но имеющие видимое влияние на другие объекты (например, гравитация, инфразвук или электромагнитные волны). Объект всегда должен рассматриваться во взаимодействии с другими объектами, с учетом их взаимодействия. Деятельность человека обычно идет по двум направлениям: исследование свойств объекта с целью их использования (или нейтрализации); создание новых объектов, имеющих полезные свойства. Первое направление относится к научным исследованиям и большую роль при их проведении имеет *гипотеза*, т.е. предсказание свойств объекта при недостаточной его изученности. Второе относится к инженерному проектированию. При этом важную роль играет понятие *аналогии* – суждения о каком-либо сходстве известного и проектируемого объекта. Аналогия может быть полной или частичной. Это понятие относительно и определяется уровнем абстрагирования и целью построения аналогии. Любой аналог (образ) какого-либо объекта, процесса или явления, используемый в качестве заместителя (представителя) оригинала, называется *моделью* (от латинского *modulus* – образец).

Исследование объектов, процессов или явлений путем построения и изучения их моделей для определения или уточнения характеристик оригинала называется *моделированием*. Можно сказать, что моделирование – это представление объекта путем проведения экспериментов с его моделью. Теория замещения объектов-оригиналов объектом-моделью называется *теорией моделирования*.

Если результаты моделирования подтверждаются и могут служить основой для прогнозирования поведения исследуемых объектов, то говорят, что модель *адекватна* объекту. Степень адекватности зависит от цели и критериев моделирования.

Теория моделирования рассматривает две группы способов моделирования: аналитическое и имитационное. Аналитическое моделирование заключается в построении модели, основанной на описании поведения объекта или системы объектов в виде аналитических выражений – формул. При таком моделировании объект описывается системой линейных или нелинейных алгебраических или дифференциальных уравнений, решение которых может дать представление о свойствах объекта. К полученной аналитической модели, с учетом вида и сложности формул применяются аналитические или приближенные численные методы. Численные методы обычно реализуются на вычислительных машинах. Применение аналитического моделирования ограничено сложностью получения и анализа выражений для большинства систем.

Имитационное моделирование предполагает построение модели с характеристиками, адекватными оригиналу, на основе какого-либо его физического или информационного принципа. Это означает, что внешние воздействия на модель и объект вызывают идентичные изменения свойств оригинала и модели. При таком моделировании отсутствует общая аналитическая модель большой размерности, а объект представлен системой элементов, взаимодействующих между собой и с внешним миром. Задавая внешние воздействия, можно получить характеристики системы и провести их анализ. В последнее время имитационное моделирование все больше реализуется моделированием объектов на компьютере, что позволяет в интерактивном режиме исследовать модели самых разных по природе объектов.

#### **4.1.3. Цели моделирования**

Практически во всех науках о природе, живой и неживой, об обществе, построение и использование моделей является мощным орудием познания. Реальные объекты и процессы бывают столь многогранны и сложны, что лучшим способом их изучения часто является построение модели, отображающей лишь какую-то грань реальности и потому многократно более простой, чем эта реальность, и исследование вначале этой модели. Модели используются для решения всевозможных задач. Из этого множества можно выделить основные цели использования моделей:

- 1) понять, как устроен конкретный объект, какова его структура, основные свойства, законы развития и взаимодействия с окружающим миром (**понимание**);
- 2) научиться управлять объектом (или процессом) и определять наилучшие способы управления при заданных целях и критериях (**управление**);
- 3) прогнозировать прямые и косвенные последствия реализации заданных способов и форм воздействия на объект (**прогнозирование**).

### **Лекция 4.2. Методы и технологии моделирования**

#### **4.1.2. Системный подход в моделировании систем**

*Классический* (или индуктивный) *подход* к моделированию рассматривает систему, переходя от частного к общему, и синтезирует ее путем слияния компонент, разрабатываемых отдельно. *Системный подход* предполагает последовательный переход от общего к частному, когда в основе рассмотрения лежит цель, при этом объект выделяется из окружающего мира.

При создании нового объекта с полезными свойствами задаются критерии, определяющие степень полезности полученных свойств. Так как любой объект моделирования представляет собой систему взаимосвязанных элементов, введено понятие системы. Система  $S$  – есть целенаправленное множество взаимосвязанных элементов любой природы. Внешняя среда  $E$  представляет собой множество существующих вне системы элементов любой природы, оказывающих влияние на систему или находящихся под ее воздействием.

При системном моделировании, прежде всего, четко определяется цель моделирования. Создание модели полного аналога оригинала дело трудоемкое и дорогое, поэтому модель создается под определенную цель.

Важным для системного подхода является определение *структуры системы* - совокупности связей между элементами системы, отражающих их взаимодействие. Существует ряд подходов к исследованию системы и ее свойств, к которым относятся структурный и функциональный. При структурном, выявляется состав выделенных элементов системы  $S$  и связи между ними. Совокупность элементов и связей позволяет судить о свойствах выделенной части системы. При функциональном подходе рассматриваются функции (алгоритмы) поведения системы, причем, каждая функция описывает поведение одного свойства при внешнем воздействии  $E$ . Такой подход не требует знания структуры системы, а ее описание состоит из набора функций ее реакции на внешние воздействия. Классический метод построения модели использует функциональный подход. В качестве элемента модели принимается компонента, описывающая поведение одного свойства и не отображающая реальный состав элементов. Компоненты изолированы друг от друга, что плохо отражает моделируемую систему. Такой метод построения модели применим лишь для простых систем, т.к. требует включения в состав функций, описывающих свойства системы, отношения между свойствами, которые могут быть плохо определены или неизвестны.

С усложнением моделируемых систем, когда невозможно учесть все взаимовлияния свойств, применяется системный метод, основанный на структурном подходе. При этом система  $S$  разбивается на ряд подсистем  $S_i$  со своими свойствами, которые, проще описать функциональными зависимостями, и определяются связи между подсистемами. В этом случае система функционирует в соответствии со свойствами отдельных подсистем и связей между ними. Это избавляет от необходимости описывать функционально взаимосвязи между свойствами системы  $S$ , что делает модель более гибкой, т.к. изменение свойств одной из подсистем автоматически изменяет свойства системы.

### **Лекция 4.3. Классификация моделей**

В зависимости от характера изучаемых процессов в системе  $S$  и цели моделирования существует множество типов моделей и способов их классификации, например, по цели использования, наличию случайных воздействий отношению ко времени, возможности реализации, области применения и др.

#### **4.3.1 Классификация видов моделирования**

<b>По цели использования</b>	<b>По наличию воздействий на систему</b>	<b>По отношению ко времени</b>	<b>По возможности реализации</b>	<b>По области применения</b>
Научный эксперимент.	Детерминированные.	Статические.	Мысленные (наглядные, символические, математические).	Универсальные.
Комплексные испытания и производственный эксперимент.	Стохастические.	Динамические (дискретные, непрерывные).	Реальные (натурные, физические).	Специализированные.
Оптимизационные модели.			Информационные.	

### **По цели использования:**

*В научном эксперименте* осуществляется исследование модели с применением различных средств получения данных об объекте, возможности влияния на ход процесса, с целью получения новых данных об объекте, процессе или явлении;

*Комплексные испытания и производственный эксперимент* используют натурное испытание физического объекта для получения высокой достоверности о его характеристиках;

*Оптимизационные* связаны с нахождением оптимальных показателей системы (например, нахождение минимальных затрат или определение максимальной прибыли).

По наличию воздействий на систему:

*Детерминированные* – в системах отсутствуют случайные воздействия;

*Стохастические* – в системах присутствуют вероятностные воздействия.

Эти же модели классифицируют по способу оценки параметров системы: в *детерминированных* системах параметры модели оцениваются одним показателем для конкретных значений их исходных данных; в *стохастических системах* наличие вероятностных характеристик исходных данных, позволяет оценивать параметры системы несколькими показателями.

### **По отношению ко времени:**

*Статические* модели описывают систему в определенный момент времени;

*Динамические* модели рассматривают поведение системы во времени. Они подразделяются на *дискретные*, в которых все события происходят по интервалам времени, и *непрерывные*, где все события происходят непрерывно во времени.

### **По возможности реализации:**

*Мысленные* модели описывают систему, которую трудно или невозможно моделировать реально. Мысленные модели разделяются на:

Наглядные - при которых моделируемые процессы и явления протекают наглядно;

Символические – модель системы представляет логический объект, в котором основные свойства и отношения реального объекта выражены системой знаков и символов;

Математические – представляют системы математических объектов, позволяющих получать исследуемые характеристики реального объекта.

*Реальные* модели системы представлены либо реальным объектом, либо его частью. Эти модели делятся на:

натурные – проведение исследования на реальном объекте и последующая обработка результатов эксперимента с применением теории подобия;

физические – проведение исследования на установках, которые сохраняют природу явления и обладают физическим подобием.

*Информационные* модели реализуют информационные процессы (возникновение, передачу, обработку и использование информации) на компьютере.

### **По области применения:**

Универсальные модели предназначены для использования многими системами;

Специализированные модели созданные для исследования конкретной системы.

Вопросы для самоконтроля

1. Укажите типы задач, которые решаются с помощью компьютера.
2. Что называется моделированием?
3. Что отражает модель объекта?
4. Для чего может быть построена модель?
5. К какому типу моделей можно отнести тренажеры?
6. Сколько моделей может иметь один прототип?
7. Какие модели называются стохастическими?

8. Какие модели называются детерминированными?
9. Перечислите последовательность этапов решения любой задачи.

## **Лекция 4.4. Интеллектуальные системы**

### **4.4.1. Искусственный интеллект**

Идеи моделирования человеческого разума известны с древнейших времен. Впервые об этом упоминается в сочинении философа и теолога Раймунда Луллия (около 1235 – 1315) «Великое искусство», который не только высказал идею логической машины для решения разнообразных задач, но и попытался ее реализовать. Рене Декарт (1596 – 1650) и Готфрид Вильгельм Лейбниц (1646 – 1716) независимо друг от друга развивали учение о прирожденной способности ума к познанию и всеобщих и необходимых истин логики и математики, работали над созданием универсального языка классификации всех знаний. Именно на этих идеях базируются теоретические основы создания искусственного интеллекта. Толчком к дальнейшему развитию модели человеческого мышления стало появление в 40-х гг. XX в. ЭВМ. В 1948 г. американский ученый *Норберт Винер* (1894 – 1964) сформулировал основные положения новой науки – кибернетики. В 1956 г. признано новое научное направление, связанное с машинным моделированием человеческих, интеллектуальных функций, и названное *искусственным интеллектом*.

Вскоре эта отрасль разделилась на два направления: нейрокибернетику и кибернетику «черного ящика».

*Нейрокибернетика* обратилась к структуре человеческого мозга, как единственно мыслящему объекту и занялась его аппаратным моделированием. Нейрокибернетика занимается созданием элементов, аналогичных нейронам (связанным друг с другом нервным клеткам, составляющим основу мозга), и их объединением в функционирующие системы, названные *нейросетями*. В середине 80-х гг. XX в. в Японии был создан первый нейрокомпьютер, моделирующий структуру человеческого мозга. Его основная область применения – *распознавание образов*.

Для кибернетики «черного ящика» структура модели не важна, важна ее реакция на заданные входные данные. На выходе модель должна реагировать как человеческий мозг. Ученые этого направления занимаются разработкой алгоритмов решения интеллектуальных задач для имеющихся вычислительных систем. Наиболее значимые результаты:

Модель лабиринтного поиска (конец 50-х гг.), в которой рассматривается граф состояний объекта и в нем происходит поиск оптимального пути от входных данных к результирующим. На практике эта модель не нашла широкого применения.

Эвристическое программирование (начало 60-х гг.) разрабатывало стратегии действий на основе заранее известных заданных правил (эвристик). Эвристика – теоретически не обоснованное правило, позволяющее уменьшить количество переборов в поиске оптимального пути.

*Методы математической логики.* Метод резолюций, позволяющий на основе определенных аксиом автоматически доказывать теоремы. В 1973 г. создан язык логического программирования *Пролог*, позволяющий обрабатывать символьную информацию.

С середины 70-х гг. реализуется идея моделирования конкретных знаний специалистов-экспертов. В США появляются первые экспертные системы. Возникает новая технология искусственного интеллекта, основанная на представлении и использовании знаний. С середины 80-х гг. искусственный интеллект коммерциализируется. Растут капиталовложения в эту отрасль, появляются промышленные системы, повышается интерес к самообучающимся системам.

#### 4.4.2. База знаний

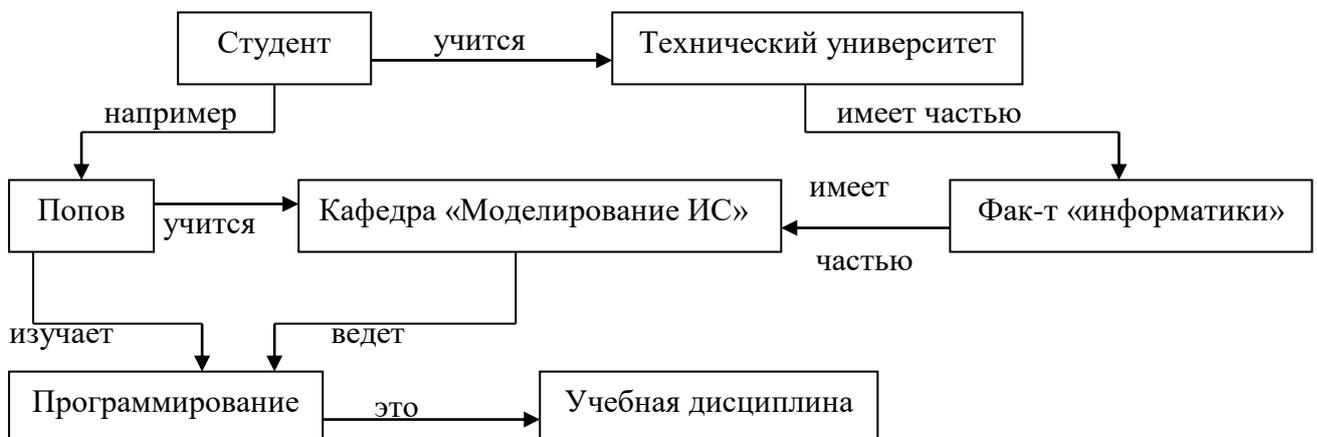
База знаний – основа любой интеллектуальной системы. В ней хранятся знания, представляющие собой выявленные закономерности предметной области (принципы, связи, законы), позволяющие решать задачи в этой предметной области. Знания – хорошо структурированные данные, или данные о данных (метаданные). С точки зрения искусственного интеллекта знания определяют как формализованную информацию, на которую ссылаются в процессе логического вывода. Существует множество моделей представления знаний, но большинство из них относятся к следующим классам: логические модели; продукционные модели; семантические сети; фреймовые модели.

*Формальные логические модели* основаны на классическом исчислении предикатов первого порядка, когда предметная область описывается в виде набора аксиом. Вся информация, необходимая для решения задач, рассматривается как совокупность правил и утверждений, которые представляются как формулы в некоторой логике предикатов. Знания отражают совокупность таких формул, а получение новых знаний сводится к реализации процедур логического вывода. Эта логическая модель применима в основном в исследовательских «идеальных» системах, так как предъявляет высокие требования и ограничения предметной области. В промышленных экспертных системах используются ее различные модификации и расширения.

Исследования процессов принятия решения человеком показали, что рассуждая и принимая решение, человек использует *продукционные правила* (от английского production – правило вывода, порождающее правило). Продукционная модель, основанная на правилах, позволяет представить знания в виде предложений: ЕСЛИ (список условий), ТО (следует выполнить перечень действий). *Условие* – это предложение, по которому происходит поиск в базе знаний, а *действие* есть некоторая операция, выполняемая при успешно осуществленном поиске. Действия могут быть как *промежуточными*, выступающими далее как условия, так и *целевыми*, завершающими работу ИС. В продукционной модели база знаний состоит из совокупности правил. Программа, управляющая перебором правил, называется *машиной вывода*. Механизм выводов связывает знания и создает из их последовательности заключение. Вывод бывает прямой (метод сопоставления, от данных к поиску цели) или обратный (метод генерации гипотезы и ее проверки, от цели к данным).

Продукционная модель наглядная, модульная, легко вносятся дополнения и изменения, имеет простой механизм логического вывода, чаще всего используется в промышленных экспертных системах.

*Семантика* – это наука, исследующая свойства знаков и знаковых систем, их смысловую связь с реальными объектами. *Семантическая сеть* – это ориентированный граф, вершины которого есть понятия, а дуги – отношения между ними (рис. 4.4.1).



### Рис. 4.4.1. Семантическая сеть

Достоинствами сетевых моделей являются: большие выразительные возможности; наглядность системы знаний, представленной графически; близость структуры сети, представляющей систему знаний, семантической структуре фраз на естественном языке; соответствие современным представлениям об организации долговременной памяти человека. К недостаткам можно отнести то, что сетевая модель не содержит ясного представления о структуре предметной области, которая ей соответствует, поэтому ее формирование и модификация затруднительны; сетевые модели представляют собой пассивные структуры, для их обработки используется специальный аппарат *формального вывода*. Проблема поиска решения в базе знаний типа семантической сети сводится к задаче поиска фрагмента сети, соответствующего некоторой подсети поставленной задачи, что, в свою очередь, говорит еще об одном недостатке модели – сложности поиска вывода на семантических сетях.

Сетевые модели являются наглядным и достаточно универсальным средством представления знаний. Однако их формализация в конкретных моделях представления, использования и модификации знаний представляет достаточно трудоемкий процесс, особенно при наличии множественных отношений между понятиями.

Термин *фрейм* (от англ. frame – каркас, рамка) предложен для обозначения структуры единицы знаний, которую можно описать некоторой совокупностью понятий, для ее пространственного восприятия. Фрейм имеет определенную внутреннюю структуру, состоящую из совокупности элементов, называемых *слотами*. Каждый слот, в свою очередь, представляется определенной структурой данных, процедурой, или может быть связан с другим фреймом. Фреймовая модель представляет собой систематизированную в виде единой теории технологическую модель памяти человека и его сознания. В отличие от других моделей во фреймах фиксируется жесткая структура.

Важным свойством фреймов является наследование свойств, заимствованных из теории семантических сетей. Наследование происходит по АКО – связям (*A-Kind-Of=это*). Слот АКО указывает на фрейм более высокого уровня иерархии, откуда неявно наследуется, т.е. переносятся значения аналогичных слотов. Например, в сети фреймов на рис. 4.4.2. «конструктор» наследует свойства фреймов «инженер» и «человек», которые стоят на более высоком уровне иерархии. Модель фреймов достаточно универсальна, позволяет отобразить все многообразие знаний о мире через:

- Фреймы – структуры, для обозначения объектов и понятий (лекция, конспект, кафедра);
- Фреймы – роли (студент, преподаватель, декан);
- Фреймы – сценарии (сдача экзамена, празднование именин, получение стипендии);
- Фреймы – ситуации (тревога, рабочий режим учебного дня) и др.

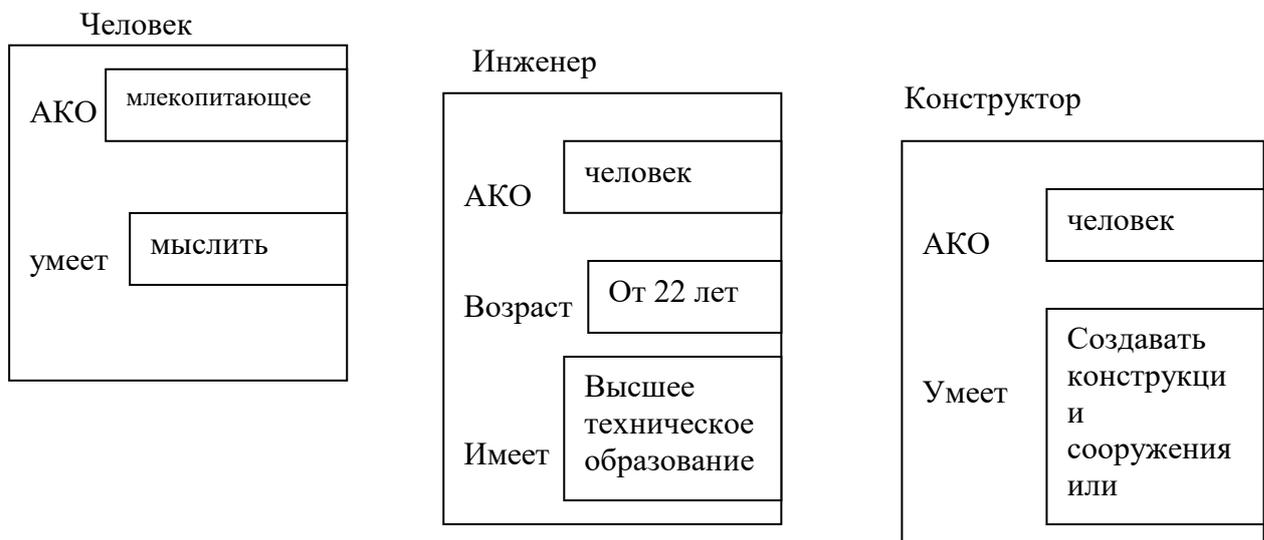




Рис. 4.4.2. Сеть фреймов

Основным преимуществом фреймов как модели представления знаний является их способность отражать концептуальную основу организации памяти человека, а также гибкость и наглядность.

Обобщая анализ моделей представления знаний, можно сделать следующие выводы:

- Нельзя дать универсальных рекомендаций по выбору модели. Этот выбор определяется возможностью и удобством представления исследуемой предметной области с учетом необходимости использования знаний.
- Наиболее мощными являются смешанные модели представления знаний.

#### 4.4.3. Экспертные системы

Экспертные системы предназначены для анализа данных, содержащихся в базах знаний, и выдачи рекомендаций по запросу пользователя. Используются в тех случаях, когда исходные данные хорошо формализуются, но для принятия решения требуются специальные, обширные знания. Экспертные системы – это сложные программные комплексы, аккумулирующие знания специалистов в конкретных предметных областях и тиражирующие этот эмпирический опыт для консультаций менее квалифицированных пользователей.

Предметные области: медицина, фармакология, химия, геология, экономика, юриспруденция и др., в которых большая часть знаний является личным опытом специалистов высокого уровня (экспертов), нуждаются в экспертных системах. Те области, где большая часть знаний представлена в виде коллективного опыта (например, высшая математика), не нуждаются в них.

Современные ЭС способны:

- По совокупности признаков заболевания установить диагноз, назначить лечение, дозировать медикаменты, выработать программу курса лечения;
- Выполнять задачи диагностических систем в исследовании явлений и процессов (например, для анализа крови; управления производством; изучения состояния недр земли, нефтяных полей, залежей угля и т.п.);
- Распознавать речь, на данном этапе в ограниченной области применения;
- Распознавать человеческие лица, отпечатки пальцев и др.

На рис. 4.4.3. изображены основные компоненты модели экспертной системы:

Интерфейс пользователя
---------------------------

Подсистема разъяснений

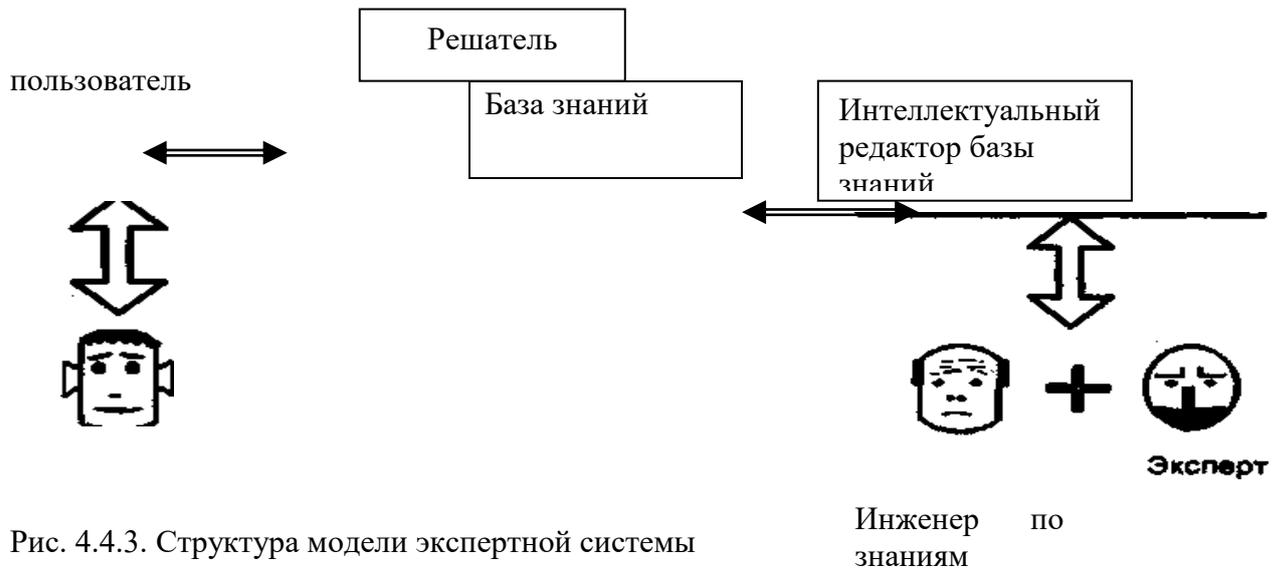


Рис. 4.4.3. Структура модели экспертной системы

Характерной особенностью любой экспертной системы является способность к саморазвитию. Исходные данные хранятся в базе знаний в виде фактов, между которыми установлены определенные логические связи. Если при тестировании выявлены некорректные рекомендации или заключения по конкретным вопросам, либо заключение не может быть сформулировано, это означает, или отсутствие важных фактов ее базе, или нарушения в логической системе связи. В любом случае система сама может сгенерировать достаточный набор вопросов к эксперту и автоматически повысить свое качество.

#### Вопросы для самоконтроля

1. Перечислите основные направления искусственного интеллекта.
2. Что такое «нейросет»?
3. Где применяются нейрокомпьютеры?
4. Что такое «эвристика»?
5. Что является основой любой интеллектуальной системы?
6. Из чего состоит семантическая сеть?
7. Какие бывают фреймы?
8. Что такое АКО - связь?
9. Что такое наследование свойств?
10. Что входит в состав экспертной системы?
11. Кто такой инженер по знаниям?

## Алгоритмизация и программирование. Языки программирования высокого уровня

### Лекция 5.1. Алгоритмизация

#### 5.1.1. Понятие алгоритма

Слово «алгоритм» происходит от «algorithmi» – латинской формы описания имени великого узбекского математика IX в. аль - Хорезми, который сформулировал правила выполнения арифметических действий. Первоначально под алгоритмом и понимали только правила выполнения четырех арифметических действий над многозначными числами.

Понятие алгоритма – одно из фундаментальных понятий информатики.

Алгоритм – это формальное описание способа решения задачи, путем разбиения ее на конечную по времени последовательность действий (элементарных операций). Под словом «формальное» подразумевается, что описание должно быть абсолютно полным и учитывать

все возможные ситуации, которые могут встретиться по ходу решения. Под элементарной операцией понимается действие, которое по определенным критериям (например, очевидности) не имеет смысла детализировать.

Алгоритм на выбранном языке программирования записывается с помощью команд описания данных, вычисления значений и управления последовательностью выполнения программ.

### 5.1.2. Свойства алгоритма

Алгоритм должен быть составлен таким образом, чтобы исполнитель, в расчете на которого он создан, мог однозначно и точно следовать командам алгоритма и эффективно получать определенный результат. Это накладывает на записи алгоритма ряд обязательных требований, суть которых вытекает из приведенного выше неформального толкования понятия алгоритма. Сформулируем эти требования в виде перечня свойств, которым должны удовлетворять алгоритмы.

*Дискретность* (разрывность) – это свойство алгоритма, характеризующее его структуру: каждый алгоритм состоит из отдельных законченных действий. Говорят: «Делится на шаги».

*Массовость* – применимость алгоритма ко всем задачам рассматриваемого типа, при любых исходных данных. Например, алгоритм решения квадратного уравнения в области действительных чисел должен содержать все возможные исходы решения, т.е., рассмотрев значения дискриминанта, алгоритм находит либо два различных корня, либо два равных, либо делает вывод о том, что действительных корней нет.

*Определенность* (детерминированность, точность) – свойство алгоритма, указывающее на то, что каждый шаг алгоритма должен быть строго определен, и не допускать различных толкований; также строго должен быть определен порядок выполнения отдельных шагов. В алгоритмах недопустимы ситуации, когда после выполнения очередной команды исполнителю неясно, какая из команд алгоритма должна выполняться на следующем шаге.

*Результативность* – свойство, состоящее в том, что любой алгоритм должен завершаться за конечное число шагов. Вывод о том, что решения не существует – тоже результат. Вопрос о рассмотрении бесконечных алгоритмов остается за рамками теории алгоритмов.

*Формальность* – это свойство указывает на то, что любой исполнитель, способный воспринимать и выполнять инструкции алгоритма, действует формально, т.е. отвлекается от содержания поставленной задачи и лишь строго выполняет инструкции. Рассуждать «что, как и почему?» должен разработчик алгоритма, а исполнитель формально (не думая) поочередно исполняет предложенные команды и получает необходимый результат.

### 5.1.3. Способы описания алгоритмов

Существуют несколько способов описания алгоритма: словесное, псевдокод, блок-схема, программа.

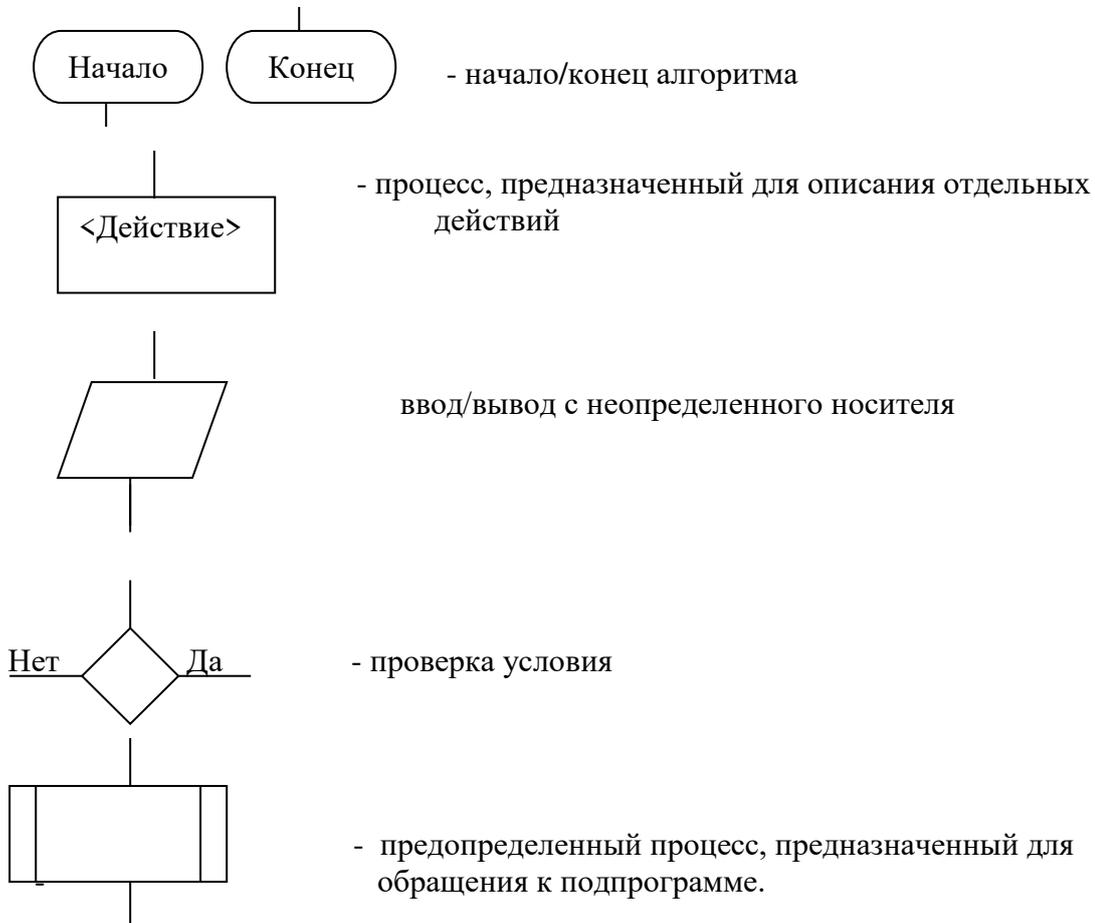
*Словесное* описание представляет структуру алгоритма на естественном языке. Запись алгоритма осуществляется в произвольной форме, никаких правил не существует.

*Псевдокод* – описание структуры алгоритма на естественном, частично формализованном языке, позволяющее выявить основные этапы решения задачи, перед точной его записью на языке программирования. В псевдокоде используются некоторые формальные конструкции и общепринятая математическая символика.

Единого или формального определения псевдокода не существует, поэтому возможны различные псевдокоды, отличающиеся набором используемых слов и конструкций.

Блок-схема – описание структуры алгоритма с помощью геометрических фигур с линиями – связями, показывающими порядок выполнения отдельных инструкций. В блок – схеме каждой формальной конструкции соответствует определенная геометрическая фигура или связанная линиями совокупность фигур.

Основные конструкции, используемые для построения блок – схем.

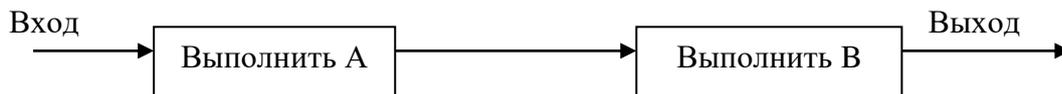


## Лекция 5.2. Блок-схемы алгоритма

### 5.2.1. Алгоритмы решения задач

Логическая структура алгоритма решения любой задачи может быть выражена комбинацией трех базовых структур: следования, ветвления и цикла (это содержание теоремы Бема – Якопини).

*Линейная структура (следование)* самая важная из структур. Она означает, что действия могут быть выполнены друг за другом (рис. 5.2.1.).



Прямоугольники могут представлять как одну единственную команду, так и множество операторов, необходимых для выполнения сложной обработки данных.

#### Пример 5.2.1.

Опишем алгоритм сложения двух чисел на псевдокоде и в виде блок-схемы (рис. 5.2.2.).

1. Псевдокод:
2. Ввод двух чисел a, b

3. Вычисляем сумму  $S = a + b$
4. Вывод  $S$
5. Конец.

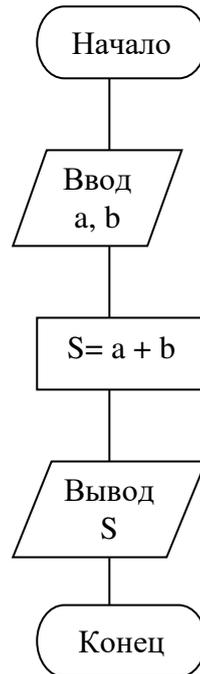


Рис. 5.2.2. Блок - схема к примеру 5.2.1.

*Ветвление* (развилка) – это структура, обеспечивающая выбор между двумя альтернативами. Выполняется проверка условия, а затем выбирается один из путей (рис. 5.2.3).

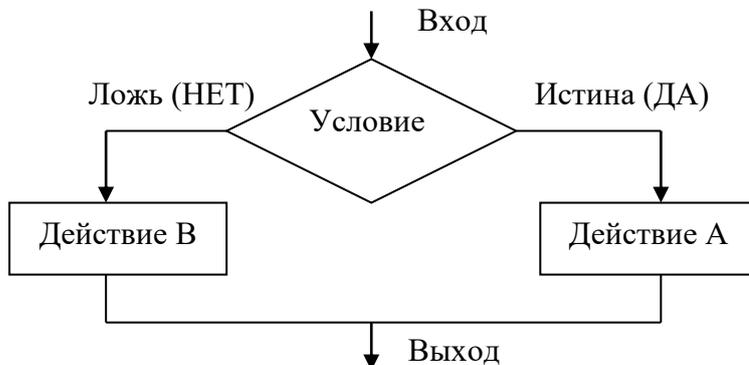
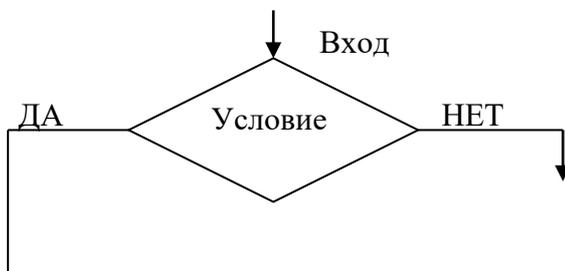


Рис. 5.2.3. Полное ветвление

Если условие имеет значение «Истина», то выполняется «Действие А». Если условие имеет значение «Ложь», выполняется «Действие В». Эта структура называется, также «Если – ТО – ИНАЧЕ» или «развилка». Каждый путь (ТО или ИНАЧЕ) ведет к общей точке слияния, так что выполнение алгоритма продолжается независимо от того, какой путь был выбран. Может оказаться, что для одного из результатов проверки ничего выполнять не надо. В этом случае можно применить только один обрабатывающий блок (рис. 5.2.4).



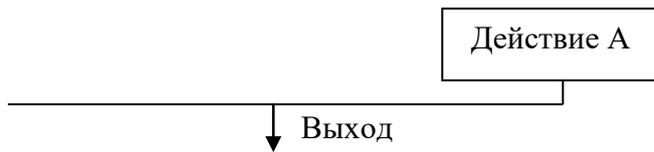


Рис. 5.2.4. Структура «неполное ветвление»

Такая структура называется «неполным ветвлением» или «неполной развилкой».

**Пример 5.2.2.**

Вывести значение наибольшего числа из двух чисел (рис. 5.2.5).

Псевдокод:

1. Ввод двух чисел a, b
2. ЕСЛИ a>b, ТО «выводим a»,  
    ИНАЧЕ «выводим b»
3. Конец.

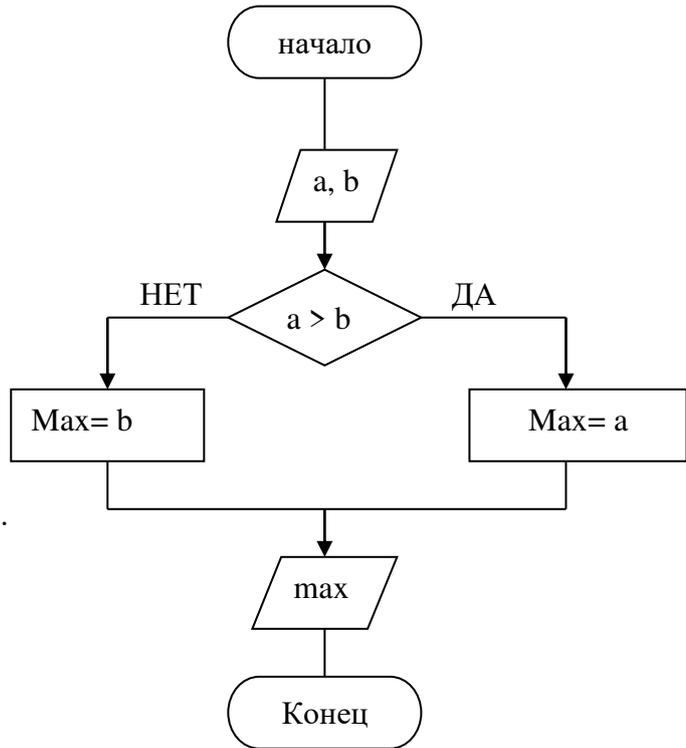


Рис. 5.2.5. Блок – схема к примеру 5.2.2.

*Цикл* (или повторение) предусматривает повторное выполнение некоторого набора команд алгоритма. Циклы позволяют записать длинные последовательности операций обработки данных с помощью небольшого числа повторяющихся команд. Различают два типа циклов: «цикл с предусловием» и «цикл с постусловием».

*Цикл с предусловием* («Пока») (рис. 5.2.6).



Рис. 5.2.6. Структура цикла «Пока».

Цикл начинается с проверки логического выражения. Если оно истинно, то выполняется тело цикла, затем все повторяется, пока логическое выражение сохраняет значение «истина». Как только оно становится ложным, выполнение операций прекращается и управление передается дальше. Особенностью цикла с предусловием является то, что если изначально логическое выражение имеет значение «ложь», то тело цикла не выполнится ни разу.

Пример 5.2.3.

Вычислить сумму 100 чисел (рис. 5.2.7).

Псевдокод:

1. НАЧ
2.  $I = 1; S = 0$
3. ПОКА  $i \leq 100$  делать  
НЦ
4. Ввести  $a_i$
5.  $S = S + a_i$
6.  $i = i + 1$   
КЦ
7. Вывод  $S$
8. Конеч.

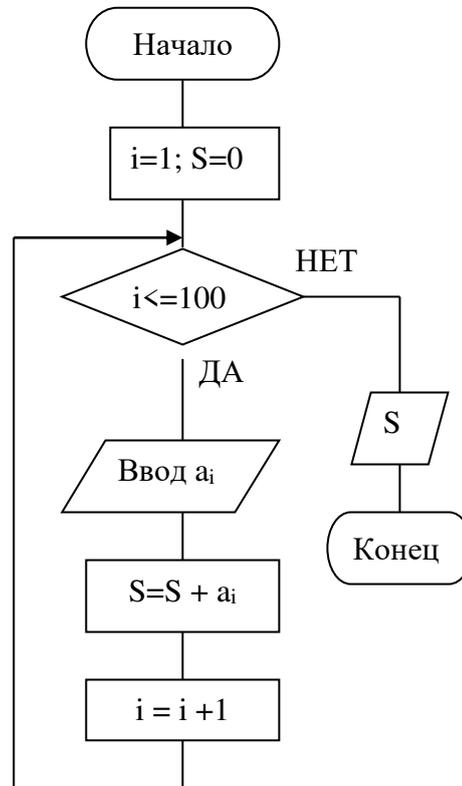


Рис. 5.2.7. Блок – схема к примеру 5.2.3 с циклом «Пока»

*Цикл с постусловием («До»).*

В отличие от цикла с предусловием, тело цикла с постусловием всегда будет выполнено хотя бы один раз, после чего проверяется условие. В этой конструкции тело цикла будет повторяться до тех пор, пока значение логического выражения ложно. Как только оно становится истинным, выполнение тела цикла прекращается (рис. 5.2.8).

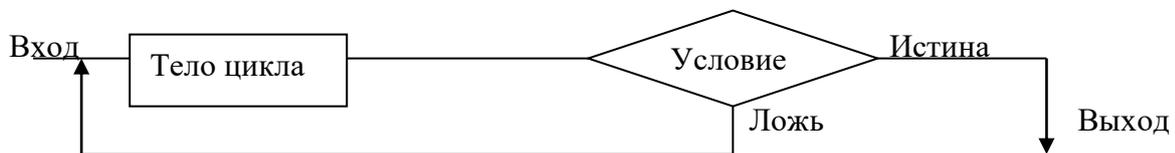


Рис. 5.2.8. Структура «цикла с постусловием».

#### Пример 5.2.4.

Вывести максимальное значение из 100 натуральных чисел (рис. 5.2.9).

Псевдокод:

1. Начало
2. Ввести  $a_1$
3.  $\max = a_1; i = 2$
4. НЦ

- a. Ввести  $a_i$
- b. ЕСЛИ  $\max < a_i$  ТО  $\max = a_i$
- c.  $i = i + 1$
5. ДО  $i > 100$ ;
6. КЦ
7. Вывести  $\max$ .
8. Конец.

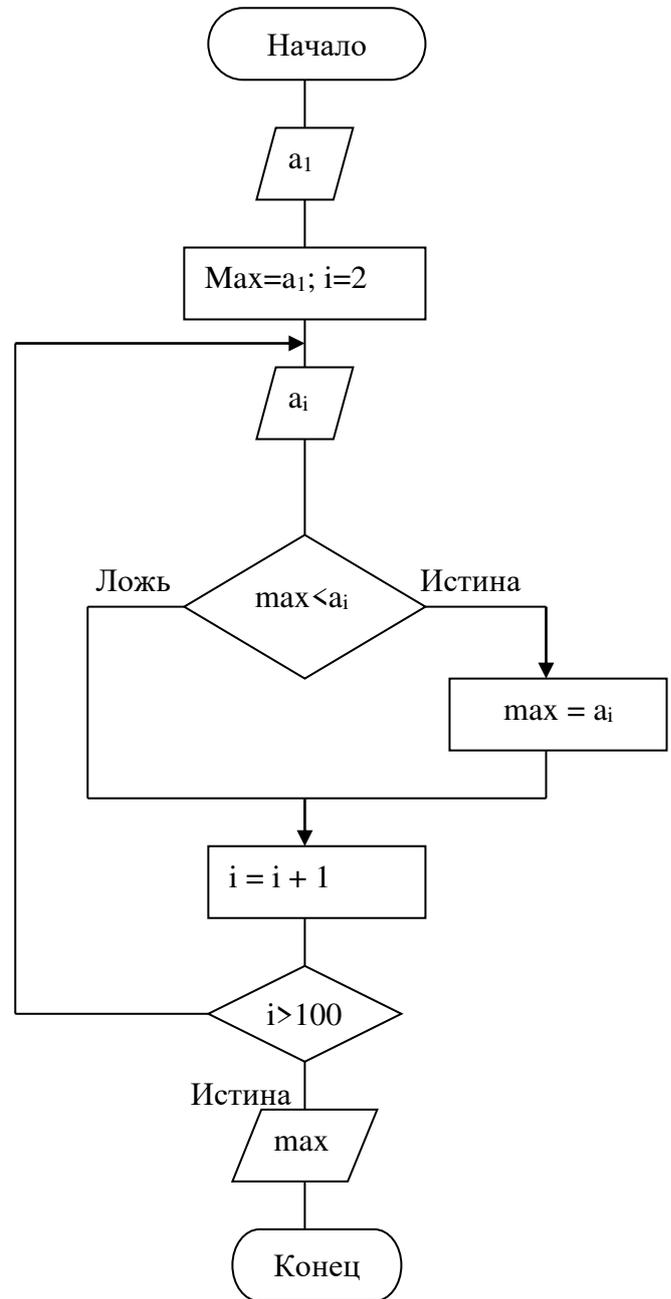


Рис. 5.2.9. Блок – схема к примеру 5.2.4. с циклом «До»

Базовые алгоритмические структуры можно комбинировать одну с другой – как путем организации их следования, так и путем создания суперпозиций (вложений одной структуры в другую). Используя описанные структуры, можно полностью исключить использование

каких-либо еще операторов условного и безусловного перехода, что является важным признаком структурного программирования. Приведем несколько примеров (рис. 5.2.10, 5.2.11, 5.2.12, 5.2.13).

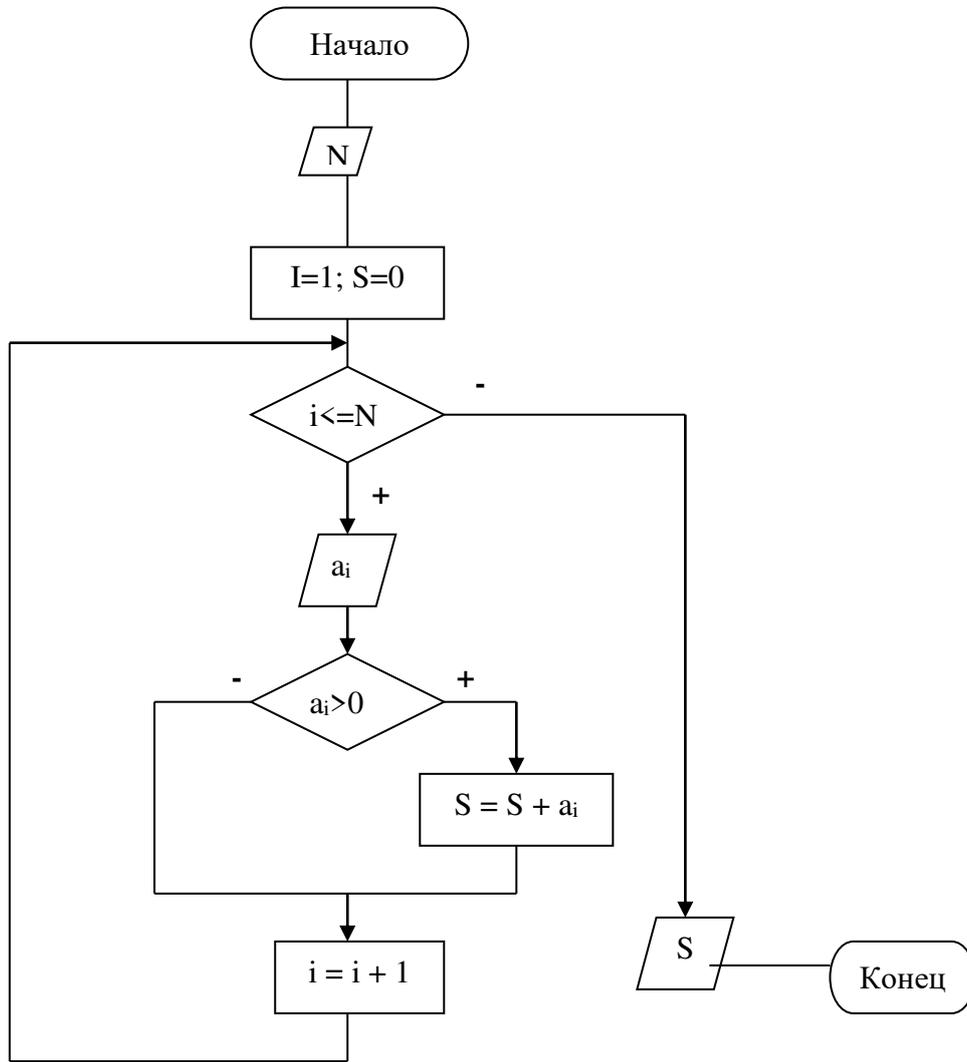


Рис. 5.2.10. Алгоритм типа «развилка, вложенная в цикл, с предусловием», для нахождения суммы положительных чисел и N возможных

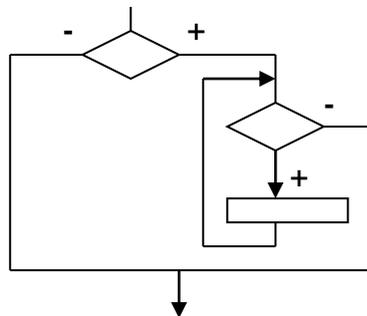


Рис. 5.2.11. Алгоритм типа «цикл, с предусловием, вложенный в неполную развилку»

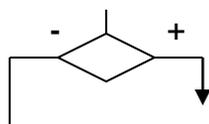
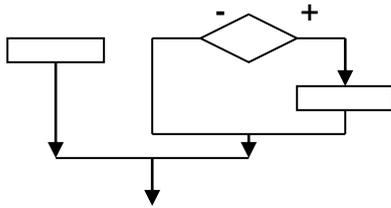


Рис. 5.2.12. Алгоритм типа «неполная развилка, вложенная в пол-



ную развилку».

### Вопросы для самоконтроля

1. Дайте определение алгоритма и поясните его.
2. Какие формы представления алгоритма вы знаете?
3. В чем особенности графического представления алгоритма?
4. Назовите основные (базовые) алгоритмические структуры?
5. Перечислите свойства алгоритмов и объясните, чем они определены.

## Лекция 5.3. Программирование

### 5.3.1. Языки программирования

Языки программирования - это искусственные языки. От естественных они отличаются ограниченным числом «слов», значение которых понятно транслятору, и очень строгими правилами записи команд (операторов).

*Транслятор*-это специальное служебное приложение, которое либо переводит текст программы в машинный код, либо исполняет его.

С помощью языка программирования создается не готовая программа, а только ее текст, описывающий ранее разработанный алгоритм. Чтобы получить работающую программу, надо этот текст либо автоматически перевести в машинный код (для этого служат *программы - компиляторы*) и затем использовать отдельно от исходного текста, либо сразу выполнять команды языка, указанные в тексте программы (этим занимаются *программы-интерпретаторы*). Компиляторы полностью обрабатывают текст программы (исходный код). Они просматривают его в поисках синтаксических ошибок, выполняют определенный смысловой анализ и затем автоматический перевод (транслируют) на машинный язык - генерируют машинный код.

Интерпретатор моделирует некую виртуальную вычислительную машину, для которой базовыми инструкциями служат не элементарные команды процессора, а операторы языка программирования.

### 5.3.2. Уровни языков программирования

Разные типы процессов имеют разные наборы команд. Если язык программирования ориентирован на конкретный тип процессора и учитывает его особенности, то он называется *языком программирования низкого уровня*. В данном случае «низкий» не значит «плохой». Имеется ввиду, что операторы языка близки к машинному коду и ориентированы на конкретные команды процессора. Языком 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 называют транслитерацией. Так как наборы инструкций для каждой модели процессора отличаются, конкретной компьютерной архитектуре соответствует свой язык ассемблера, 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 устройства 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, быстродействие 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 изображения.

*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. Разрабатывать программы на языках высокого уровня с помощью понятных и мощных команд значительно проще, а ошибок при создании программы допускается гораздо меньше.

### 5.3.3. Поколение языков программирования

Языки программирования принято делить на пять поколений. В первое 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 принято относить к 60-м годам. В это время родились универсальные языки высокого уровня, с их помощью удается решать задачи из любых областей. Такие качества новых языков, как относительная простота, независимость от конкретного компьютера, позволили резко повысить производительность труда программиста. Благодаря понятной структуре этих языков, значительное число специалистов некомпьютерных областей могли написать небольшие программы. Подавляющее большинство языков этого поколения успешно применяется и сегодня.

С начала 70-х годов по настоящее время продолжается период языков 4-го 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 80-х годов. К ним относятся также системы автоматического создания прикладных программ с помощью визуальных средств разработки, без знания программирования. Главная идея заключается в возможности автоматического формирования результирующего текста на универсальных языках программирования. Инструкции же вводятся в компьютер в максимально наглядном виде с помощью методов, наиболее удобных для человека, незнакомого с программированием.

#### **Вопросы для самоконтроля**

1. Что такое язык программирования?
2. В чем различие компиляторов и интерпретаторов?
3. Объясните термин «язык низкого уровня» и «язык высокого уровня».
4. Расскажите о поколениях языков программирования.

#### **Программное обеспечение и технологии программирования**

### Лекция 6.1. Системы программирования

Процесс создания программы включает:  
Составление исходного кода программ (рис. 6.1) на языке программирования;  
Этап трансляции, необходимый для создания объектного кода программы;  
Построение загрузочного модуля, готового к исполнению.

Все перечисленные выше действия требуют наличия специальных программных средств.



Рис. 6.1. Процесс создания программы, готовой к исполнению

Совокупность этих программных средств, входит в состав *системы программирования*:

- Текстовый редактор (необходимый для создания и редактирования исходного кода программы на языке программирования);
- Компилятор;
- Редактор связей;
- Отладчик;
- Библиотеки функций;
- Справочная система.

#### 6.1.1. Классификация языков программирования

Многие языки, первоначально разработанные для больших и малых ЭВМ, в дальнейшем были приспособлены к персональным компьютерам. Хорошо вписались в ПК не только Паскаль, Бейсик, Си, Лого, но и ЛИСП, ПРОЛОГ - языки искусственного интеллекта.

В течение многих лет программное обеспечение строилось на основе операционных и процедурных языков, таких как Фортран, Бейсик, Паскаль, Ада, Си. И сегодня современные версии этих и им подобных языков (Модула, Форт и др.) доминируют при разработке прикладных программных средств. Однако по мере эволюции языков программирования получили широкое распространение и другие, принципиально иные, подходы к созданию программ (рис. 6.1.1).

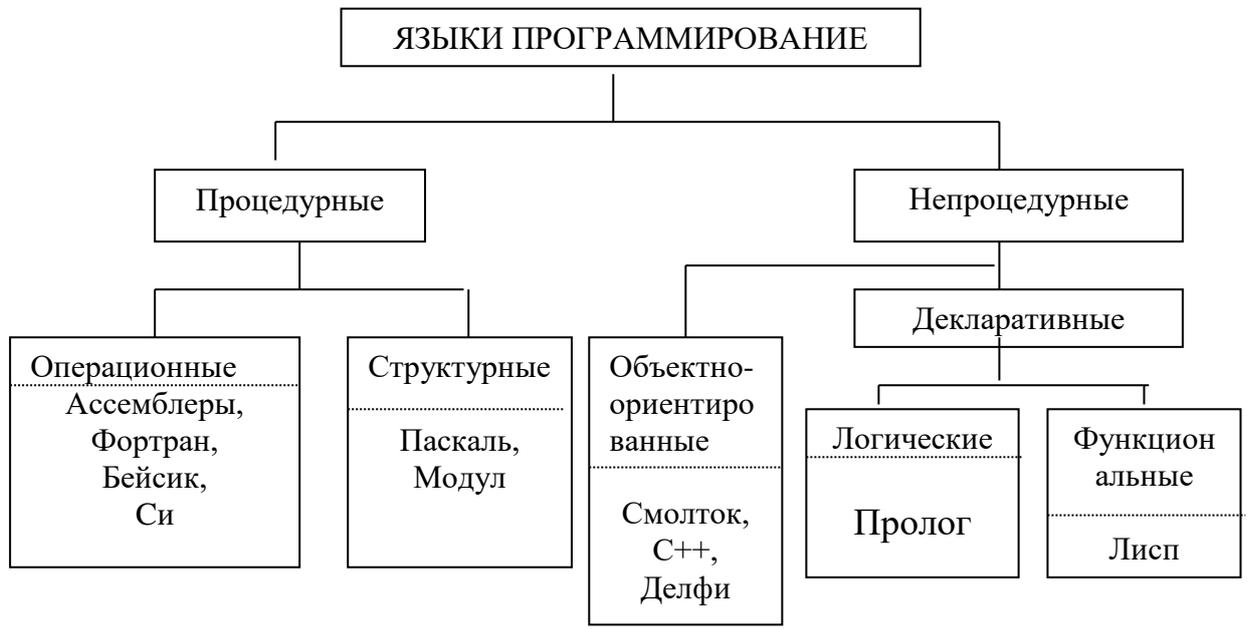


Рис. 6.1.1. Классификация языков программирования

Классическое операционное и/или процедурное программирование требует от программиста детального описания того, как решать задачу, т.е. формулировки алгоритма и его специальной записи. При этом ожидаемые свойства результата обычно не указываются. Основные понятия языков этих групп - оператор и данные. При процедурном подходе операторы объединяются в группы - процедуры. Структурное программирование в целом не выходит за рамки этого направления, оно лишь дополнительно фиксирует некоторые полезные приемы технологии программирования.

Принципиально иное направление в программировании связано с методологиями (иногда говорят «парадигмами») непроцедурного программирования. К ним можно отнести объектно-ориентированное и декларативное программирование. Объектно-ориентированный язык создает окружение в виде множества независимых объектов. Каждый объект ведет себя подобно отдельному компьютеру, их можно использовать для решения задач как «черные ящики», не вникая во внутренние механизмы их функционирования. Из языков объектного программирования, популярных среди профессионалов, следует назвать, прежде всего, Си++, для более широкого круга программистов предпочтительны среды типа Delphi и Visual Basic.

При использовании декларативного языка программист указывает исходные информационные структуры, взаимосвязи между ними и то, какими свойствами должен обладать результат. При этом процедуру его получения («алгоритм») программист не строит (по крайней мере, в идеале). В этих языках отсутствует понятие «оператор» («команда»). Декларативные языки можно подразделить на два семейства - логические (типичный представитель - Пролог) и функциональные (Лисп). По всей видимости, непроцедурные языки имеют большое будущее. Все сказанное выше можно отобразить схемой – крупно-структурной классификацией языков программирования. В ней указаны основные методологии программирования; в нижнем ряду - типичные языки соответствующих групп.

### 6.1.2. Обзор языков программирования высокого уровня

*Fortran* (Фортран ). Первый компилируемый язык, созданный Джимом Бэкусом в 50-е годы. Основным критерием при разработке компиляторов Фортрана является эффективность исполняемого кода. В Фортране впервые был реализован ряд важнейших понятий программирования, но создание программы не очень удобно. Однако для этого языка было создано огромное количество библиотек, поэтому Фортран продолжает активно использоваться во многих организациях. Сейчас ведутся работы над очередным стандартом Фортрана (HPF) для параллельных суперкомпьютеров, со множеством процессоров.

*Cobol* (Кобол). Это компилируемый язык для применения в экономической области и решения бизнес задач, разработанный в начале 60-х годов. Отличается большой «многословностью»- его операторы иногда выглядят как обычные английские фразы. В Коболе были реализованы очень мощные средства работы с большими объемами данных, хранящихся 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 году. Был признан 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). Создан 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). Для этого языка имеются и компиляторы, и интерпретаторы, а по популярности он занимает первое место в мире. Он создан в 60-х годах в качестве учебного языка и очень прост в изучении.

*C* (си). Был создан в лаборатории Bell и первоначально не рассматривался как массовый. Он планировался для замены Ассемблера. Си во многом похож на Паскаль и имеет дополнительные средства для прямой работы с памятью (указатели). На этом языке в 70-х годах написано множество прикладных и системных программ и ряд известных ОС (Unix).

*C++* (Си++). Это объектно-ориентированное расширение языка Си, созданное Бьярном Страуструмом в 1980 году. Множество новых мощных возможностей, позволивших резко повысить производительность программистов, наложилось на унаследованную от языка Си определенную низкоуровневость. В результате чего создание сложных и надежных программ требует от разработчиков высокого профессионального уровня.

*Java* (Джава, Ява). Этот язык был создан компанией Sun в начале 90-х годов на основе Си++ путем исключения из него низкоуровневых возможностей, главная особенность этого языка компиляция не в машинный код, а в платформно - независимый байт-код (каждая команда занимает один байт). Этот байт-код может выполняться с помощью интерпретатора виртуальной Java –машины JVM (Java virtual machine), версии которой созданы сегодня для любых платформ. Благодаря наличию множества Java-машин программы на Java можно переносить не только на уровне исходных текстов, но и на уровне двоичного байт-кода, поэтому по популярности язык Ява сегодня занимает второе место в мире после Бейсика.

Особое внимание в развитии этого языка уделяется двум направлениям: поддержке всевозможных мобильных устройств и микрокомпьютеров, встраиваемых в бытовую технику (технология Jini) и создание платформно - независимых программных модулей, способных работать на серверах в глобальных и локальных сетях с различными ОС (технология Java Beans). Пока основной недостаток этого языка – невысокое быстродействие, т.к. язык Ява интерпретируемый.

### 6.1.3. Языки программирования БД

Эта группа отличается от алгоритмических языков, прежде всего решаемыми задачами. Первые БД появились очень рано, как только появилась потребность в обработке больших массивов информации и выборке групп записей по определенным признакам. Для этого был создан структурированный язык запросов SQL. Он основан на могучей математической теории и позволяет выполнять эффективную обработку БД, манипулируя не отдельными записями, а группами записей.

### 6.1.4. Языки программирования для Интернета

*HTML*. Язык для оформления документов. 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.

#### Вопросы для самоконтроля

1. Какие языки программирования высокого уровня вы знаете?

2. Какие языки программирования активно используются и сегодня?

## **Лекция 6.2. Стратегии решения задач**

### **6.2.1. Этапы подготовки и решения задач на компьютере**

Компьютер предназначен для решения разнообразных задач: научно-технических, инженерных, разработки системного программного обеспечения, обучения, управления производственными процессами и т.д. В процессе подготовки и решения на компьютере научно-технических задач можно выделить следующие этапы:

1. *Постановка задачи* – формулируется цель решения задачи, подробно описывается ее содержание; проводится анализ условий, при которых решается поставленная задача, выявляется область определения входных параметров задачи.
2. *Формальное построение модели задачи* – предполагает построение модели с характеристиками, адекватными оригиналу, на основе какого-либо его физического или информационного принципа; анализируется характер и сущность величин, используемых в задаче;
3. *Построение математической модели задачи* – характеризуется математической формализацией задачи, при которой существующие взаимосвязи между величинами выражаются с помощью математических соотношений. Как правило, математическая модель строится с определенной точностью, допущениями и ограничениями;
4. *Выбор и обоснование метода решения* – модель решения задачи реализуется на основе конкретных приемов и методов решения. В большинстве случаев математическое описание задачи трудно перевести на машинный язык. Выбор и использование метода решения позволяет свести решение задачи к конкретному набору машинных команд. При обосновании метода решения рассматриваются вопросы влияния различных факторов и условий на конечный результат, в том числе на точность вычислений, время решения задачи на компьютере, требуемый объем памяти и др.;
5. *Построение алгоритма* – на данном этапе составляется алгоритм решения задачи, в соответствии с выбранным методом решения. Процесс обработки данных разбивается на отдельные относительно самостоятельные блоки, определяется последовательность выполнения этих блоков;
6. *Составление программы* – алгоритм решения переводится на конкретный язык программирования;
7. *Отладка программы* – процесс устранения синтаксических и логических ошибок в программе. В процессе трансляции программы с помощью синтаксического и семантического контроля выявляются недопустимые конструкции и символы для данного языка программирования. Затем проверяется логика работы программы в процессе ее выполнения с конкретными исходными данными. Для этого используются специальные методы и отладчики, выполняющие специальные действия на этапе отладки, такие как удаление, замена или вставка отдельных операторов или целых фрагментов программы, вывод промежуточных результатов, изменение значений заданных переменных и др.
8. *Решение задачи на компьютере и анализ результатов.* Первоначально выполняется многократное решение задачи на компьютере для различных наборов исходных данных. Получаемые результаты анализируются специалистом, поставившим задачу. Разработанная программа поставляется заказчику в виде готовой к исполнению машинной программы. К ней прилагается документация, включающая инструкцию по эксплуатации.

В задачах другого типа некоторые этапы могут отсутствовать. Например, проектирование программного обеспечения не требует построения математической модели.

Все этапы тесно связаны между собой. Например, анализ результатов может привести к необходимости внесения изменений в программу, алгоритм, метод решения или даже в постановку задачи.

#### **Вопросы для самоконтроля**

3. Что называют системой программирования?
4. Что нужно для создания программы?
5. Перечислите этапы подготовки и решения задач на компьютере.

## **Базы данных**

### **Лекция 7.1. Базы данных и системы управления базами данных**

#### **7.1.1. Основные понятия баз данных**

*База данных* – это организованная структура, предназначенная для хранения информации. С понятием базы данных тесно связано понятие *системы управления базой данных* (СУБД). Это комплекс программных средств, предназначенный для создания структуры новой базы, наполнения ее содержимым, редактирования содержимого и визуализация информации. Под *визуализацией информации* базы понимается отбор отображаемых данных в соответствии с заданным критерием, их упорядочение, оформление и последующая выдача на устройство вывода или передача по каналам связи.

Сегодня большинство СУБД позволяют размещать в своих структурах не только данные, но и методы (т.е. программный код), с помощью которых происходит взаимодействие с потребителем или с другим программно-аппаратным комплексом. Таким образом, можно сказать, что в современных базах данных хранятся не только данные, но и информация.

#### **7.1.2. Классификация БД**

**По технологии обработки данных** БД подразделяются на *централизованные* и *распределенные*.

*Централизованная* БД хранится целиком в памяти одной вычислительной системы. Если система входит в состав сети, то возможен доступ к этой БД других систем.

*Распределенная* БД состоит из нескольких, возможно пересекающихся или дублирующих друг друга БД, хранимых в памяти разных вычислительных систем, объединенных в сеть.

**По способу доступа к данным** БД распределяются на локальный и удаленный (сетевой) доступ.

*Локальный доступ* предполагает, что СУБД обрабатывает БД, которая хранится на той же вычислительной системе.

*Удаленный доступ* – это обращение к БД, которая хранится на одной из систем, входящих в компьютерную сеть. Удаленный доступ может быть выполнен по принципу файл-сервер или клиент-сервер.

Архитектура *файл-сервер* предполагает выделение одного из компьютеров сети (сервер) для хранения централизованной БД. Все остальные компьютеры сети (клиенты) исполняют роль рабочих станций, которые копируют требуемую часть централизованной БД в свою память, где и происходит обработка. При большой интенсивности запросов к централизованной БД увеличивается нагрузка на каналы сети, что приводит к снижению производительности ИС в целом.

Архитектура *клиент-сервер* предполагает, что сервер, выделенный для хранения централизованной БД, дополнительно производит обработку клиентских запросов. Клиенты получают по сети уже обработанные данные. Учитывая широкое распространение БД в

самых различных областях, в последнее время архитектура клиент-сервер применяется и на одиночных вычислительных системах. В этом случае клиент-программа, которой понадобились данные из БД, посылает запрос серверу-программе, управляющей велением БД, на специальном универсальном языке запросов. Сервер пересылает программные данные, являющиеся результатом поиска в БД по ее запросу. Этот способ удобен тем, что программа-клиент не обязана содержать все функции поддержания и ведения БД, этим занимается сервер. В результате упрощается написание программ-клиентов. Кроме того, к серверу может обращаться любое количество клиентов.

### 7.1.3. Виды структур данных

В информатике совокупность взаимосвязанных данных называется информационной структурой, или структурой данных.

#### Табличная или реляционная структура

Предположим мы хотим собрать информацию об альбомах музыкальных групп. Пусть имеется информация о некоторых альбомах: 1965, Led Zeppelin 4, Lp, Help!, Atlantic, 1971, Lp (England), EMI, 1970, Flash Gordon, Parlophone, 1980, Led Zeppelin 3, Soundtrack, Lp, Atlantic. Этот список мало о чем говорит. Извлечь какую-либо информацию из этого набора данных практически невозможно.

Представим данные в виде табл. 7.1.1.

Таблица 7.1.1.

#### **Информация об альбомах музыкальных групп**

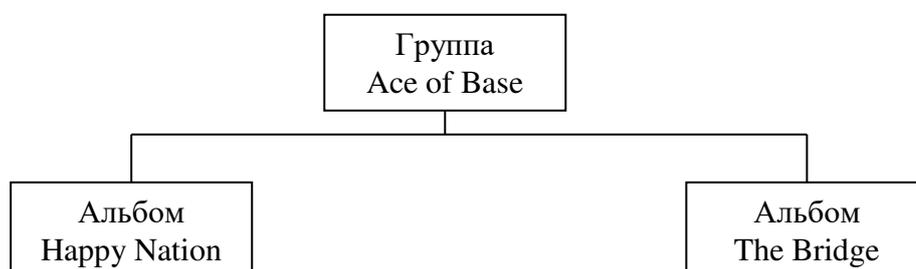
Название альбома	Год выпуска	Тип	Фирма альбома
Help!	1965	Lp (England)	Parlophone
Led Zeppelin 4	1971	Lp	Atlantic
Led Zeppelin 3	1970	Lp	Atlantic
Flash Gordon	1980	Soundtrack	EMI

Теперь воспринимать и использовать информацию стало гораздо удобнее. Представленная таблица является **информационной моделью**. Объектами, отраженными в этой модели, являются музыкальные альбомы (группы), причем все данные взаимосвязаны. Свойствами этих объектов находятся в столбцах таблицы («Название альбома», «Год выпуска», «Тип альбома», «Фирма»), их называют **атрибутами** объектов. Таким образом, каждая строка таблицы – есть совокупность атрибутов объекта. Такую строку называют **записью**, а столбец – **полем записи**.

Табличная организация данных позволяет получить дополнительную информацию, помимо сведений, указанных в атрибутах. К примеру:

- какая группа выпустила больше альбомов за определенный период;
- число альбомов данной группы;
- сколько имеется альбомов типа Soundtrack (музыка к фильму);
- какая фирма выпустила наибольшее число альбомов данной группы.

Иерархическая структура



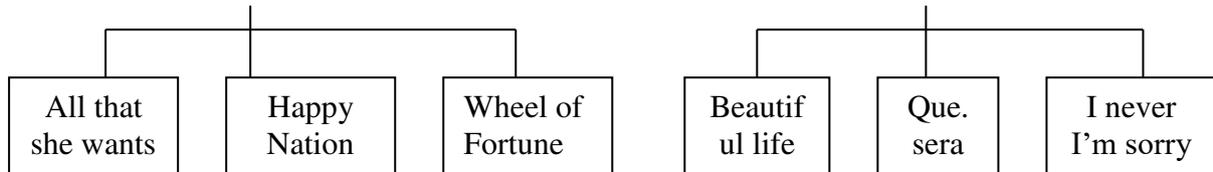


Рис. 7.1.1. Пример иерархической организации данных

Для иерархической структуры (рис. 7.1.1) характерна подчиненность объектов нижнего уровня объектам верхнего уровня.

Важно отметить, что в дереве, между верхними и нижними объектами, задано отношение «один ко многим» (т.е. одной группе соответствует много альбомов, одному альбому соответствует много песен).

В атрибутах, описывающих песню, нет названия альбома, но по линиям связи можно сказать, какая песня принадлежит альбому. По линиям связи можно также определить принадлежность альбома группе. Из данной иерархической структуры можно узнать:

- В каком альбоме больше песен;
- Число альбомов выпущенных группой;
- Есть ли в альбомах одинаковые песни и т.д.

### Сетевая структура

Сетевую структуру данных можно представить в виде схемы, рис. 7.1.2.

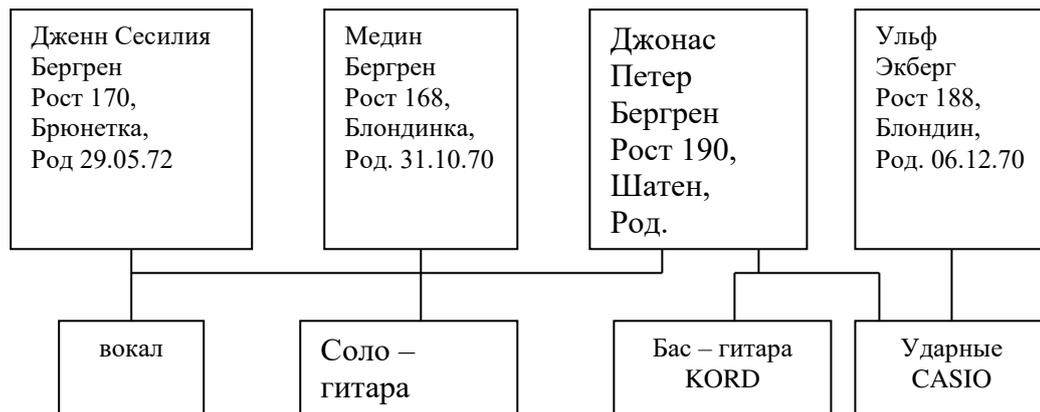


Рис. 7.1.2. Пример сетевой организации данных

В этом случае есть два уровня взаимосвязанных объектов, но отношения между ними «многие ко многим». Глядя на схему можно определить, какими инструментами владеет музыкант, является ли он вокалистом.

Схема позволяет ответить на следующие вопросы:

- Гитары, какой фирмы предпочитает большинство музыкантов;
- Какой музыкант владеет наибольшим количеством инструментов и т.д.

Построение структуры данных происходит в следующем порядке:

- Определяются объекты описания;
- Определяются структуры этих объектов;
- Выбирается тип структуры, отображающий отношения между объектами (табличная, иерархическая, сетевая);
- Строится конкретная информационная структура.

#### 7.1.4. Виды баз данных.

Дадим основное определение. База данных – это реализованная с помощью компьютера информационная структура (модель), отражающая состояние объекта и их отношения. Поскольку основу любой БД составляет информационная структура,

соответственно возможностям организации реляционных (табличных), иерархических или сетевых информационных структур существуют и аналогичные виды БД. В них данные представлены в формах, адекватных соответствующим структурам.

Опыт использования БД позволяет выделить общий набор их рабочих характеристик:

- Полнота – чем полнее БД, тем вероятнее, что она содержит нужную информацию (однако, не должно быть избыточной информации);
- Правильная организация – чем лучше структурирована БД, тем легче в ней найти необходимые сведения;
- Актуальность – любая БД может быть точной и полной, если она постоянно обновляется, т.е. необходимо, чтобы база данных в каждый момент времени полностью соответствовала состоянию отображаемого ею, объекта;
- Удобство для использования – БД должна быть проста и удобна в использовании, и иметь развитые методы доступа к любой части информации.

Наиболее распространенными являются реляционные БД, чем иерархические и сетевые, которые не могут быть реализованы с помощью наиболее популярных СУБД, входящих в состав ПО ЭВМ. Поэтому мы рассмотрим подробнее реляционные БД.

## **Лекция 7.2. Реляционные БД**

Название «реляционные» (в переводе с англ. relation-отношение) связано с тем, что каждая запись в таблице содержит информацию, относящуюся только к одному конкретному объекту. БД могут содержать различные объекты, но основными объектами любой БД являются ее таблицы. Простейшая БД имеет хотя бы одну таблицу. Соответственно, структура простейшей БД тождественно равна структуре ее таблицы. Структуру таблицы простейшей БД составляют поля и записи. Если записей в таблице пока нет, значит, ее структура образована только набором полей. Однако, чаще, БД строятся на основе нескольких таблиц, связанных между собой через общие атрибуты.

### **7.2.1. Свойства полей БД**

Поля БД не просто определяют структуру базы - они еще определяют групповые свойства данных, записываемых в ячейки, принадлежащие каждому из полей. Рассмотрим основные свойства полей таблиц, на примере СУБД MS Access.

*Имя поля* - определяет, как следует обращаться к данным этого поля при автоматических операциях с базой (по умолчанию имена полей используются в качестве заголовков столбцов таблицы);

*Тип поля* - определяет тип данных, которые могут содержаться в данном поле;

*Размер поля* - определяет предельную длину (в символах) данных, которые могут размещаться в данном поле;

*Формат поля* - определяет способ форматирования данных в ячейках, принадлежащих полю;

*Маска ввода* - определяет форму, в которую вводятся данные в поле (средства автоматизации ввода данных);

*Подпись* - определяет заголовок столбца таблицы для данного поля (если подпись не задана, то в качестве заголовка столбца используется имя поля);

*Значения по умолчанию* - то значение, которое вводится в ячейку поля автоматически (средства автоматизации ввода данных);

*Условия на значение* – ограничение, используемое для проверки правильности ввода данных;

*Сообщение об ошибке* – текстовое сообщение, которое выдается автоматически при попытке ввода в поле ошибочных данных (проверка ошибочности производится автоматически, если задано свойство *Условие назначения*);

*Обязательное поле* – свойство, определяющее обязательность заполнения данного поля при наполнении базы;

*Пустые строки* – свойство, разрешающее ввод пустых строковых данных (от свойства обязательное поле отличается тем, что относится не ко всем типам данных, а лишь к некоторым, например, к текстовым);

*Индексированное поле* – если поле обладает этим свойством, все операции, связанные с поиском записи по значению, хранящемуся в данном поле, или сортировкой существенно ускоряются. Кроме того, для индексированных полей можно сделать так, что значения в записях будут проверяться по этому полю на наличие повторов, что позволяет автоматически исключить дублирование данных.

Перечисленные свойства полей относятся в основном к полям текстового типа. Поля других типов могут иметь или не иметь эти свойства, но могут добавлять к ним новые. Например, для данных, представляющих действительные числа, важным свойством является число десятичных знаков.

## 7.2.2. Типы полей

Таблицы БД, как правило, допускают работу с большим количеством разных типов данных. Например, БД MS Access работают со следующими типами данных:

*Текстовый* – тип данных, используемый для хранения обычного неформатированного текста ограниченного размера (до 255 символов);

*Поле Мемо* – специальный тип данных для хранения больших объемов текста (до 65535 символов). Физически текст не хранится в поле. Он хранится в другом месте БД, а в поле хранится указатель на него (поле для комментариев);

*Числовой* – тип данных для хранения действительных чисел;

*Дата/время* – тип данных для хранения календарных дат и текущего времени (от 100 до 9999 г.).

*Денежный* – тип данных для хранения денежных сумм;

*Счетчик* – специальный тип данных для уникальных (не повторяющихся в поле) натуральных чисел с автоматическим наращиванием. Естественное использование – для порядковой нумерации записей;

*Логический* – тип для хранения логических данных (может принимать только одно из значений «Да» или «Нет»);

*Поле объекта OLE* – специальный тип, предназначенный для хранения объектов OLE, например, мультимедийных. В основном, для хранения объектов, созданных другими приложениями: рисунки или таблицы, созданные, например MS Excel;

*Гиперссылка* – специальное поле для хранения адресов URL Web-объектов Интернет. При щелчке на ссылке автоматически происходит запуск броузера и воспроизведение объекта в его окне;

*Мастер подстановок* – это объект, настройкой которого можно автоматизировать ввод данных в поле так, чтобы не вводить их вручную, а выбирать из раскрывающегося списка.

## Лекция 7.3. Проектирование БД

### 7.3.1. Режимы работы с БД

Обычно с БД работают две категории исполнителей. Первая – проектировщики. Их задача состоит в разработке структуры таблиц БД и согласование ее заказчиком. Кроме таблиц проектировщики разрабатывают и др. объекты БД, предназначенные, с одной стороны, для автоматизации работы с базой, а с другой – для ограничения функциональных возможностей работы с базой (если это необходимо из соображений безопасности). Проектировщики не наполняют базу конкретными данными - заказчик может считать их

конфиденциальными и не предоставлять посторонним лицам. Исключение составляет экспериментальное наполнение модельными данными на этапе отладки объектов базы.

Вторая категория - пользователи. Получают исходную базу от проектировщиков и занимаются ее наполнением и обслуживанием. В общем случае пользователи не имеют средств доступа к управлению структурой базы - только к данным, да и то не ко всем, а к тем, работа с которыми предусмотрена на конкретном рабочем месте.

**Соответственно СУБД имеет два режима работы: проектировочный и пользовательский. Первый предназначен для создания или изменения структуры базы и ее объектов. Во втором режиме происходит использование ранее подготовленных объектов для пополнения базы или получения данных из нее.**

### 7.3.2. Объекты БД

Кроме таблиц БД может содержать и другие типы объектов. Каждая СУБД может реализовывать свои типы объектов. Поэтому привести полную классификацию возможных объектов БД затруднительно. Рассмотрим основные типы объектов на примере СУБД Microsoft Access 2000. Эта СУБД позволяет создавать и использовать объекты семи различных типов.

**Таблицы** – это основные объекты любой БД. Во-первых, в таблицах хранятся все данные, имеющиеся в базе, а во-вторых, таблицы хранят и структуру базы (поля, их типы и свойства).

**Запросы** служат для извлечения данных из таблиц и предоставления их пользователю в удобном виде. С помощью запросов выполняется отбор данных, их сортировка и фильтрация. Можно выполнять преобразование данных по заданному алгоритму, создавать новые таблицы, выполнять автоматическое наполнение таблиц данными, импортированными из других источников, выполнять простейшие вычисления в таблицах и многое другое. Все те же операции можно делать и с таблицами. Но запросы используют из соображения удобства (в первую очередь быстродействия) и соображения безопасности. Из соображений безопасности, чем меньше доступа к базовым таблицам имеют конечные пользователи, тем лучше. Во-первых, снижается риск повреждения данных в таблицах неумелыми действиями. Во-вторых, можно эффективно разграничивать доступ к данным, предоставив разным пользователям разные запросы в строгом соответствии с кругом персональных обязанностей. Например, в банках одни сотрудники имеют доступ к таблицам данных о клиенте, другие - к их расчетным счетам, третьи - к таблицам активов банка. Специалисты, имеющие доступ ко всем информационным ресурсам банка, лишены средств, для внесения изменений. Все сделано для того, чтобы один человек не мог совершить фиктивную операцию, независимо от того, какую должность он занимает.

Особенность запросов состоит в том, что они черпают данные из базовых таблиц и создают на их основе временную *результатирующую таблицу* (ее еще называют моментальным снимком).

При работе с основными таблицами базы, имеют дело с жестким диском, т.е. очень медленным устройством. При получении результирующей таблицы на основании запроса имеют дело с электронной таблицей, не имеющей аналога на жестком диске – это только образ отобранных полей и записей. Разумеется, работа с «образом» происходит гораздо быстрее и эффективнее. Это еще одно основание широкого использования запроса.

В СУБД от базовых таблиц не требуется никакой упорядоченности. Все записи в основные таблицы вносят по мере их поступления, т.е. в неупорядоченном виде. Если надо видеть данные, отсортированные или отфильтрованные по какому-либо принципу, для этого также используется соответствующий запрос.

**Формы.** Если запросы – это специальные средства для отбора и анализа данных, то формы – это средства для ввода данных. Смысл их тот же – предоставить пользователю средства для заполнения только тех полей, которые ему заполнять положено. Одновременно

с этим в форме можно разместить специальные элементы управления для автоматизации ввода (счетчики, раскрывающие списки, переключатели, флажки и прочее). Преимущества раскрываются особенно наглядно, когда происходит ввод данных с заполненных бланков. В этом случае форму делают так, чтобы она повторяла оформление бланка – это заметно упрощает работу наборщика, снижает его утомляемость и предотвращает появления печатных ошибок. Формы являются основными объектами для работы с данными, перемещения по записям и подтверждения выполнения различных операций с БД.

**Отчеты.** По своим свойствам и структуре отчеты во многом похожи на формы, но предназначены только для вывода данных, причем не на экран, а на печатающее устройство (принтер). В связи с этим, отчеты отличаются тем, что в них приняты специальные меры для группирования выводимых данных и для ввода специальных элементов оформления, характерных для печатных документов (верхний и нижний колонтитулы, номера страниц, служебная информация о времени создания отчета и т.п.).

**Макросы.** Эти объекты предназначены для автоматизации повторяющихся операций при работе с СУБД. В СУБД MS Access макросы состоят из последовательности внутренних команд СУБД и являются одним из средств автоматизации работы с базой.

**Модули** предназначены для создания новых функций путем программирования. Эти объекты создаются средствами внешнего языка программирования, в данном случае, языка Visual Basic for applications. Это одно из средств, с помощью которых разработчик базы может заложить в нее нестандартные, функциональные возможности, удовлетворяя специфические требования заказчика, повысить быстродействие системы управления, а также уровень ее защищенности.

### 7.3.3. Проектирование БД

Методически правильно начинать работу с карандашом и листом бумаги, не используя компьютер. Неоптимальные решения и прямые ошибки, заложенные на этапе проектирования, в последствии очень трудно устраняются, поэтому этот этап является основополагающим.

Разработка технического задания

Техническое задание на проектирование БД должен предоставить заказчик. Но для этого он должен владеть соответствующей терминологией и знать, хотя бы в общих чертах, технические возможности основных СУБД. Поэтому обычно используют следующие подходы:

- Демонстрируют заказчику работу аналогичной БД, после чего согласовывают спецификацию отличий;
- Если аналога нет, выясняют круг задач и потребностей заказчика, после чего помогают ему подготовить техническое задание.

При подготовке технического задания составляют:

- Список исходных данных, с которыми работает заказчик;
- Список выходных данных, которые необходимы заказчику для управления структурой своего предприятия;
- Список выходных данных, которые не являются необходимыми для заказчика, но которые он должен предоставить в другие организации (в вышестоящие структуры, в органы статистического учета, прочие административные и контролирующие организации).

### 7.3.4. Разработка структуры БД

Выяснив основную часть данных, которые заказчик потребляет или поставляет, можно приступить к созданию структуры базы, т.е. структуры ее основных таблиц.

1. Работа начинается с составления генерального списка полей – он может насчитывать десятки и даже сотни позиций.

2. В соответствии с типом данных, размещаемых в каждом поле, определяют наиболее подходящий тип для каждого поля.

3. Далее распределяют поля генерального списка по базовым таблицам. Цель - обеспечить, чтобы ввод данных в одну таблицу производился, по возможности, в рамках одного подразделения, а еще лучше – на одном рабочем месте.

4. Наметив столько таблиц, сколько подразделений охватывает база данных, приступают к дальнейшему делению таблиц. Критерием деления является факт множественного повтора данных в соседних записях. Например, табл. 7.3.1.

Таблица 7.3.1.

**Сведения о процессор**

Марка	модель	цена	поставщик	Адрес
Celeron	300A	56	ООО «Самшит»	123456, Москва, ул. Индустриальная, д.10
Celeron	333	60	ООО «Самшит»	123456, Москва, ул. Индустриальная, д.10
Intel	366	68	ТОО «Планета»	111222, Москва, ул. Промышленная, д.22
Celeron	406	95	ООО «Самшит»	123456, Москва, ул. Индустриальная, д.10
Intel	433	122	ЗАО «Кристина»	121212, Москва, пр. Технический, 44

В данной таблице в полях «Поставщик» и «Адрес» наблюдается повтор данных. Это говорит о том, таблицу надо поделить на две взаимосвязанные таблицы (табл. 7.3.2 и табл. 7.3.3).

Таблица 7.3.2.

**«Процессор»**

Марка	модель	цена
Celeron	300A	56
Celeron	333	60
Intel	366	68
Celeron	406	95
Intel	433	122

Таблица 7.3.3.

«Поставщики»

Наименование	Индекс	Город	Адрес	Телефон
ООО «Самшит»	123456	Москва	ул. Индустриальная, д.10	129-62-51
ТОО «Планета»	111222	Москва	ул. Промышленная, д.22	456-43-21
ЗАО «Кристина»	121212	Москва	пр. Технический, 44	175-84-22

5. В каждой из таблиц намечают **ключевое поле**. В качестве такового выбирают поле, данные в котором повторяться не могут. Например, для таблицы данных о студентах таким полем может служить индивидуальный шифр студента (№ зачетной книжки). Для таблицы 7.3.2 «Процессор» такого поля нет, но его можно создать искусственно комбинированием полей «Марка » и «Модель». Эта комбинация неповторима, т.к. в таблице незачем описывать одну и ту же модель.

Если же в таблице таких полей нет, которые можно было бы использовать как ключевые, всегда можно ввести дополнительное поле типа Счетчик – оно не может содержать повторяющихся данных по определению. Например, в таблицу 7.3.3 «Поставщики» можно ввести поле «Код поставщика».

Код поставщика	Наименование	Индекс	Город	Адрес	Телефон
1	ООО «Самшит»	123456	Москва	ул. Индустриальная, д.10	129-62-51
2	ТОО «Планета»	111222	Москва	ул. Промышленная, д.22	456-43-21
3	ЗАО «Кристина»	121212	Москва	пр. Технический, 44	175-84-22

6. С помощью карандаша и бумаги расчерчивают связи между таблицами. Существует несколько типов возможных связей между таблицами. Наиболее распространенными являются связи «один ко многим» и «один к одному». Связь между таблицами организуется на основе общего поля, причем в одной из таблиц оно обязательно должно быть ключевым, т.е. на стороне «один» должно выступать ключевое поле, содержащее уникальные, неповторяющиеся значения. Значения на стороне «многие» могут повторяться.

Для того, чтобы можно было получать сведения о поставщиках процессоров, надо связать таблицы 7.3.2 и 7.3.3. Чтобы можно было осуществить связь, в таблице «Поставщики» поле «Код поставщика» надо сделать ключевым. В таблицу 7.3.2. надо добавить поле «Код поставщика» и связать таблицы по этому полю (рис. 7.3.1).

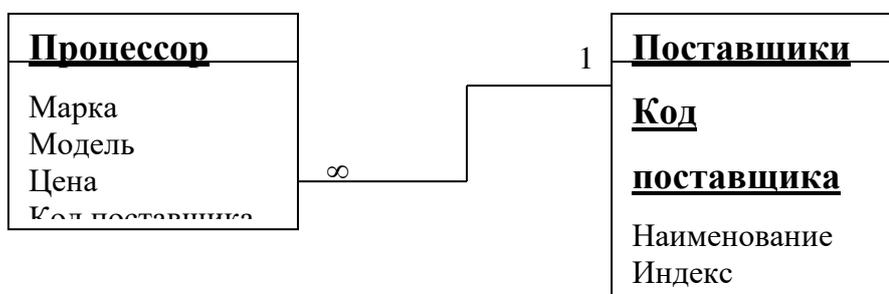


Рис. 7.3.1. Схема связей между таблицами

Про подобные таблицы говорят, что они связаны *реляционными отношениями*. Соответственно, системы управления, способные работать со связанными таблицами,

называются *системами управления реляционными БД*, а схему данных могут называть *схемой реляционных отношений*.

7. Разработкой схемы данных заканчивается «бумажный» этап работы над техническим предложением. Схему можно согласовать с заказчиком, после чего приступить к непосредственному созданию БД.

### 7.3.5. Создание объектов БД MS Access и режимы работы с ними

**Таблица** – это объект, который определяется и используется для хранения данных. Каждая таблица хранит информацию об объекте определенного типа. Как вы уже знаете, таблица содержит поля (столбцы) и записи (строки). Работать с таблицей можно в двух основных режимах: в режиме конструктора и в режиме таблицы.

В режиме конструктора задается структура таблицы, т.е. определяются типы, свойство полей, их число и названия (заголовки столбцов). Он используется, если нужно изменить структуру таблицы, а не хранимых в ней данных. Чтобы перейти в режим таблицы, надо дважды щелкнуть мышью по имени нужной таблицы в окне БД (или, выделив в окне БД имя нужной таблицы, воспользоваться кнопкой [Открыть] окна БД).

Из режима конструктора перейти в режим таблицы можно, щелкнув по кнопке [Таблицы] на панели инструментов.

В режиме конструктора и в режиме таблицы перемещение между полями осуществляется с помощью клавиши TAB, а также [вверх] или [вниз] по записям с помощью клавиш, гораздо удобнее использовать мышь.

**Создание межтабличных связей.** Если структура БД продумана заранее, а связи между таблицами намечены, то создание реляционных отношений между таблицами выполняется с помощью мыши в специальном окне *Схема данных*. Образовавшаяся межтабличная связь отображается в окне *Схема данных* в виде линии, соединяющей два поля разных таблиц. При этом одна из таблиц считается *главной*, а другая – *связанной*. *Главная* – это таблица, участвующая в связи своим ключевым полем (название этого поля на схеме данных отображается полужирным шрифтом). Ключевое поле определяется в таблице в режиме конструктора. У связи два основных назначения. Первое – обеспечение целостности данных, а второе – автоматизация задач обслуживания базы.

Связь между таблицами позволяет:

- либо исключить возможность удаления или изменения данных в ключевом поле главной таблицы, если с этим полем связаны какие-либо поля других таблиц;
- либо сделать так, что при удалении (или изменении) данных в ключевом поле главной таблицы автоматически (и абсолютно корректно) произойдет удаление или изменение соответствующих данных в полях связанных таблиц.

Для настройки свойств связи надо в окне *Схема данных* выделить линию связи правой кнопкой мыши и в открывшемся контекстном меню выбрать пункт, *Изменить связь*. Откроется окно *Изменение связи*. В нем показаны названия связанных таблиц, и имена полей, участвующих в связи (здесь же их можно изменить). А также приведены элементы управления для обеспечения условий целостности данных.

Если установлен флажок *Обеспечение целостности данных*, то удалять данные из ключевого поля главной таблицы нельзя. Если вместе с ним включены флажки *Каскадное обновление связанных полей* и *Каскадное удаление связанных записей*, то, соответственно, операции редактирования и удаления данных в ключевом поле главной таблицы разрешены, но сопровождаются автоматическими изменениями в связанной таблице.

Таким образом, смысл создания реляционных связей между таблицами состоит, с одной стороны, в защите данных, а с другой – в автоматизации внесения изменений сразу в несколько таблиц, при изменениях в одной таблице.

**Работа с запросами.** Если структура БД хорошо продумана, то исполнители, работающие с базой, должны навсегда забыть, что в базе есть таблицы, а лучше, если они

об этом вообще ничего не знают. Таблицы – слишком ценные объекты базы, чтобы с ними имел дело кто-либо, кроме разработчика базы.

Если исполнителю надо получить данные из базы, он должен использовать специальные объекты – запросы. Все необходимые запросы разработчик базы должен подготовить заранее. Если запрос подготовлен, надо открыть вкладку Запросы в окне БД, выбрать его и открыть двойным щелчком на значке – откроется результирующая таблица.

В общем случае результирующая таблица может не соответствовать на одной из базовых таблиц БД. Ее поля могут представлять набор из полей разных таблиц, а ее записи могут содержать отфильтрованные и отсортированные записи таблиц, на основе которых формировался запрос.

Запросы лучше готовить вручную, с помощью *Конструктора*. Для этого есть специальный значок в окне БД. Он называется *Создание запроса в режиме конструктора* и открывает специальный бланк, называемый *бланком запроса по образцу*. Хотя запросы к таблицам пишутся на специальном языке программирования SQL, пользователя изучать его не обязательно, а большинство операций можно выполнить щелчками кнопкой мыши и приемом перетаскивания в бланк (рис. 7.3.2).

Запрос1: запрос по образцу								
<table border="1"> <tr> <td>Клиенты</td> <td>Заказы</td> </tr> <tr> <td>Код клиента</td> <td>Код заказа Код клиента</td> </tr> </table>		Клиенты	Заказы	Код клиента	Код заказа Код клиента			
Клиенты	Заказы							
Код клиента	Код заказа Код клиента							
Поле	Код заказа	Код клиента	Название	Адрес				
Имя таблицы	Заказы	Заказы	Клиенты	Клиенты				
Сортировка								
Вывод на экран	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
Условие отбора								

Рис. 7.3.2. Бланк запроса по образцу

Бланк запроса по образцу состоит из двух областей. В верхней отображается структура таблиц, к которым запрос адресован, а нижняя область разбита на столбцы и строки – по одному столбцу на каждое поле будущей результирующей таблицы. С помощью контекстного меню в верхней части бланка открывают те таблицы, к которым обращен запрос. Затем в них щелкают двойными щелчками на названиях тех полей, которые должны войти в результирующую таблицу. При этом автоматически заполняются столбцы в нижней части бланка. Сформировав структуру запроса, его закрывают, дают ему имя и в дальнейшем запускают двойным щелчком на значке в окне БД. Так создается простейший запрос, называемый *запросом на выборку*. Он позволяет выбрать данные из полей таблиц, на основе которых запрос сформирован.

**Упорядочение записей в результирующей таблице.** Если необходимо, чтобы данные, отобранные в результате запроса на выборку, были упорядочены по какому-либо полю, применяют сортировку. В нижней части бланка запроса имеется специальная строка *Сортировка*. При щелчке на этой строке открывается кнопка раскрывающегося списка, в котором можно выбрать метод сортировки: по возрастанию или по убыванию. В результирующей таблице данные будут отсортированы по тому полю, для которого задан порядок сортировки. Возможна многоуровневая сортировка – сразу по нескольким полям. В этом случае данные сначала сортируются по тому полю, которое в бланке запроса находится левее, затем по следующему полю, для которого включена сортировка, и т.д. слева направо.

При формировании запроса надо располагать поля результирующей таблицы не как попало, а с учетом будущей сортировки. Если запрос уже сформирован и надо изменить порядок следования столбцов, используют следующий прием:

- выделяют столбец щелчком на его заголовке (кнопку мыши отпускают);
- еще раз щелкают на заголовке уже выделенного столбца (но кнопку не отпускают);
- перетаскивают столбец в другое место.

**Управление отображением данных в результирующей таблице.** В нижней части бланка запроса по образцу имеется строка *Вывод на экран*. По умолчанию предполагается, что все поля, включенные в запрос, должны выводиться на экран. Когда не желательно, чтобы пользователь базы видел содержание некоего поля, но его необходимо включить в запрос (например, потому, что оно является полем сортировки или по другой причине), то отображение содержимого поля подавляется сбросом флажка *Вывод на экран*.

**Использование условий отбора.** Дополнительным средством, обеспечивающим отбор данных по заданному критерию, является так называемое *Условие отбора* – строка в нижней части бланка запроса по образцу. Для каждого поля в этой строке можно задать индивидуальное условие.

#### **Формирование условий отбора:**

**1. Текста:** В большинстве случаев данные имеют текстовый тип. Для формирования условий отбора текста имеются некоторые специфические выражения.

а) для поиска полной текстовой строки, следует ввести ее в соответствующую ячейку бланка запроса нужного поля; Access заключит выражение в кавычки;

б) в качестве указателя произвольного текста используется символ «\*». Одна звездочка интерпретируется как произвольное количество любых символов. В одном выражении может использоваться несколько звездочек. Условию «Ив\*н\*ов» удовлетворяют, например, фамилии: «Иванов», «Иванцов» и «Иванников».

в) если нужно отобрать записи, несоответствующие указанному тексту, перед ним вводится ключевое слово «NOT» или оператор сравнения «<>». Условие NOT Москва инициирует отбор всех городов, кроме Москвы.

г) в качестве заменителя одного произвольного символа применяется вопросительный знак «?». Например, по условию «к?t» могут быть найдены слова «кит» и «кот».

**2. Чисел:**

а) для отбора конкретной величины, в ячейку вводится число;

б) чтобы отобрать записи с величинами, меньшими либо равными определенному значению, вводится <число или <=число;

в) чтобы отобрать записи с величинами большими либо равными определенному значению, вводится >число или >=число;

г) для поиска величин, лежащих в некотором промежутке чисел (включая его начало и конец), вводят Between число1 and число2;

д) для поиска величин, лежащих в некотором промежутке чисел (исключая его начало и конец), вводится выражение >число1 and <число2.

**3. Сложное условие отбора:**

а) если осуществляется поиск по нескольким альтернативным условиям для одного поля, их вводят в одну ячейку и соединяют ключевым словом OR (например, «Юг or Север»);

б) если имеется много возможных вариантов, используется ключевое слово IN (например, IN(Юг; Север; Восток)).

в) если одно поле должно удовлетворять сразу двум критериям, используют ключевое слово AND (например, >5 and <10);

г) чтобы определить условие для нескольких полей в случае поиска тех записей, которые удовлетворяют хотя бы одному условию, вводится условие для первого поля, переходят в следующую строку условий отбора, которая называется «ИЛИ» и вводят выражение для второго поля в столбце этого поля. При наличии третьего условия перемещаются еще на одну строку ниже.

**Другие виды запросов.** Запросы на выборку самые простые, но и самые распространенные. Существуют и другие виды запросов, некоторые из них выполняются на базе предварительно созданного запроса на выборку.

К ним относятся, прежде всего:

1. Запросы с параметром. Часто приходится иметь дело с запросами, устроенными в принципе одинаково, но имеющими некоторые различия в поставленных условиях отбора. В таком случае, чтобы каждый раз заново не создавать отдельный запрос, следует сформировать запрос с параметром. В запросе с параметром в бланк запроса в строке **Условие отбора** нужного поля вводят, выражение, например: <[Введите максимальное значение]. При выполнении запроса с параметром Access запрашивает у пользователя значение параметра для определения условий выборки.

2. Запрос на выполнение действий, которые используются для создания новых реальных таблиц данных, в дальнейшем существующих уже независимо от тех таблиц базы, которые были использованы для их построения. Эти запросы позволяют также изменять таблицы базы данных: обновлять их, дополнять новыми записями или удалять некоторые записи. Различают четыре вида запросов на выполнение действия:

а) *Запрос на добавление.* Можно добавлять отобранные записи из таблицы или запроса текущей БД в конец другой таблицы. Дополняемая таблица может находиться как в той же самой, так и в другой БД;

б) *Запрос на удаление.* С помощью запроса на удаление можно удалить группу записей данных, удовлетворяющих заданным условиям;

в) *Запрос на обновление.* Можно изменить группу записей данных, удовлетворяющих определенному условию.

г) *Запрос на создание новой таблицы.* Из выборки, являющейся динамической таблицей, сформированной при выполнении запроса и существующей только до окончания работы с результатами запроса, можно создать новую таблицу БД. Она начнет свое независимое существование. Изменения в этой новой таблице не будут затрагивать использованные для ее построения исходные таблицы БД.

3. Итоговые запросы. Назначение их отдаленно напоминает итоговые функции электронных таблиц (производят математические вычисления по заданному полю и выдают результат).

4. Запросы на изменение, позволяют автоматизировать заполнение полей таблиц.

5. Перекрестные запросы, позволяющие создавать результирующие таблицы на основе результатов расчетов, полученных при анализе группы таблиц.

6. Специфические запросы SQL – запросы к серверу БД, написанные на языке запросов SQL.

### **Работа с формами**

Форма – это объект, в основном, предназначенный для удобного ввода отображения данных. Надо отметить, что в отличие от таблиц, в формах не содержится информация баз данных. Формы всего лишь формат (бланк) показа данных на экране компьютера. Формы могут строиться только на основе таблиц или запросов. Построение форм на основе запросов позволяет представлять в них информацию из нескольких таблиц.

В форму могут быть внедрены рисунки, диаграммы, аудио (звук) и видео (изображение).

Режимы работы с формой:

- *режим формы* – используется для просмотра и редактирования данных, предоставляет дружественную среду для работы с данными и удобный дизайн их представления на экране;

- режим конструктора форм необходим, если надо изменить определение формы (структуру или шаблон формы, а не представленные в ней данные). Для этого надо открыть форму в режиме конструктор
- режим таблицы позволяет увидеть таблицу, включающую все поля формы. Чтобы переключиться в этот режим при работе с формой надо нажать кнопку [таблицы] на панели инструментов.

Комплектующие	
Процессор	Celeron
Модель	300A
Частота	300
Цена	58
Поставщик	ТОО «Кристина»
Адрес	Москва, ул. Пятницкая
Записи	1 из 15

Рис. 7.3.3. Автоформа «в столбец»

Автоформа «в столбец» отображает все поля одной записи. Она удобна для ввода и редактирования данных.

Анализ прайс листа		Запрос на выборку	
	Процессор	Частота	Цена
▷	Celeron	366	68
	АМД КБ-2	400	77
*		0	0
<b><u>Запись 1</u></b>   < < 1 > >   ▷ * Из 2			

Рис. 7.3.4. Автоформа «ленточная»

**Автоформы.** В отличие от таблиц и запросов, которые лучше формировать вручную, формы удобнее готовить с помощью средств автоматизации. Полностью автоматическими являются средства, названные *автоформами*.

Существует три вида автоформ: «в столбец», «ленточные» и «табличные» (рис. 7.3.3-73.5)

Ленточная автоформа отображает одновременно группу записей. Её удобно использовать для оформления вывода данных.

Комплектующие			

	Процессор	Модель	<u>Частота</u>	Цена	Δ			
▷	Celeron	300A	300	58				
	Celeron	333	333	61				
	Celeron	366	366	68				
	Celeron	366A	366	85				
Запись	◀	◀	1	▷	▷	▷*		▽

Рис. 7.3.5. Автоформа табличная:

Табличная автоформа по внешнему виду ничем не отличается от таблицы, на которой она основана.

Для создания автоформы следует открыть панель **Формы** в окне БД и воспользоваться командной кнопкой [Создать]. В открывающемся диалоговом окне **Новая форма** выбирают тип автоформы и таблицу (или запрос), на которой она основывается. После щелчка на кнопке [ОК] автоформа формируется автоматически и немедленно готова к работе, то есть к вводу или отображению данных.

На рисунках 7.3.3-7.3.5 отображены автоформы, основанные только на одном объекте. Такая форма называется *Простой формой*. В основу структуры формы можно закладывать поля нескольких таблиц или запросов. Если форма основывается на полях из нескольких связанных таблиц, то она называется *сложной* или *составной* и представляет собой композицию из нескольких форм.

#### **Создание форм с помощью мастера**

Автоматизированные средства предоставляет *Мастер форм* - специальное программное средство, создающее структуру формы в режиме диалога. *Мастер форм* можно запустить из окна БД щелчком на значке **Создание форм с помощью мастера** на панели **Формы**.

1. На первом этапе работы *Мастера форм* выбирают таблицы и поля, которые войдут в будущую форму.

2. На втором этапе выбирается внешний вид формы.

3. На третьем - выбирается стиль оформления формы.

4. На четвертом (последнем) – форма сохраняется под заданным именем. Здесь же можно включить переключатель [Изменить макет формы], который открывает только что созданную форму в режиме конструктора. Форма имеет три основных раздела: область заголовка, область данных и область примечания. Линии, разделяющие разделы, перетаскиваются по вертикали с помощью мыши – это позволяет изменять размеры разделов так, как требуется.

#### *Структура формы*

<b>Форма2: Форма</b>						Δ
..	☞ Заголовок формы					
	Наименование	Индекс	Город	Адрес	Телефон	
..	☞ Область данных					
	Наименование	Индекс	Город	Адрес	Телефон	

..	↶ Примечание формы	
<		> ▾

Рис. 7.3.6. Структура формы, созданной мастером

Разделы заголовка и примечания имеют чисто оформительское назначение – их содержимое напрямую не связано с таблицей или запросом, на котором основана форма. Раздел данных имеет содержательное значение – в нем представлены элементы управления, с помощью которых выполняется отображение данных или их ввод. Здесь можно разместить дополнительные элементы управления для автоматизации ввода данных (переключатели, флажки, списки и др., типичные для приложений Windows).

**Элементы управления формы** представлены на *Панели элементов*, которая открывается щелчком на соответствующей кнопке панели инструментов или командой Вид → *Панель элементов*. Выбор элемента управления выполняется одним щелчком на его значке в *Панели элементов*, после чего следующим щелчком в поле формы отмечается место, куда он должен быть поставлен. Вместе с элементом в поле формы вставляется его *присоединенная подпись*. По умолчанию, эта подпись стандартная, например для переключателя это Переключатель1, Переключатель2 и т.д. Редактированием свойства элемента управления (через контекстное меню) можно дать элементу управления более содержательную подпись.

Основными элементами оформления формы являются *текстовые надписи* и *рисунки*. Для создания текстовых надписей служат два элемента – *Надпись* и *Поле*. В качестве надписи можно задать произвольный текст. Элемент *Поле* отличается тем, что в нем отображается содержимое одного из полей таблицы, на которой основана форма, т.е. при переходе от записи к записи текст может меняться.

Для создания графических элементов оформления служат элементы управления *Рисунок*, *Свободная рамка объекта* и *Присоединенная рамка объекта*. Рисунок выбирается из графического файла и вставляется в форму. Элемент *Свободная рамка объекта* отличается тем, что это не обязательно рисунок – это может быть любой другой объект OLE, например, мультимедийный. Элемент *Присоединенная рамка объекта* тоже в какой-то степени может служить для оформления формы, но его содержимое берется непосредственно из таблицы БД (если она имеет поле объекта OLE). При переходе между записями содержимое этого элемента будет меняться.

**Дизайн формы.** В то время, как таблицы БД глубоко скрыты от посторонних глаз, формы БД – это средства, с помощью которых с ней общаются люди. Поэтому к формам предъявляются повышенные требования по дизайну.

В первую очередь, все элементы управления форм должны быть аккуратно выровнены. Это обеспечивается командой Формат → Выровнять. Если нужно равномерно распределить элементы управления по полю формы, используются средства меню Формат → Интервал по горизонтали или Формат → Интервал по вертикали.

Возможно также ручное изменение размеров и положения элементов управления, но оно редко приводит к качественным результатам. При работе вручную используется перетаскивание маркеров, которые становятся видны, когда элемент управления выделен. Обычно элементы управления перетаскиваются вместе с присоединенными к ним надписями. Перетаскивание с помощью маркера левого верхнего угла позволяет оторвать присоединенную надпись от элемента.

Существенную помощь при разработке дизайна формы оказывает вспомогательная сетка. Ее отображение включают командой Вид → сетка. Автоматическую привязку к узлам сетки включают командой Формат → Привязать к сетке.

Закончив разработку макета формы, ее следует закрыть и сохранить под заданным именем. После открытия формы в окне БД, с ней можно работать: просматривать или редактировать данные из базовой таблицы.

Отчеты во многом похожи на формы, но имеют иное функциональное назначение – они служат для форматированного вывода данных на печатающее устройство и, соответственно, при этом должна учитывать параметры принтера и параметры используемой бумаги.

Так же, как и для формы, существуют средства автоматического, автоматизированного и ручного проектирования отчетов. Средства автоматического проектирования реализованы автоотчетами (БД → Создать → Новый отчет → Автоотчет в столбец/ленточный автоотчет).

Средством автоматизированного создания отчетов является *Мастер отчетов*. Он работает в шесть этапов:

1. Выбор базовых таблиц или запросов, на которых отчет базируется;
2. Выбор полей, отображаемый в отчете;
3. Выбор полей группировки;
4. Выбор полей и методов сортировки;
5. Выбор формы печатного макета;
6. Выбор стиля оформления.

Структура готового отчета отличается от структуры формы только увеличенным количеством разделов. Кроме разделов заголовка, примечания и данных, отчет может содержать разделы верхнего и нижнего колонтитулов. Если отчет занимает более одной страницы, эти разделы необходимы для печати служебной информации, например, номеров страниц.

Редактирование структуры отчета выполняют в режиме *Конструктора*. Приемы редактирования те же, что и для форм. Элементы управления в данном случае выполняют функции элементов оформления. Размещение элементов управления выполняют с помощью *Панели элементов* (Вид → Панель элементов), которая по составу практически не отличается от Панели элементов формы. Важной особенностью отчетов является наличие средства для вставки в область верхнего или нижнего колонтитулов текущего номера страницы и полного количества страниц. Эту операцию выполняют с помощью диалогового окна *Номера страниц* (Вставка → Номера страниц).

х

1. Какие данные называются структурированными?
2. Дайте определение и опишите назначение базы данных.
3. Дайте определение и опишите назначение системы управления базой данных.
4. Поясните, в чем заключается различие архитектур баз данных, организованных по принципу клиент-сервер и файл-сервер.
5. Назовите и поясните взаимосвязь структурных элементов базы данных.
6. Дайте понятие ключа. Какие виды ключей вы знаете?
7. Данные, какие типов могут храниться в полях базы данных?
8. Какие модели данных вы знаете?
9. Поясните назначение ключевых полей в реляционной базе данных.
10. Какие виды связей между объектами вам известны?
11. Каковы основные, функциональные возможности СУБД?
12. Назовите и охарактеризуйте основные этапы проектирования БД.
13. Назовите и охарактеризуйте назначение основных объектов БД.

## Локальные и глобальные сети ЭВМ

### **Лекция 8.1. Программные и аппаратные компоненты компьютерных сетей**

#### **8.1.1. Определение и назначение**

Компьютерной сетью называется объединение нескольких ЭВМ для совместного решения информационных, вычислительных, учебных и др. задач.

Простейшим видом сети является *одно - ранговая* сеть, обеспечивающая связь ПК конечных пользователей и позволяющая совместно использовать дисководы, принтеры, файлы.

Более развитые сети включают специальные выделенные компьютеры-серверы. Сервер – ЭВМ, выполняющая в сети особые функции обслуживания остальных компьютеров, которые называются рабочими станциями. Есть серверы файловые, коммуникационные, серверы для проведения математических расчетов, серверы баз данных.

#### **8.1.2 Классификация сетей**

Компьютерные сети можно классифицировать по ряду признаков, в том числе по степени территориальной распределенности. При этом различают: глобальные, региональные и локальные сети.

Глобальные сети объединяют пользователей по всему миру и часто используют спутниковые каналы связи. В этих сетях осуществляется формирование единого научного, экономического, социального и культурно информационного пространства. При этом соединяются узлы связи и ЭВМ, находящиеся на расстоянии 10-15 тыс. км друг от друга.

Региональные сети объединяют пользователей города, области и маленькой страны. Расстояние между узлами сети составляют 10-1000км.

Локальные сети ЭВМ связывают абонентов одного или близлежащих зданий одного предприятия, учреждения (обычно 10-100 компьютеров). Особенностью ЛВС является наличие одного высокоскоростного канала связи для передачи информации в цифровом виде. На больших расстояниях такой тип передачи неприемлем из-за неизбежного затухания высокочастотного сигнала. Существуют проводные и беспроводные (радио) каналы. Расстояния в ЛВС между ЭВМ небольшие – до 10км, а в радио канале – до 20км.

Назначение всех видов компьютерных сетей определяется двумя функциями:

- обеспечение *совместного использования* аппаратных и программных ресурсов сети;
- обеспечение *совместного доступа* к ресурсам данных.

Так, например, все участники локальной сети могут совместно использовать одно общее устройство печати (сетевой принтер) или, например, ресурсы жестких дисков одного выделенного компьютера (файлового сервера). Это же относится и к программному, и к информационному обеспечению. Если в сети имеется специальный компьютер, выделенный для совместного использования участниками сети, он называется файловым сервером. Компьютерные сети, в которых нет выделенного сервера, а все локальные компьютеры могут общаться друг с другом на «равных правах» (обычно это небольшие сети), называются *одноранговыми*.

#### **8.1.3. Сетевое программное обеспечение**

Для реализации услуг в сети и доступа пользователей к услугам разрабатываются программное обеспечение. В настоящее время получили распространение 2 основные концепции такого программного обеспечения.

В первой концепции сетевое программное обеспечение ориентировано на предоставление пользователям некоторого общедоступного главного компьютера сети, называемого файловым сервером. Основным ресурсом такого сервера являются файлы. Это могут быть файлы, содержащие программные модули или данные. Сетевые программные средства, управляющие ресурсами файлового сервера и предоставляющие к ним доступ пользователям сети, называются *сетевой операционной системой*. Ее основная часть размещается на файловом сервере, на рабочих страницах устанавливается только небольшая оболочка, выполняющая функции интерфейса между программами, обращающимися за ресурсами, и файловым сервером.

Во второй концепции - популярной и чрезвычайно перспективной, называемой архитектурой «клиент-сервер», программное обеспечение ориентировано не только на коллективное использование ресурсов, но и на их обработку в месте размещения ресурсов по запросам пользователя. Программные системы архитектуры «клиент-сервер» состоят из 2 частей: программное обеспечение сервера и программное обеспечение пользователя-клиента. При этом в функции клиента входит:

- Предоставление пользовательского интерфейса, ориентированного на определенные производственные обязанности и полномочия пользователя;
- Формирование запросов к серверу;
- Анализ ответов сервера на запросы и предъявление их пользователю.

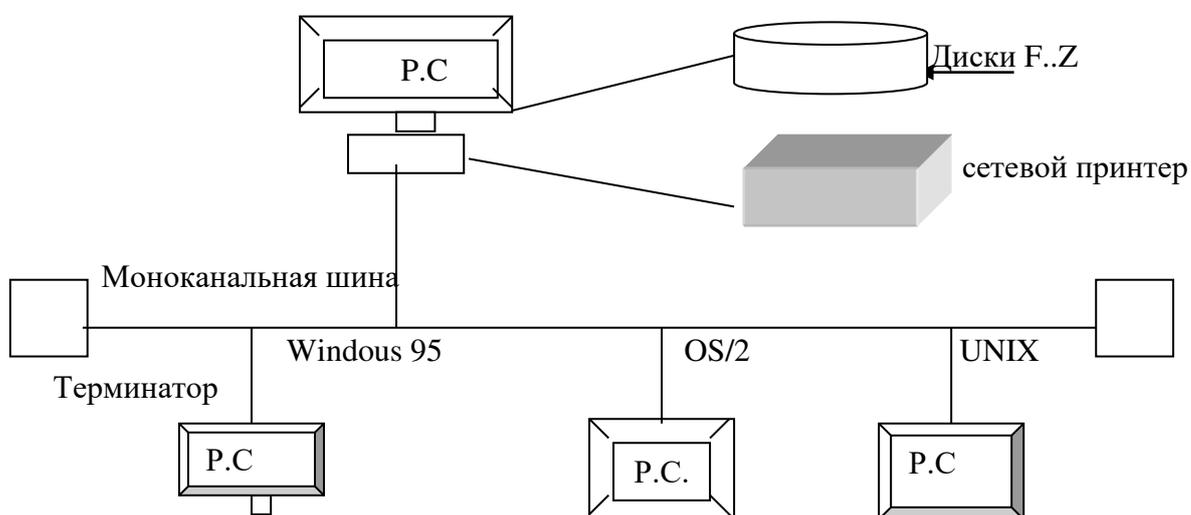
Основная функция сервера – выполнение специальных действий по запросам клиента, например, решение сложной математической задачи, соединение клиента с другим клиентом и т.д. Если сервер не в состоянии решить задачу из-за нехватки ресурсов, он в идеале сам находит более мощный сервер и передает ему задачу, становясь, в свою очередь клиентом, но, не информируя об этом, без особой нужды, начального клиента. Клиентом может быть весьма мощный компьютер, который в силу своих возможностей решает задачу самостоятельно.

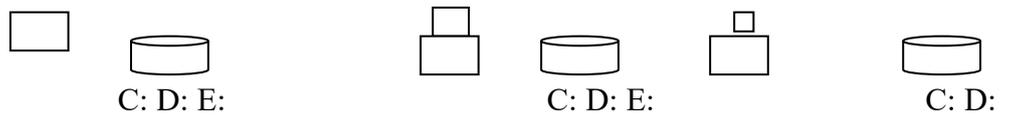
#### 8.1.4. Сетевая операционная система NET Ware

В любой физической конфигурации поддержка доступа от одного компьютера к другому, наличие и отсутствие выделенного компьютера, выполняется программой – сетевой ОС. Сетевая ОС является надстройкой по отношению к ОС отдельных компьютеров.

Для современных высококоразвитых ОС ПК вполне характерно наличие сетевых возможностей (например, OS/2, Windows 95-2000 - XP, Windows NT, UNIX).

Для обеспечения функционирования ЛС, использующей файловый сервер, в настоящее время разработан целый ряд сетевых ОС. Одной из них является NET Ware фирмы Novel.





Е:

Рис. 8.1. Локальная сеть ПК Novel Net Ware с моноканальной структурой.

ФС – файловый сервер с сетевой платой Ethernet и ядром Net Ware;

РС – рабочая станция с сетевой платой Ethernet и с оболочкой Net Ware.

С: D: E: – локальные диски.

*Основные функции сетевой ОС, реализуемые Net Ware:*

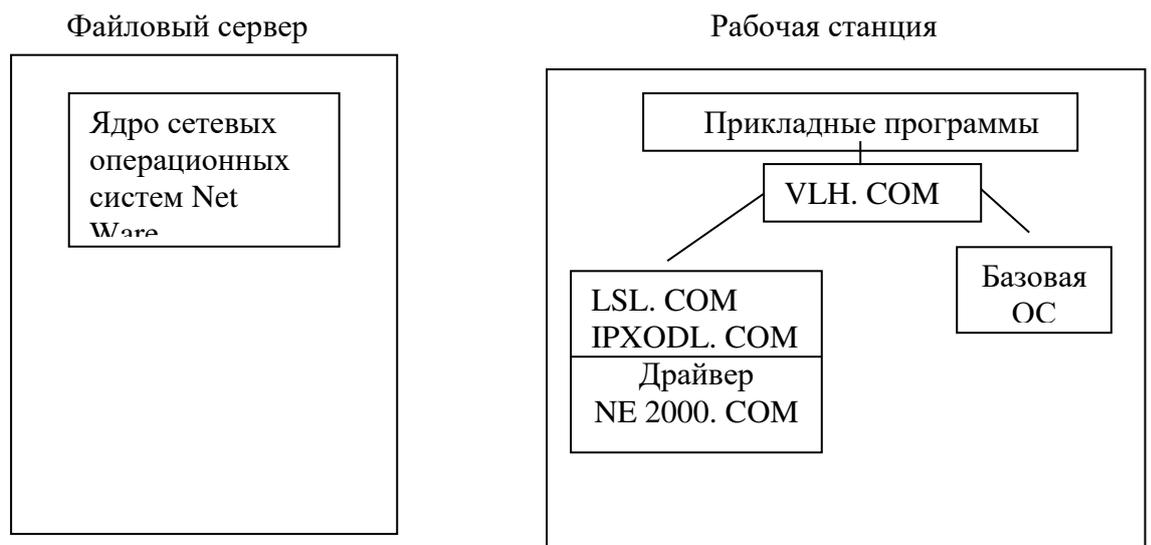
1. Разделение дисков файлового сервера, обеспечивающее надежное хранение и доступ к файлам;
2. Управление коммуникацией при различных топологиях подсетей и коммуникационных протоколах. Так, рабочие станции UNIX, как правило, используют протокол TCP/IP, а ПК сети Net Ware – IPX/SPX, и сетевая ОС обеспечивает поддержку этих протоколов.
3. Обеспечение работы на сервере программ, расширяющих и дополняющих функции самой сетевой ОС. Эти программы называются загружаемыми модулями Net Ware (N2H, Net Ware, loadable Modules).

## Структура сетевой ОС Net Ware

Сетевая ОС Net Ware координирует функционирование P.C. и регулирует процесс совместного использования сетевых ресурсов. Эти средства обеспечивают защиту и их целостность, контролируя право доступа к ним. Novel Net Ware, как и другие операционные системы, ориентирована на работу с различными сетевыми платами. Возможных видов плат очень много. Поддерживаются многие платы Ethernet, Token – Ring, ARCnet.

Схема размещения и взаимодействия компонентов программного обеспечения сетевой ОС Net Ware Novel представлена на рис. 8.2.

Net Ware 386 является сетевой ОС с централизованным управлением, т.е. в сети один или несколько компьютеров должны быть выделены в качестве файл-сервера. На файловых серверах работает ОС Net Ware. На P.C. (число которых может достигать нескольких сотен) должна быть загружена, т.н. клиентская часть Net Ware – специальная компонента системы (оболочка Net Ware ). Оболочка Net Ware состоит из модулей VLH.COM, LSL.COM и IPXODL.COM. При запросе некоторой сетевой услуги VLH.COM передает запрос другим модулям, которые отправляют его через драйвер сетевой карты на файловый сервер.



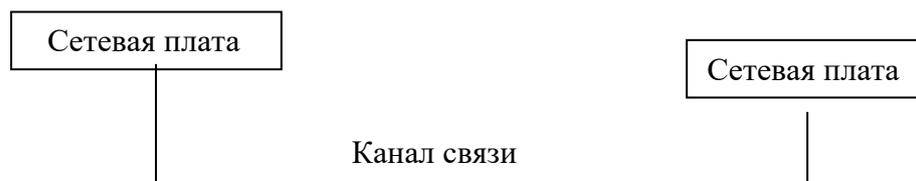


Рис. 8.2. Схема размещения и взаимодействия компонентов программного обеспечения сетевой ОС Net Ware Novel

### 8.1.6. Утилиты Net Ware

Утилиты разделяются на диалоговые и утилиты командной строки. Имеются варианты диалоговой утилиты выполняющейся под DOS и Windows. Развитый интерфейс DOS – утилит представляет собой выбор из меню. Графический интерфейс Windows – утилит существенно упрощает и ускоряет работу пользователя в сети. Утилиты командной строки позволяют задавать все параметры непосредственно при запуске утилиты в командной строке. Все утилиты обычно располагаются в сетевом каталоге SYSPUBLIC на винчестере файл-сервера.

## Лекция 8.2. Конфигурация (архитектура, топология) ЛВС

ЛВС в зависимости от назначения и технических решений могут иметь различные конфигурации (топологии), представленные на рис. 8.2.1-8.2.3 :

### 8.2.1. Кольцевая топология ЛВС

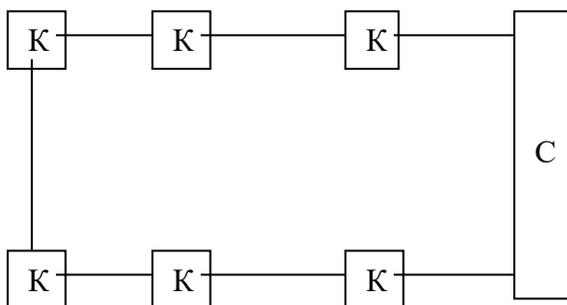


Рис. 8.2.1. Кольцевая топология

Кольцевая ЛС – информация передается по замкнутому кольцу. Каждый компьютер непосредственно связан с двумя соседними, хотя в принципе может связываться с любым абонентом сети. Каждый компьютер выступает в роли репитера (повторителя), усиливая сигналы и передавая их следующему компьютеру. Поэтому выход из строя хотя бы одного компьютера приводит к падению сети.

Способ передачи данных по кольцу называется передачей маркера. Маркер (token) – это специальная последовательность бит, передающаяся по сети. В каждой сети существует только один маркер. Маркер передает по кольцу последовательно от одного компьютера к другому до тех пор, пока его не захватит тот компьютер, который хочет передать данные. Передающий компьютер добавляет к маркеру данные и адрес получателя, и отправляет его дальше по кольцу. Данные проходят через каждый компьютер, пока не окажутся у того, чей адрес совпадает с адресом получателя. Затем принимающий компьютер посылает передающему сообщение, в котором подтверждает факт приема. Получив подтверждение, передающий компьютер восстанавливает маркер и возвращает его в сеть. Скорость движения маркера сопоставима со скоростью света. Так, в кольце диаметром 200 м маркер может циркулировать с частотой 477 376 об/с.

### 8.2.2. Радиальная топология ЛВС

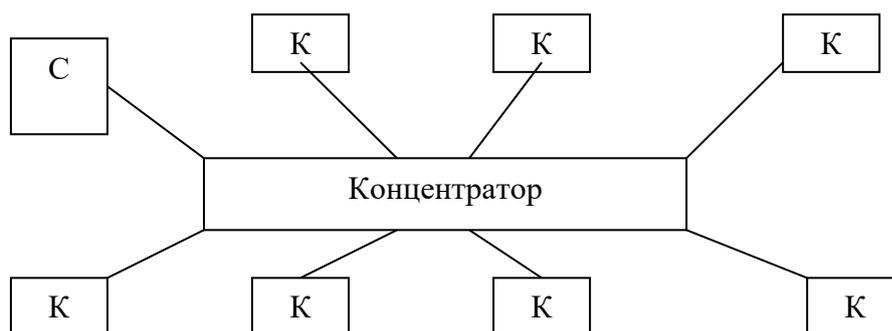


Рис. 8.2.2. Радиальная (звздообразная) топология

При топологии звезда (рис. 8.2.2) все компьютеры с помощью сегментов кабеля подключаются к центральному устройству, называемому концентратором (hub). Сигналы от передающего компьютера поступают через концентратор ко всем остальным. В сетях с топологией звезда, концентратор служит центральным узлом. Концентраторы делятся на активные (многопортовые повторители) и пассивные. Активные регенерируют и передают сигналы так же, как репитеры. Обычно они имеют от 8 до 12 портов для подключения компьютеров. Активные концентраторы следует подключать к электрической сети. К пассивным концентраторам относятся монтажные или коммутирующие панели. Они просто пропускают через себя сигнал, не усиливая и не восстанавливая его. Их не надо подключать к электрической сети.

Недостатки этой топологии: дополнительный расход кабеля, установка концентратора. Главное преимущество перед шинной топологией – более высокая надежность. Выход из строя одного или нескольких компьютеров на работу сети не влияет. Только неисправность концентратора приводит к падению сети.

### 8.2.3. Шинная топология

Компьютеры подключены к общему каналу (шине), через который могут обмениваться сообщениями (рис. 8.2.2).

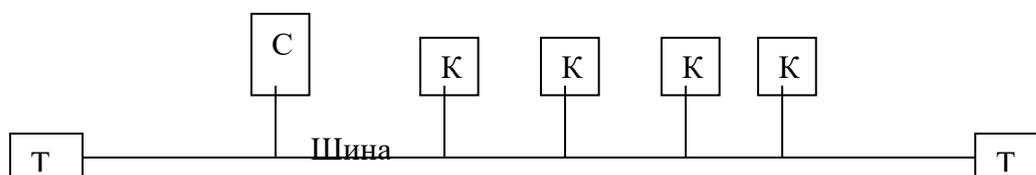


Рис. 8.2.3. Топология шина:  
С – сервер; К – компьютер; Т – терминатор.

Производительность такой сети зависит от количества компьютеров, подключенных к шине. Чем больше компьютеров, ожидающих передачи данных, тем медленнее сеть. На быстродействие также влияют:

- тип аппаратного обеспечения сетевых компьютеров;
- частота, с которой компьютеры передают данные;
- тип работающих сетевых приложений;
- тип сетевого кабеля;
- расстояние между компьютерами в сети.

Выход одного или нескольких компьютеров из строя никак не сказывается на работе сети.

Электрические сигналы распространяются по всему кабелю – от одного конца к другому. Сигналы, достигшие концов кабеля, отражаются от них. Возникает наложение сигналов, находящихся в разных фазах, и, как следствие, их искажение и ослабление. Поэтому, для гашения сигналов на концах кабеля устанавливают терминаторы. При разрыве кабеля или отсутствии терминаторов функционирование сети прекращается. Сеть падает.

### Ячеистая топология

Сеть с ячеистой топологией обладает высокой избыточностью и надежностью, так как каждый компьютер в такой сети соединен с каждым другим отдельным кабелем (рис. 8.2.4). Сигнал от компьютера-отправителя до компьютера-получателя может проходить по разным маршрутам, поэтому разрыв кабеля не сказывается на работоспособности сети. Основной недостаток – большие затраты на прокладку кабеля, что компенсируется высокой надежностью и простотой обслуживания. Ячеистая топология применяется в комбинации с другими топологиями при построении больших сетей.

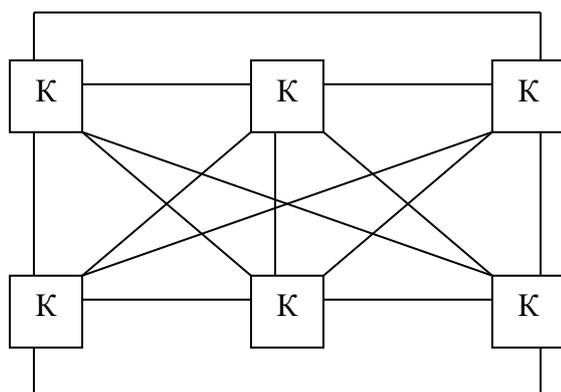


Рис. 8.2.4. Ячеистая топология.

Существуют комбинированные топологии. Чаще всего используются две комбинированные топологии: звезда-шина и звезда-кольцо.

**Звезда-шина** – несколько сетей с топологией звезда объединяются при помощи магистральной линейной шины (к концентратору подключены компьютеры, а сами концентраторы соединены шиной). Выход из строя одного компьютера не сказывается на работе всей сети, а сбой в работе концентратора влечет за собой отсоединение от сети только подключенных к нему компьютеров и концентраторов.

**Звезда-кольцо** – отличие состоит только в том, что концентраторы в звезде-кольце подсоединены к главному концентратору, внутри которого физически реализовано кольцо.

### Сетевые кабели

На сегодня подавляющая часть компьютерных сетей использует для соединения кабели. Это среда передачи сигналов между компьютерами.

В большинстве сетей применяются три основные группы кабелей.

**Коаксиальный кабель** – до недавнего времени был самым распространенным. Недорогой, легкий, гибкий, удобный, безопасный и простой в установке. Существует два типа коаксиальных кабелей: тонкий и толстый.

**Тонкий** – гибкий, диаметр 0,64 см (0,25"). Прост в применении и подходит практически для любого типа сети. Подключается непосредственно к плате сетевого адаптера. Передает сигнал на 185 м практически без затухания.

*Толстый* – жесткий, диаметр 1,27 см (0,5”). Его иногда называют стандартный Ethernet. Жила толще, затухание меньше. Передает сигнал без затухания на 500 м. Используют в качестве магистрали, соединяющей несколько небольших сетей.

**Витая пара** - это два переплетенных изолированных медных провода. Несколько витых пар проводов часто помещают в одну защитную оболочку. Переплетение проводов позволяет избавляться от электрических помех, наводимых соседними проводами и другими внешними источниками.

**Неэкранированная витая пара (UTP)** широко используется ЛВС, максимальная длина 100 м. UTP определена особым стандартом, в котором указаны нормативные характеристики кабелей для различных применений, что гарантирует единообразие продукции.

**Экранированная витая пара (STP)** помещена в медную оплетку. Кроме того, пары проводов обмотаны фольгой. Поэтому STP меньше подвержены влиянию электрических помех и может передавать сигналы с более высокой скоростью и на большие расстояния.

Преимущества – дешевизна, простота при подключении. Недостатки – нельзя использовать при передаче данных на большие расстояния с высокой скоростью.

**Оптоволоконный кабель.** В оптоволоконном кабеле цифровые данные распространяются по оптическим волокнам в виде модулированных световых импульсов. Это надежный способ передачи, так как электрические сигналы при этом не передаются. Следовательно, оптоволоконный кабель нельзя вскрыть и перехватить данные.

Оптоволоконные линии предназначены для перемещения больших объемов данных на очень высоких скоростях, так как сигнал в них практически не затухает и не искажается. Оптоволокно передает сигналы только в одном направлении, поэтому кабель состоит из двух волокон с отдельными коннекторами: одно – для передачи, другое – для приема.

Скорость передачи данных от 100 Мбит/с, теоретически – до 200 Гбит/с. Расстояние – многие километры. Существенным недостатком этой технологии является дороговизна и сложность в установке и подключении.

Типичная оптическая сеть состоит из лазерного передатчика света, мультиплексера/ демультиплексера для объединения оптических сигналов с разными длинами волн, усилителей оптических сигналов, демультиплексеров и приемников, преобразующих оптический сигнал обратно в электрический.

Для передачи по кабелю кодированных сигналов использую две технологии - *немодулированную* и *модулированную* передачу.

*Немодулированные* системы передают данные в виде цифровых сигналов, которые представляют собой дискретные электрические или световые импульсы.

*Модулированные* системы передают данные в виде аналогового сигнала (электрического или светового).

## **Беспроводная среда**

Беспроводная среда не означает полное отсутствие проводов в сети. Обычно беспроводные компоненты взаимодействуют с сетью, в которой в качестве среды передачи используется кабель. Такие сети называются *гибридными*.

Существуют следующие типы беспроводных сетей: ЛВС, расширенные ЛВС и мобильные сети (переносные компьютеры). Основные различия между ними – параметры передачи. ЛВС и расширенные ЛВС используют передатчики и приемники той организации, в которой функционирует сеть. Для переносных компьютеров средой передачи служат общедоступные сети (например, телефонная или Internet).

Вопросы для самоконтроля

1. Что такое компьютерная сеть?
2. Для чего создаются локальные сети ЭВМ?
3. Что такое сервер? Рабочая станция?
4. Какие сетевые технологии называются клиент - серверными?
5. Какие виды линий (накалов) используются для связи компьютеров в ЛВС?
6. Какие бывают конфигурации (топологии) ЛВС?
7. Какая сетевая ОС используется в ЛВС?

## Лекция 8.3. Глобальная сеть Интернет

### 8.3.1. Принципы построения сети Интернет

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. Решение этой задачи относится к области стандартизации и основано на так называемой модели OSI (модель взаимодействия открытых систем — Model of Open System Interconnections). Она создана на основе технических предложений Международного института стандартов ISO (International Standards Organization).

Согласно модели ISO/OSI архитектуру компьютерных сетей следует рассматривать на разных уровнях (общее число уровней — до семи). Самый верхний уровень — прикладной. На этом уровне пользователь взаимодействует с вычислительной системой. Самый нижний уровень — физический. Он обеспечивает обмен сигналами между устройствами. Обмен данными в системах связи происходит путем их перемещения с верхнего уровня на нижний, затем транспортировки и, наконец, обратным воспроизведением на компьютере клиента в результате перемещения с нижнего уровня на верхний. Простейшая модель обмена данными в компьютерной сети показана на рис. 8.3.1.

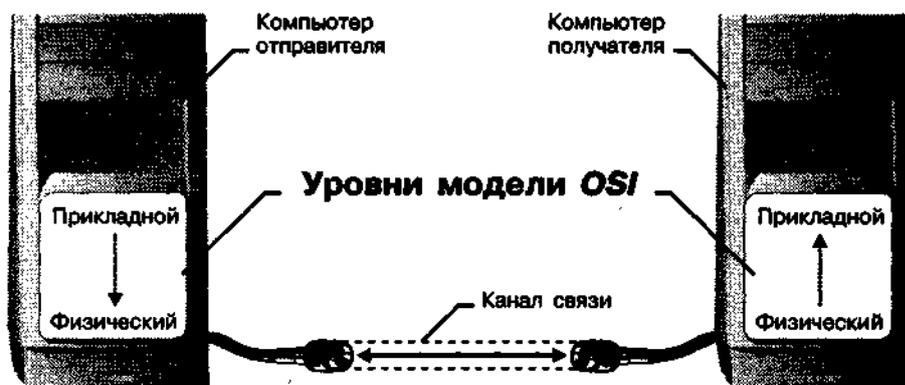


Рис. 8.3.1. Простейшая модель обмена данными в компьютерной сети

Для обеспечения необходимой совместимости на каждом из семи возможных уровней архитектуры компьютерной сети действуют специальные стандарты, называемые протоколами. Они определяют характер аппаратного взаимодействия компонентов сети (аппаратные протоколы) и характер взаимодействия программ и данных (программные протоколы). Физически функции поддержки протоколов исполняют аппаратные устройства (интерфейсы) и программные средства (программы поддержки протоколов). Программы, выполняющие поддержку протоколов, также называют протоколами. Так, например, 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 Windows 98 это стандартная программа *Прямое кабельное соединение*.

Отдельные локальные сети могут объединяться в глобальные сети. В этом случае локальные сети связывают между собой с помощью любых традиционных каналов связи

Уровень	Аналогия
Прикладной уровень	Письмо написано на бумаге. Определено его содержание

(кабельных, спутниковых, радиорелейных и т. п.). При соблюдении специальных условий для этой цели могут быть использованы даже телефонные каналы, хотя они в наименьшей степени удовлетворяют требованиям цифровой связи.

Для связи между собой нескольких локальных сетей, работающих по разным протоколам, служат специальные средства, называемые шлюзами. Шлюзы могут быть как аппаратными, так и программными. Например, это может быть специальный компьютер (шлюзовый сервер), а может быть и компьютерная программа. В последнем случае компьютер может выполнять не только функцию шлюза, но и какие-то иные функции, типичные для рабочих станций.

При подключении локальной сети предприятия к глобальной сети важную роль играет понятие сетевой безопасности. В частности, должен быть ограничен доступ в локальную сеть для посторонних лиц извне, а также ограничен выход за пределы локальной сети для сотрудников предприятия, не имеющих соответствующих прав. Для обеспечения сетевой безопасности между локальной и глобальной сетью устанавливают так называемые брандмауэры. Брандмауэром может быть специальный компьютер или компьютерная программа, препятствующая несанкционированному перемещению данных между сетями.

### **8.3.2. Сетевые службы**

*Модель взаимодействия открытых систем.* Выше мы упомянули о том, что согласно рекомендациям Международного института стандартизации ISO системы компьютерной связи рекомендуется рассматривать на семи разных уровнях (таблица 1).

Таблица 8. 1 . Уровни модели связи

Уровень представления	Письмо запечатано в конверт. Конверт заполнен. Наклеена марка, Клиентом соблюдены необходимые требования протокола доставки
Сеансовый уровень	Письмо опущено в почтовый ящик. Выбрана служба доставки (письмо можно было бы запечатать в бутылку и бросить в реку, но избрана другая служба)
Транспортный уровень	Письмо доставлено на почтамт. Оно отделено от писем, с доставкой которых местная почтовая служба справилась бы самостоятельно
Сетевой уровень	После сортировки письмо уложено в мешок. Появилась новая единица доставки — мешок
Уровень соединения	Мешки писем уложены в вагон. Появилась новая единица доставки — вагон
Физический уровень	Вагон прицеплен к локомотиву. Появилась новая единица доставки — состав. За доставку взялось другое ведомство, действующее по другим протоколам

Из таблицы видно, что каждый новый уровень все больше и больше увеличивает функциональность системы связи. Проведем аналогию с почтовой службой. Местная почтовая служба работает с письмами, бандеролями и посылками. Почтово-багажная служба занимается еще и доставкой грузов. Вагоны перевозят не только почту, но и людей. По рельсам ходят не только почтово-пассажирские поезда, но и грузовые составы и т. д. То есть чем выше уровень в модели связи, тем больше различных, функциональных служб его используют.

Возвращаясь к системам компьютерной связи, рассмотрим, как в модели ISO/OSI происходит обмен данными между пользователями, находящимися на разных континентах.

1. На прикладном уровне с помощью специальных приложений пользователь создает документ (сообщение, рисунок и т. п.).

2. На уровне представления операционная система его компьютера фиксирует, где находятся созданные данные (в оперативной памяти, в файле на жестком диске и т. п.), и обеспечивает взаимодействие со следующим уровнем.

3. На сеансовом уровне компьютер пользователя взаимодействует с локальной или глобальной сетью. Протоколы этого уровня проверяют права пользователя на «выход в эфир» и передают документ к протоколам транспортного уровня.

4. На транспортном уровне документ преобразуется в ту форму, в которой положено передавать данные в используемой сети. Например, он может нарезаться на небольшие пакеты стандартного размера.

5. Сетевой уровень определяет маршрут движения данных в сети. Так, например, если на транспортном уровне данные были «нарезаны» на пакеты, то на сетевом уровне каждый пакет должен получить адрес, по которому он должен быть доставлен независимо от прочих пакетов.

6. Уровень соединения необходим для того, чтобы промодулировать сигналы, циркулирующие на физическом уровне, в соответствии с данными, полученными с сетевого уровня. Например, в компьютере эти функции выполняет сетевая карта или модем.

7. Реальная передача данных происходит на физическом уровне. Здесь нет ни документов, ни пакетов, ни даже байтов — только биты, то есть, элементарные единицы представления данных. Восстановление документа из них произойдет постепенно, при переходе с нижнего на верхний уровень на компьютере клиента.

Средства физического уровня лежат за пределами компьютера. В локальных сетях это оборудование самой сети. При удаленной связи с использованием телефонных модемов это линии телефонной связи, коммутационное оборудование телефонных станций и т. п.

На компьютере получателя информации происходит обратный процесс преобразования данных от битовых сигналов до документа.

**Особенности виртуальных соединений.** Разные уровни протоколов сервера и клиента не взаимодействуют друг с другом напрямую, но они взаимодействуют через физический уровень. Постепенно переходя с верхнего уровня на нижний, данные непрерывно преобразуются, «обрастают» дополнительными данными, которые анализируются протоколами соответствующих уровней на сопредельной стороне. Это и создает эффект виртуального взаимодействия уровней между собой. Однако, несмотря на виртуальность, это все-таки соединения, через которые тоже проходят данные.

На использовании виртуальных соединений основаны такие позитивные свойства электронных систем связи, как возможность работать по одному физическому каналу сразу с несколькими серверами. Но на них же основаны и такие негативные средства, как «троянские программы». Троянская программа — разновидность «компьютерного вируса», создающая во время сеансов связи виртуальные соединения для передачи данных о компьютере, на котором установлена. Среди этих данных может быть парольная информация, информация о содержании жесткого диска и т. п. В отличие от обычных компьютерных вирусов троянские программы не производят разрушительных действий на компьютере и потому лучше маскируются.

**Сетевые службы.** На виртуальных соединениях основаны все службы современного *Интернета*. Так, например, пересылка сообщения от сервера к клиенту может проходить через десятки различных компьютеров. Это совсем не означает, что на каждом компьютере сообщение должно пройти через все уровни — ему достаточно «подняться» до сетевого уровня, (определяющего адресацию) при приеме и вновь «опуститься» до физического уровня при передаче. В данном случае служба передачи сообщений основывается на виртуальном соединении сетевого уровня и соответствующих ему протоколах (рис. 2).

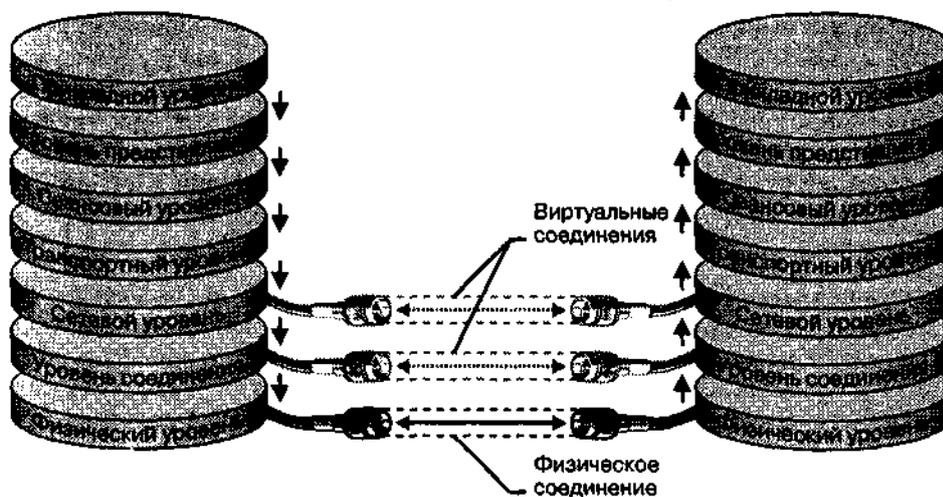


Рис. 8.2. Простейшая модель службы передачи сообщений

### 8.3.3. Основные понятия Интернет

В дословном переводе на русский язык интернет — это межсеть, то есть в узком смысле слова Интернет — это объединение сетей. Однако в последние годы у этого слова появился и более широкий смысл: Всемирная компьютерная сеть. Интернет можно рассматривать в физическом смысле как несколько миллионов компьютеров, связанных друг с другом всевозможными линиями связи, однако такой «физический» взгляд на Интернет слишком узок. Лучше рассматривать Интернет как некое информационное пространство.

Интернет — это не совокупность прямых соединений между компьютерами. Так, например, если два компьютера, находящиеся на разных континентах, обмениваются данными в Интернете, это совсем не значит, что между ними действует одно прямое или виртуальное соединение. Данные, которые они посылают друг другу, разбиваются на пакеты, и даже в одном сеансе связи разные пакеты одного сообщения могут пройти разными маршрутами. Какими бы маршрутами ни двигались пакеты данных, они все равно достигнут пункта назначения и будут собраны вместе в цельный документ. При этом данные, отправленные позже, могут приходиться раньше, но это не мешает правильно собрать документ, поскольку каждый пакет имеет свою маркировку.

Таким образом, Интернет представляет собой как бы «пространство», внутри которого осуществляется непрерывная циркуляция данных. Данные перемещаются между компьютерами, составляющими узлы сети, и какое-то время хранятся на их жестких дисках.

### 8.3.4. Теоретические основы Интернета

Ранние эксперименты по передаче и приему информации с помощью компьютеров начались еще в 50-х годах и имели лабораторный характер. Лишь в конце 60-х годов на средства Агентства Перспективных Разработок министерства обороны США (DARPA) была создана первая сеть национального масштаба. По имени агентства она получила название ARPANET. Эта сеть связала несколько крупных научных, исследовательских и образовательных центров. Ее основной задачей стала координация групп коллективов, работающих над едиными научно-техническими проектами, а основным назначением стал обмен электронной почтой и файлами с научной и проектно-конструкторской документацией. Сеть ARPANET заработала в 1969 году.

По-настоящему рождением Интернета принято считать 1983 год. В этом году произошли революционные изменения в программном обеспечении компьютерной связи. Днем рождения Интернета в современном понимании этого слова стала дата стандартизации протокола связи TCP/IP, лежащего в основе Всемирной сети по нынешний день.

В современном понимании TCP/IP — это не один сетевой протокол, а два протокола, лежащих на разных уровнях (это так называемый стек протоколов). Протокол TCP — протокол транспортного уровня. Он управляет тем, как происходит передача информации. Протокол IP — адресный. Он принадлежит сетевому уровню и определяет, куда происходит передача.

**Протокол TCP.** Согласно протоколу TCP, отправляемые данные «нарезаются» на небольшие пакеты, после чего каждый пакет маркируется таким образом, чтобы в нем были данные, необходимые для правильной сборки документа на компьютере получателя.

Два компьютера, связанные между собой одним физическим соединением, могут поддерживать одновременно несколько TCP- соединений. Так, например, два промежуточных сетевых сервера могут одновременно по одной линии связи передавать друг другу в обе стороны множество TCP- пакетов от многочисленных клиентов.

**Протокол IP** (Internet Protocol). Его суть состоит в том, что у каждого участника Всемирной сети должен быть свой уникальный адрес (IP-адрес). Без этого нельзя говорить о точной доставке TCP- пакетов на нужное рабочее место. Этот адрес выражается четырьмя байтами, например: 195.38.46.11. Структура IP-адреса организована так, что каждый компьютер, через который проходит какой-либо TCP- пакет, может по этим четырем числам определить, кому из ближайших «соседей» надо переслать пакет, чтобы он оказался «ближе» к получателю. В результате конечного числа перебросок TCP- пакет достигает адресата. В данном случае оценивается не географическая «близость». В расчет принимаются условия связи и пропускная способность линии. Два компьютера, находящиеся на разных континентах, но связанные высокопроизводительной линией

космической связи, считаются более «близкими» друг к другу, чем два компьютера из соседних поселков, связанные простым телефонным проводом. Решением вопросов, что считать «ближе», а что «дальше», занимаются специальные средства - маршрутизаторы. Роль маршрутизатора в сети может выполнять как специализированный компьютер, так и специальная программа, работающая на узловом сервере сети.

Поскольку один байт содержит до 256 различных значений, то теоретически с помощью четырех байтов можно выразить более четырех миллиардов уникальных IP-адресов ( $256^4$  за вычетом некоторого количества адресов, используемых в качестве служебных). На практике же из-за особенностей адресации к некоторым типам локальных сетей количество возможных адресов составляет порядка двух миллиардов, но и это по современным меркам достаточно большая величина.

### **8.3.5. Сервисы Интернет**

Когда говорят о работе в Интернете или об использовании Интернета, то на самом деле речь идет не об Интернете в целом, а только об одной или нескольких из его многочисленных служб. В зависимости от конкретных целей и задач клиенты Сети используют те службы, которые им необходимы.

Разные службы имеют разные протоколы. Они называются прикладными протоколами. Их соблюдение обеспечивается и поддерживается работой специальных программ. Таким образом, чтобы воспользоваться какой-то из служб Интернета, необходимо установить на компьютере программу, способную работать по протоколу данной службы. Такие программы называют клиентскими или просто клиентами.

Так, например, для передачи файлов в Интернете используется специальный прикладной протокол FTP (File Transfer Protocol). Соответственно, чтобы получить из Интернета файл, необходимо:

- иметь на компьютере программу, являющуюся клиентом FTP (FTP- клиент);
- установить связь с сервером, предоставляющим услуги FTP (FTP- сервером).

Другой пример: чтобы воспользоваться электронной почтой, необходимо соблюсти протоколы отправки и получения сообщений. Для этого надо иметь программу (почтовый клиент) и установить связь с почтовым сервером. Так же обстоит дело и с другими службами.

**Терминальный режим.** Исторически одной из ранних является служба удаленного управления компьютером Telnet. Подключившись к удаленному компьютеру по протоколу этой службы, можно управлять его работой. Такое управление еще называют консольным или терминальным. В прошлом эту службу широко использовали для проведения сложных математических расчетов на удаленных вычислительных центрах. Так, например, если для очень сложных вычислений на персональном компьютере требовались недели непрерывной работы, а на удаленной супер-ЭВМ всего несколько минут, то персональный компьютер применяли для удаленного ввода данных в ЭВМ и для приема полученных результатов.

В наши дни в связи с быстрым увеличением мощности персональных компьютеров необходимость в подобной услуге сократилась, но, тем не менее, службы Telnet в Интернете продолжают существовать. Часто протоколы Telnet применяют для дистанционного управления техническими объектами, например телескопами, видеокамерами, промышленными роботами.

**Электронная почта (E-Mail).** Эта служба также является одной из наиболее ранних. Ее обеспечением в Интернете занимаются специальные почтовые серверы. Обратите внимание на то, что когда мы говорим о каком-либо сервере, не имеется в виду, что это специальный выделенный компьютер. Здесь и далее под сервером может пониматься программное обеспечение. Таким образом, один узловой компьютер Интернета может выполнять функции нескольких серверов и обеспечивать работу

различных служб, оставаясь при этом универсальным компьютером, на котором можно выполнять и другие задачи, характерные для средств вычислительной техники.

Почтовые серверы получают сообщения от клиентов и пересылают их по цепочке к почтовым серверам адресатов, где эти сообщения накапливаются. При установлении соединения между адресатом и его почтовым сервером происходит автоматическая передача поступивших сообщений на компьютер адресата.

Почтовая служба основана на двух прикладных протоколах *SMTP* и *POP3*. По первому происходит отправка корреспонденции с компьютера на сервер, а по второму — прием поступивших сообщений. Существует большое разнообразие клиентских почтовых программ. К ним относится, например, программа *Microsoft Outlook Express*, входящая в состав операционной системы Windows 98 как стандартная. Более мощная программа, интегрирующая в себе кроме поддержки электронной почты и другие средства делопроизводства, *Microsoft Outlook 2000*, входит в состав известного пакета *Microsoft office 2000*. Из специализированных почтовых программ хорошую популярность имеют программы *The Bat!* и *Eudora Pro*.

**Списки рассылки (Mail List).** Обычная электронная почта предполагает наличие двух партнеров по переписке. Если же партнеров нет, то достаточно большой поток почтовой информации в свой адрес можно обеспечить, подписавшись на списки рассылки. Это специальные тематические серверы, собирающие информацию по определенным темам и переправляющие ее подписчикам в виде сообщений электронной почты.

Темами списков рассылки может быть что угодно, например вопросы, связанные с изучением иностранных языков, научно-технические обзоры, презентация новых программных и аппаратных средств вычислительной техники (рис. 8.3). Большинство телекомпаний создают списки рассылки на своих узлах, через которые рассылают клиентам аннотированные обзоры телепрограмм. Списки рассылки позволяют эффективно решать вопросы регулярной доставки данных.



Рис. 8.3. Список рассылки, посвященный отечественным железным дорогам

**Служба телеконференций (Usenet).** Служба телеконференций похожа на циркулярную рассылку электронной почты, в ходе которой одно сообщение отправляется

не одному корреспонденту, а большой группе (такие группы называются телеконференциями или группами новостей).

Обычное сообщение электронной почты пересылается по узкой цепочке серверов от отправителя к получателю. При этом не предполагается его хранение на промежуточных серверах. Сообщения, направленные на сервер группы новостей, отправляются с него на все серверы, с которыми он связан, если на них данного сообщения еще нет. Далее процесс повторяется. Характер распространения каждого отдельного сообщения напоминает лесной пожар.

На каждом из серверов поступившее сообщение хранится ограниченное время (обычно неделю), и все желающие могут в течение этого времени с ним ознакомиться. Распространяясь во все стороны, менее чем за сутки сообщения охватывают весь земной шар. Далее распространение затухает, поскольку на сервер, который уже имеет данное сообщение, повторная передача производится не может.

Ежедневно в мире создается порядка миллиона сообщений для групп новостей. Выбрать в этом массиве действительно полезную информацию практически невозможно. Поэтому вся система телеконференций разбита на тематические группы. Сегодня в мире насчитывают порядка 50 000 тематических групп новостей. Они охватывают большинство тем, интересующих массы. Особой популярностью пользуются группы, посвященные вычислительной технике.

Огромный объем сообщений в группах новостей значительно затрудняет их целенаправленный мониторинг, поэтому в некоторых группах производится предварительный «отсев» бесполезной информации (в частности, рекламной), не относящейся к теме конференции. Такие конференции называют модерлируемыми. В качестве модератора может выступать не только человек, но и программа, фильтрующая сообщения по определенным ключевым словам. В последнем случае говорят об *автоматической модерации*.

Для работы со службой телеконференций существуют специальные клиентские программы. Так, например, приложение Microsoft Outlook Express, указанное выше как почтовый клиент, позволяет работать также и со службой телеконференций. Для начала работы надо настроить программу на взаимодействие с сервером групп новостей, оформить «подписку» на определенные группы и- периодически, как и электронную почту, получать все сообщения, проходящие по теме этой группы. В данном случае слово «подписка» не предполагает со стороны клиента никаких обязательств или платежей — это просто указание серверу о том, что сообщения по указанным темам надо доставлять, а по прочим — нет. Отменить подписку или изменить ее состав можно в любой удобный момент.

**Служба World Wide Web (WWW).** Безусловно, это самая популярная служба современного Интернета. Ее нередко отождествляют с Интернетом, хотя на самом деле это лишь одна из его многочисленных служб.

*World Wide Web* — это единое информационное пространство, состоящее из сотен миллионов взаимосвязанных электронных документов, хранящихся на Web-серверах. Отдельные документы, составляющие пространство Web, называют Web-страницами. Группы тематически объединенных Web-страниц называют Web-узлами (жаргонный термин — Web-сайт или просто сайт). Один физический Web-сервер может содержать достаточно много Web-узлов, каждому из которых, как правило, отводится отдельный каталог на жестком диске сервера.

От обычных текстовых документов Web-страницы отличаются тем, что они оформлены без привязки к конкретному носителю. Электронные Web-документы предназначены для просмотра на экране компьютера, причем заранее не известно на каком. Неизвестны ни размеры экрана, ни параметры цветового и графического разрешения, неизвестна даже операционная система, с которой работает компьютер клиента. Поэтому Web-документы не могут иметь «жесткого» форматирования.

Оформление выполняется непосредственно во время их воспроизведения на компьютере клиента и происходит оно в соответствии с настройками программы, выполняющей просмотр.

Программы для просмотра Web-страниц называют браузерами. В литературе также можно встретить «неустоявшиеся» термины браузер или обозреватель. Во всех случаях речь идет о некотором средстве просмотра Web-документов.

Браузер выполняет отображение документа на экране, руководствуясь командами, которые автор документа внедрил в его текст (если автор применяет автоматические средства подготовки Web-документов, необходимые команды внедряются автоматически). Такие команды называются тегами. От обычного текста они отличаются тем, что заключены в угловые скобки. Большинство тегов используются парами: открывающий тег и закрывающий. Закрывающий тег начинается с символа «/».

<CENTER> Этот текст должен выравниваться по центру экрана </CENTER>

<LEFT> Этот текст выравнивается по левой границе экрана </LEFT>

<RIGHT> Этот текст выравнивается по правой границе экрана </RIGHT>

Сложные теги имеют кроме ключевого слова дополнительные атрибуты и параметры, детализирующие способ их применения. Правила записи тегов содержатся в спецификации особого языка разметки, близкого к языкам программирования. Он называется языком разметки гипертекста — HTML (HyperText Markup Language). Таким образом, документ представляет собой обычный текстовый документ, размеченный тегами HTML. Такие документы также называют HTML - документами или документами в формате HTML.

При отображении HTML-документа на экране с помощью браузера теги не показываются, и мы видим только текст, составляющий документ. Однако оформление этого текста (выравнивание, цвет, размер и начертание шрифта, и прочее) выполняется в соответствии с тем, какие теги имплантированы в текст документа.

Существуют специальные теги для внедрения графических и мультимедийных объектов (звук, музыка, видеоклипы). Встретив, такой тег, обозреватель делает запрос к серверу на доставку файла, связанного с тегом, и воспроизводит его в соответствии с заданными

Наиболее важной чертой Web-страниц, реализуемой с помощью тегов HTML, являются гипертекстовые ссылки. С любым фрагментом текста или, например, с рисунком с помощью тегов можно связать иной Web-документ, то есть установить гиперссылку. В этом случае при щелчке левой кнопкой мыши на тексте или рисунке, являющемся гиперссылкой, отправляется запрос на доставку нового документа. Этот документ, в свою очередь, тоже может иметь гиперссылки на другие документы.

Таким образом, совокупность огромного числа гипертекстовых электронных документов, хранящихся на серверах WWW, образует своеобразное гиперпространство документов, между которыми возможно перемещение. Произвольное перемещение между документами в Web-пространстве называют Web-серфингом (выполняется с целью ознакомительного просмотра). Целенаправленное перемещение между Web-документами называют Web-навигацией (выполняется с целью поиска нужной информации).

Гипертекстовая связь между сотнями миллионов документов, хранящихся на физических серверах Интернета, является основой существования логического пространства Word Wide Web. Однако такая связь не могла бы существовать, если бы каждый документ в этом пространстве не обладал своим уникальным адресом. Адрес любого файла во всемирном масштабе определяется унифицированным указателем ресурса — URL.

Адрес URL состоит из трех частей.

1. Указание службы, которая осуществляет доступ к данному ресурсу (обычно обозначается именем прикладного протокола, соответствующего данной службе). Так, например, для службы WWW прикладным является протокол HTTP (HyperText Transfer

Protocol — протокол передачи гипертекста). После имени протокола ставится двоеточие (:), и два знака «/» (косая черта):

**http://...**

2. Указание доменного имени компьютера (сервера), на котором хранится данный ресурс:

**http://www.abcde.com...**

3. Указания полного пути доступа к файлу на данном компьютере. В качестве разделителя используется символ «/» (косая черта):

**http://www.abcde.com/Files/New/abcdefg.zip**

При записи URL-адреса важно точно соблюдать регистр символов. В отличие от правил работы в MS-DOS и Windows, в Интернете строчные и прописные символы считаются разными.

Именно в форме URL и связывают адрес ресурса с гипертекстовыми ссылками на Web-страницах. При щелчке на гиперссылке браузер посылает запрос для поиска и доставки ресурса, указанного в ссылке. Если по каким-то причинам он не найден, выдается сообщение о том, что ресурс недоступен (возможно, что сервер временно отключен или изменился адрес ресурса).

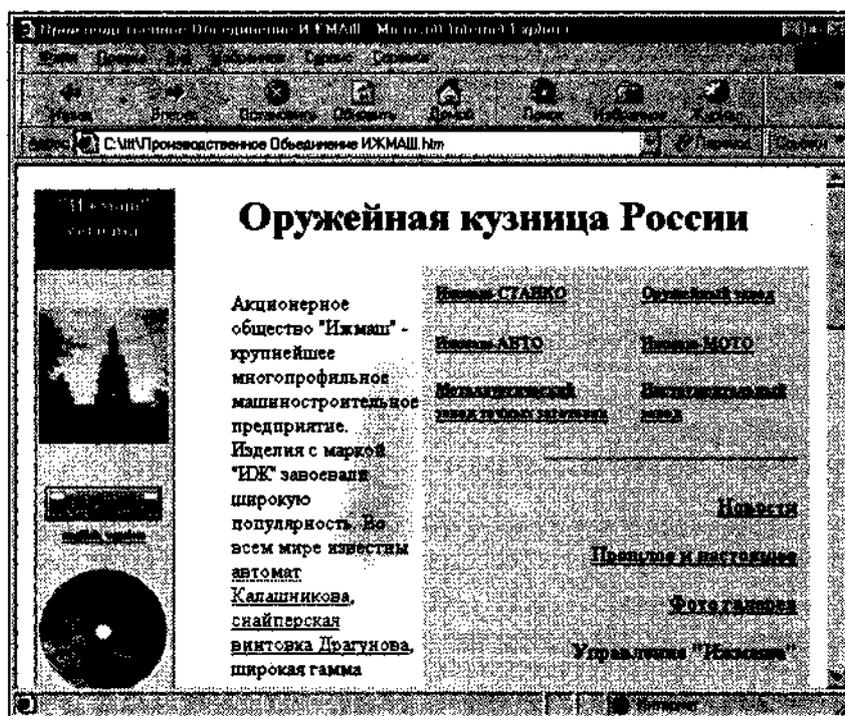


Рис. 8.4. Web-страница отечественного предприятия

**Служба имен доменов (DNS).** Когда мы говорили о протоколах Интернета, то сказали, что адрес любого компьютера или любой локальной сети в Интернете может быть выражен четырьмя байтами, например так:

195.28.132.97

А только что мы заявили, что каждый компьютер имеет уникальное доменное имя, например такое:

[www.abcdef.com](http://www.abcdef.com)

Это просто две разных формы записи адреса одного и того же сетевого компьютера. Человеку неудобно работать с числовым представлением IP-адреса, зато доменное имя запоминается легко, особенно если учесть, что, как правило, это имя имеет содержание. Например, Web-сервер компании Microsoft имеет имя [www.microsoft.com](http://www.microsoft.com), а Web-сервер компании «Космос ТВ» имеет имя [www.kosmostv.ru](http://www.kosmostv.ru) (суффикс .ru в конце

имени говорит о том, что сервер компании принадлежит российскому сектору Интернета). Нетрудно «реконструировать» и имена для других компаний.

С другой стороны, автоматическая работа серверов сети организована с использованием четырехзначного числового адреса. Благодаря ему, промежуточные серверы могут осуществлять передачу запросов и ответов в нужном направлении, не зная, где конкретно находятся отправитель и получатель. Поэтому необходим перевод доменных имен в связанные с ними IP-адреса. Этим и занимаются серверы службы имен доменов DNS. Наш запрос на получение одной из страниц сервера [www.abcde.com](http://www.abcde.com) сначала обрабатывается сервером DNS, и далее он направляется по IP-адресу, а не по доменному имени.

**Служба передачи файлов (FTP).** Прием и передача файлов составляют значительный процент от прочих Интернет - услуг. Необходимость в передаче файлов возникает, например, при приеме файлов программ, при пересылке крупных документов (например, книг), а также при передаче архивных файлов, в которых запакованы большие объемы информации.

Служба FTP имеет свои серверы в мировой сети, на которых хранятся архивы данных. Со стороны клиента для работы с серверами FTP может быть установлено специальное программное обеспечение, хотя в большинстве случаев браузеры WWW обладают встроенными возможностями для работы и по протоколу FTP.

Протокол FTP работает одновременно с двумя TCP - соединениями между сервером и клиентом. По одному соединению идет передача данных, а второе соединение используется как управляющее. Протокол FTP также предоставляет серверу средства для идентификации обратившегося клиента. Этим часто пользуются коммерческие серверы и серверы ограниченного доступа, поставляющие информацию только зарегистрированным клиентам, — они выдают запрос на ввод имени пользователя и связанного с ним пароля. Однако существуют и десятки тысяч FTP-серверов с анонимным доступом для всех желающих. В этом случае в качестве имени пользователя надо ввести слово: `anonymous`, а в качестве пароля задать адрес электронной почты. В большинстве случаев программы-клиенты FTP делают это автоматически.

**IRC.** Служба IRC (Internet Relay Chat) предназначена для прямого общения нескольких человек в режиме реального времени. Иногда службу IRC называют чат - конференциями или просто чатом. В отличие от системы телеконференций, в которой общение между участниками обсуждения темы открыто всему миру, в системе IRC общение происходит только в пределах одного канала, в работе которого принимают участие обычно лишь несколько человек. Каждый пользователь может создать собственный канал и пригласить в него участников «беседы» или присоединиться к одному из открытых в данный момент каналов.

Существует несколько популярных клиентских программ для работы с серверами и сетями, поддерживающими сервис IRC. Одна из наиболее популярных, программа `mIRC.exe`.

**ICQ.** Эта служба предназначена для поиска сетевого IP-адреса человека, подключенного в данный момент к Интернету. Необходимость в подобной услуге связана с тем, что большинство пользователей не имеют постоянного IP-адреса. Название службы является антонимом выражения `I seek you` — я тебя ищу. Для пользования этой службой надо зарегистрироваться на ее центральном сервере (<http://www.icq.com>) и получить персональный идентификационный номер UIN (Universal Internet Number). Данный номер можно сообщить партнерам по контактам, и тогда служба ICQ приобретает характер Интернет - пейджера. Зная номер UIN партнера, но, не зная его текущий IP-адрес, можно через центральный сервер службы отправить ему сообщение с предложением установить соединение.

Как было указано выше, каждый компьютер, подключенный к Интернету, должен иметь четырехзначный IP-адрес. Этот адрес может быть постоянным или динамически

временным. Те компьютеры, которые включены в Интернет на постоянной основе, имеют постоянные IP-адреса. Большинство же пользователей подключаются к Интернету лишь на время сеанса. Им выдается динамический IP-адрес, действующий только в течение данного сеанса. Этот адрес выдает тот сервер, через который происходит подключение. В разных сеансах динамический IP-адрес может быть различным, причем заранее неизвестно каким.

При каждом подключении к Интернету программа ICQ, установленная на нашем компьютере, определяет текущий IP-адрес и сообщает его центральной службе, которая, в свою очередь, оповещает наших партнеров по контактам. Далее наши партнеры (если они тоже являются клиентами данной службы) могут установить с нами прямую связь. Программа предоставляет возможность выбора режима связи («готов к контакту»; «прошу не беспокоить, но готов принять срочное сообщение»; «закрыт для контакта» и т. п.). После установления контакта связь происходит в режиме, аналогичном сервису IRC.

Вводить собственный IP-адрес для настройки программы не надо. Сервер поставщика услуг выделит его автоматически на время проведения сеанса работы.

#### **Вопросы для самоконтроля**

1. Что такое глобальная компьютерная сеть Internet?
2. Как работает электронная почта?
3. Какие виды обмена информацией приняты в электронной почте?
4. Каков электронный адрес нашего вуза и что означают его составляющие?
5. Какие виды работ поддерживает сеть Internet?
6. Что такое телеконференции?
7. Как можно передать файлы по Internet?
8. Что такое World-Wide-Web?
9. Что такое гиперссылка?
10. Что представляет собой гипертекст?

**Основы защиты информации и сведений, составляющих государственную тайну. Методы защиты информации**

### **Лекция 9.1. Информационная безопасность (ИБ) и ее составляющие**

В вычислительной технике понятие безопасности является весьма широким. Оно подразумевает и надежность работы компьютера, и сохранность ценных данных, и защиту информации от внесения в нее изменений неуполномоченными лицами, и сохранение тайны переписки в электронной связи. Разумеется, во всех цивилизованных странах на страже безопасности граждан стоят законы, но в сфере вычислительной техники правоприменительная практика пока развита недостаточно, а законотворческий процесс не успевает за развитием технологий, поэтому надежность работы компьютерных систем во многом опирается на меры самозащиты.

#### **Компьютерные вирусы**

Компьютерный вирус — это программный код, встроенный в другую программу, или в документ, или в определенные области носителя данных и предназначенный для выполнения несанкционированных действий на несущем компьютере.

Основными типами компьютерных вирусов являются:

- программные вирусы;
- загрузочные вирусы;
- макровирусы.

К компьютерным вирусам примыкают и так называемые троянские кони (троянские программы, троянцы).

**Программные вирусы.** Программные вирусы — это блоки программного кода, целенаправленно внедренные внутрь других прикладных программ. При запуске программы, несущей вирус, происходит запуск имплантированного в нее вирусного кода. Работа этого кода вызывает скрытые от пользователя изменения в файловой системе жестких дисков и/или в содержании других программ. Так, например, вирусный код может воспроизводить себя в теле других программ — этот процесс называется размножением. По прошествии определенного времени, создав достаточное количество копий, программный вирус может перейти к разрушительным действиям — нарушению работы программ и операционной системы, удалению информации, хранящейся на жестком диске. Этот процесс называется вирусной атакой.

Самые разрушительные вирусы могут инициировать форматирование жестких дисков. Поскольку форматирование диска — достаточно продолжительный процесс, который не должен пройти незамеченным со стороны пользователя, во многих случаях программные вирусы ограничиваются уничтожением данных только в системных секторах жесткого диска, что эквивалентно потере таблиц файловой структуры. В этом случае данные на жестком диске остаются нетронутыми, но воспользоваться ими без применения специальных средств нельзя, поскольку неизвестно, какие сектора диска каким файлам принадлежат. Теоретически восстановить данные в этом случае можно, но трудоемкость этих работ исключительно высока.

Считается, что никакой вирус не в состоянии вывести из строя аппаратное обеспечение компьютера. Однако бывают случаи, когда аппаратное и программное обеспечение настолько взаимосвязаны, что программные повреждения приходится устранять заменой аппаратных средств. Так, например, в большинстве современных материнских плат базовая система ввода-вывода (BIOS) хранится в перезаписываемых постоянных запоминающих устройствах (так называемая флэш-память). Возможность перезаписи информации в микросхеме флэш-памяти используют некоторые программные вирусы для уничтожения данных BIOS. В этом случае для восстановления работоспособности компьютера требуется либо замена микросхемы, хранящей BIOS, либо ее перепрограммирование на специальных устройствах, называемых программаторами.

Программные вирусы поступают на компьютер при запуске непроверенных программ, полученных на внешнем носителе (гибкий диск, компакт-диск и т. п.) или принятых из Интернета. Особое внимание следует обратить на слова при запуске. При обычном копировании зараженных файлов заражение компьютера произойти не может. В связи с этим все данные, принятые из Интернета, должны проходить обязательную проверку на безопасность, а если получены незатребованные данные из незнакомого источника, их следует уничтожать, не рассматривая. Обычный прием распространения «троянских» программ — приложение к электронному письму с «рекомендацией» извлечь и запустить якобы полезную программу.

**Загрузочные вирусы.** От программных вирусов загрузочные вирусы отличаются методом распространения. Они поражают не программные файлы, а определенные системные области магнитных носителей (гибких и жестких дисков). Кроме того, на включенном компьютере они могут временно располагаться в оперативной памяти.

Обычно заражение происходит при попытке загрузки компьютера с магнитного носителя, системная область которого содержит загрузочный вирус. Так, например, при попытке загрузить компьютер с гибкого диска происходит сначала проникновение вируса в оперативную память, а затем в загрузочный сектор жестких дисков. Далее этот компьютер сам становится источником распространения загрузочного вируса.

**Макровирусы.** Эта особая разновидность вирусов поражает документы, выполненные в некоторых прикладных программах, имеющих средства для исполнения так называемых макрокоманд. В частности, к таким документам относятся документы текстового процессора Microsoft Word (они имеют расширение .DOCX). Заражение происходит при открытии файла документа в окне программы, если в ней не отключена

возможность исполнения макрокоманд. Как и для других типов вирусов, результат атаки может быть как относительно безобидным, так и разрушительным.

## **Методы защиты от компьютерных вирусов**

Существуют три рубежа защиты от компьютерных вирусов:

- предотвращение поступления вирусов;
- предотвращение вирусной атаки, если вирус все-таки поступил на компьютер;
- предотвращение разрушительных последствий, если атака все-таки произошла.

Существуют три метода реализации защиты:

- программные методы защиты;
- аппаратные методы защиты;
- организационные методы защиты.

В вопросе защиты ценных данных часто используют бытовой подход: «болезнь лучше предотвратить, чем лечить». К сожалению, именно он и вызывает наиболее разрушительные последствия. Создав бастионы на пути проникновения вирусов в компьютер, нельзя положиться на их прочность и остаться неготовым к действиям после разрушительной атаки. К тому же, вирусная атака — далеко не единственная и даже не самая распространенная причина утраты важных данных. Существуют программные сбои, которые могут вывести из строя операционную систему, а также аппаратные сбои, способные сделать жесткий диск неработоспособным. Всегда существует вероятность утраты компьютера вместе с ценными данными в результате кражи, пожара или иного стихийного бедствия.

Поэтому создавать систему безопасности следует в первую очередь «с конца» — с предотвращения разрушительных последствий любого воздействия, будь то вирусная атака, кража в помещении или физический выход жесткого диска из строя. Надежная и безопасная работа с данными достигается только тогда, когда любое неожиданное событие, в том числе и полное физическое уничтожение компьютера не приведет к катастрофическим последствиям.

### **Средства антивирусной защиты**

Основным средством защиты информации является резервное копирование наиболее ценных данных. При резервировании данных следует также иметь в виду и то, что надо отдельно сохранять все регистрационные и парольные данные для доступа к сетевым службам Интернета. Их не следует хранить на компьютере.

Создавая план мероприятий по резервному копированию информации, необходимо учитывать, что резервные копии должны храниться отдельно от компьютера. То есть, например, резервирование информации на отдельном жестком диске того же компьютера только создает иллюзию безопасности. Относительно новым и достаточно надежным приемом хранения ценных, но не конфиденциальных данных является их хранение в Web-папках на удаленных серверах в Интернете. Есть службы, бесплатно предоставляющие пространство (до нескольких Мбайт) для хранения данных пользователя.

Резервные копии конфиденциальных данных сохраняют на внешних носителях, которые хранят в сейфах, желательно в отдельных помещениях. При разработке организационного плана резервного копирования учитывают необходимость создания не менее двух резервных копий, сохраняемых в разных местах. Между копиями осуществляют ротацию. Например в течение недели ежедневно копируют данные на носители резервного комплекта А, а через неделю их заменяют комплектом Б, и т. д.

Вспомогательными средствами защиты информации являются антивирусные программы и средства аппаратной защиты. Так, например, простое отключение переключателя на материнской плате не позволит осуществить стирание перепрограммируемой микросхемы ПЗУ (флэш - BIOS), независимо от того, кто будет пытаться это сделать: компьютерный вирус, злоумышленник или неаккуратный пользователь.

Существует достаточно много программных средств антивирусной защиты. Они предоставляют следующие возможности.

1. Создание образа жесткого диска на внешних носителях (например, на гибких дисках). В случае выхода из строя данных в системных областях жесткого диска сохраненный «образ диска» может позволить восстановить если не все данные, то, по крайней мере, их большую часть. Это же средство может защитить от утраты данных при аппаратных сбоях и при неаккуратном форматировании жесткого диска.

2. Регулярное сканирование жестких дисков в поисках компьютерных вирусов. Сканирование обычно выполняется автоматически при каждом включении компьютера и при размещении внешнего диска в считывающем устройстве. При сканировании следует иметь в виду, что антивирусная программа ищет вирус путем сравнения кода программ с кодами известных ей вирусов, хранящимися в базе данных. Если база данных устарела, а вирус является новым, сканирующая программа его не обнаружит. Для надежной работы следует регулярно обновлять антивирусную программу. Желательная периодичность обновления — один раз в две недели; допустимая — один раз в три месяца.

3. Контроль, за изменением размеров и других атрибутов файлов. Поскольку некоторые компьютерные вирусы на этапе размножения изменяют параметры зараженных файлов, контролирующая программа может обнаружить их деятельность и предупредить пользователя.

4. Контроль, за обращениями к жесткому диску. Поскольку наиболее опасные операции, связанные с работой компьютерных вирусов, так или иначе, обращены на модификацию данных, записанных на жестком диске, антивирусные программы могут контролировать обращения к нему и предупреждать пользователя о подозрительной активности.

#### ***Популярные антивирусные средства***

Среди наиболее популярных к российским пользователям антивирусных пакетов можно назвать программы: *Antivirus Касперского*, *Norton Antivirus*, *Avast* и *Dr.Web*. По различным оценкам, в настоящее время продукты Лаборатории Касперского занимают большую часть российского рынка. Прочие производители, в первую очередь Symantec, «Диалог – Наука», Trend Micro и Panda, делят оставшуюся долю рынка.

### **9.1.2. Защита от несанкционированного вмешательства**

Организация надежной защиты КС невозможна с помощью только программно - аппаратных средств. Очень важным является административный контроль работы КС и защита от несанкционированного доступа к информации (объектам).

При организации доступа субъектов к объектам выполняются следующие действия:

- идентификация и аутентификация субъекта доступа;
- проверка прав доступа субъекта к объекту;
- ведение журнала учета действий субъекта.

При входе в КС, при получении доступа к программам и конфиденциальным данным субъект должен быть идентифицирован и аутентифицирован. Эти операции обычно выполняются вместе. Пользователь сначала сообщает сведения, позволяющие выделить его из множества субъектов (идентификация), а затем сообщает секретные сведения, подтверждающие, что он тот, за кого себя выдает.

Иногда проводится дополнительно авторизация субъекта, под которой понимается создание программной среды для его работы.

Для аутентификации субъекта чаще всего используются атрибутивные идентификаторы, которые делят на следующие категории:

- пароли;
- съемные носители информации;
- электронные жетоны;
- пластиковые карты;

- механические ключи.

В настоящее время аппаратура КС поддерживает ввод пароля до начала загрузки ОС. Такой пароль хранится в энергонезависимой памяти и обеспечивает предотвращение НСД до загрузки любых программных средств.

Другие способы (съёмные носители, карты и др.) предполагают наличие технических средств, хранящих идентификационную информацию.

### **9.1.3. Специфика обработки конфиденциальной информации**

При работе в Интернете следует иметь в виду, что насколько ресурсы Всемирной сети открыты каждому клиенту, настолько же и ресурсы его компьютерной системы могут быть при определенных условиях открыты всем, кто обладает необходимыми средствами.

Для частного пользователя этот факт не играет особой роли, но знать о нем необходимо, чтобы не допускать действий, нарушающих законодательства тех стран, на территории которых расположены серверы Интернета. К таким действиям относятся вольные или невольные попытки нарушить работоспособность компьютерных систем, попытки взлома защищенных систем, использование и распространение программ, нарушающих работоспособность компьютерных систем (в частности, компьютерных вирусов).

Работая во Всемирной сети, следует помнить о том, что абсолютно все действия фиксируются и протоколируются специальными программными средствами и информация, как о законных, так и о незаконных действиях обязательно где-то накапливается. Однако даже в обычной почтовой связи наряду с открытками существуют и почтовые конверты.

Потребность в аналогичных «конвертах» для защиты информации существует и в Интернете. Сегодня Интернет является не только средством общения и универсальной справочной системой — в нем циркулируют договорные и финансовые обязательства, необходимость защиты которых как от просмотра, так и от фальсификации, очевидна. Начиная с 1999, года Интернет становится мощным средством обеспечения розничного торгового оборота, а это требует защиты данных кредитных карт и других электронных платежных средств.

Принципы защиты информации в Интернете опираются на определение информации, сформулированное нами в первой главе. Информация — это продукт взаимодействия данных и адекватных им методов. Если в ходе коммуникационного процесса данные передаются через открытые системы (а Интернет относится именно к таковым), то исключить доступ к ним посторонних лиц невозможно даже теоретически. Соответственно, системы защиты сосредоточены на втором компоненте информации — на методах. Их принцип действия основан на том, чтобы исключить или, по крайней мере, затруднить возможность подбора адекватного метода для преобразования данных в информацию. Одним из приемов такой защиты является шифрование данных.

### ***Понятие о несимметричном шифровании информации***

Системам шифрования столько же лет, сколько письменному обмену информацией. Обычный подход состоит в том, что к документу применяется некий метод шифрования (назовем его ключом), после чего документ становится недоступен для чтения обычными средствами. Его можно прочитать только тот, кто знает ключ, — только он может применить адекватный метод чтения. Аналогично происходит шифрование и ответного сообщения. Если в процессе обмена информацией для шифрования и чтения используются одним и тем же ключом, то такой криптографический процесс является симметричным.

Основной недостаток симметричного процесса заключается в том, что, прежде чем начать обмен информацией, надо выполнить передачу ключа, а для этого опять-таки нужна защищенная связь, то есть проблема повторяется, хотя и на другом уровне. Если

рассмотреть оплату клиентом товара или услуги с помощью кредитной карты, то получается, что торговая фирма должна создать по одному ключу для каждого своего клиента и каким-то образом передать им эти ключи. Это крайне неудобно.

Поэтому в настоящее время в Интернете используют несимметричные криптографические системы, основанные на использовании не одного, а двух ключей. Происходит это следующим образом. Компания для работы с клиентами создает два ключа: один открытый (public — публичный) ключ, а другой закрытый (private — личный) ключ. На самом деле это как бы две «половинки» одного целого ключа, связанные друг с другом.

Ключи устроены так, что сообщение, зашифрованное одной половинкой, можно расшифровать только другой половинкой (не той, которой оно было закодировано). Создав пару ключей, торговая компания широко распространяет публичный ключ (открытую половинку) и надежно сохраняет закрытый ключ (свою половинку).

Как публичный, так и закрытый ключ представляют собой некую кодовую последовательность. Публичный ключ компании может быть опубликован на ее сервере, откуда каждый желающий может его получить. Если клиент хочет сделать фирме заказ, он возьмет ее публичный ключ и с его помощью закодирует свое сообщение о заказе и данные о своей кредитной карте. После кодирования это сообщение может прочесть только владелец закрытого ключа. Никто из участников цепочки, по которой пересылается информация, не в состоянии это сделать. Даже сам отправитель не может прочитать собственное сообщение, хотя ему хорошо известно содержание. Лишь получатель сможет прочесть сообщение, поскольку только у него есть закрытый ключ, дополняющий использованный публичный ключ.

Если фирме надо будет отправить клиенту квитанцию о том, что заказ принят к исполнению, она закодирует ее своим закрытым ключом. Клиент сможет прочитать квитанцию, воспользовавшись имеющимся у него публичным ключом данной фирмы. Он может быть уверен, что квитанцию ему отправила именно эта фирма, и никто иной, поскольку никто иной доступа к закрытому ключу фирмы не имеет.

### ***Принцип достаточности защиты***

Защита публичным ключом (впрочем, как и большинство других видов защиты, информации) не является абсолютно надежной. Дело в том, что поскольку каждый желающий может получить и использовать чей-то публичный ключ, то он может сколь угодно подробно изучить алгоритм работы механизма шифрования и попытаться установить метод расшифровки сообщения, то есть реконструировать закрытый ключ.

Это настолько справедливо, что алгоритмы кодирования публичным ключом даже нет смысла скрывать. Обычно к ним есть доступ, а часто они просто широко публикуются. Тонкость заключается в том, что знание алгоритма еще не означает возможности провести реконструкцию ключа в разумно приемлемые сроки. Так, например, правила игры в шахматы известны всем, и нетрудно создать алгоритм для перебора всех возможных шахматных партий, но он никому не нужен, поскольку даже самый быстрый современный суперкомпьютер будет работать над этой задачей дольше, чем существует жизнь на нашей планете.

Количество комбинаций, которое надо проверить при реконструкции закрытого ключа, не столь велико, как количество возможных шахматных партий, однако защиту информации принято считать достаточной, если затраты на ее преодоление превышают ожидаемую ценность самой информации. В этом состоит принцип достаточности защиты, которым руководствуются при использовании несимметричных средств шифрования данных. Он предполагает, что защита не абсолютна, и приемы ее снятия известны, но она все же достаточна для того, чтобы сделать это мероприятие нецелесообразным. При появлении иных средств, позволяющих-таки получить зашифрованную информацию в разумные сроки, изменяют принцип работы алгоритма, и проблема повторяется на более высоком уровне.

Разумеется, не всегда реконструкцию закрытого ключа производят методами простого перебора комбинаций. Для этого существуют специальные методы, основанные на исследовании особенностей взаимодействия открытого ключами с определенными структурами данных. Область науки, посвященная этим исследованиям, называется криптоанализом, а средняя продолжительность времени, необходимого для реконструкции закрытого ключа по его опубликованному открытому ключу, называется криптостойкостью алгоритма шифрования.

Для многих методов несимметричного шифрования криптостойкость, полученная в результате криптоанализа, существенно отличается от величин, заявляемых разработчиками алгоритмов на основании теоретических оценок. Поэтому во многих странах вопрос применения алгоритмов шифрования данных находится в поле законодательного регулирования. В частности, в России к использованию в государственных и коммерческих организациях разрешены только те программные средства шифрования данных, которые прошли государственную сертификацию в административных органах, в частности, в Федеральном агентстве правительственной связи и информации при Президенте Российской Федерации (ФАПСИ).

### ***Понятие об электронной подписи***

Мы рассмотрели, как клиент может переслать организации свои конфиденциальные данные (например, номер электронного счета). Точно так же он может общаться и с банком, отдавая ему распоряжения о перечислении своих средств на счета других лиц и организаций. Ему не надо ездить в банк и стоять в очереди — все можно сделать, не отходя от компьютера. Однако здесь возникает проблема: как банк узнает, что распоряжение поступило именно от данного лица, а не от злоумышленника, выдающего себя за него? Эта проблема решается с помощью, так называемой электронной подписи.

Принцип ее создания тот же, что и рассмотренный выше. Если нам надо создать себе электронную подпись, следует с помощью специальной программы (полученной от банка) создать те же два ключа: закрытый и публичный. Публичный ключ передается банку. Если теперь надо отправить поручение банку на операцию с расчетным счетом, оно кодируется публичным ключом банка, а своя подпись под ним кодируется собственным закрытым ключом. Банк поступает наоборот. Он читает поручение с помощью своего закрытого ключа, а подпись — с помощью публичного ключа поручителя. Если подпись читаема, банк может быть уверен, что поручение ему отправили именно мы, и никто другой.

### **Вопросы для самоконтроля**

1. Какие угрозы безопасности информации вы знаете?
2. Какие типы вирусов вы знаете, и какую угрозу они несут?
3. Какие меры следует предпринять для обеспечения безопасности информации?
4. Какие антивирусные программы вы знаете?
5. В чем заключаются особенности обработки конфиденциальной информации, с т.з. обеспечения ее безопасности?
6. Что такое электронная подпись?

## **Тематика лабораторных работ**

### **Лабораторная работа №1 Арифметические основы ЭВМ**

Цель работы: изучение основных принципов позиционных систем счисления.

### **Лабораторная работа №2 Основы машинной арифметики с двоичными числами**

Цель работы: изучение основных принципов кодирования числовых данных в ЭВМ.

### **Лабораторная работа №3 Формы представления чисел в ЭВМ**

Цель работы: изучение основных форм представления числовых данных в ЭВМ.

### **Лабораторная работа №4 Первичные настройки текстового процессора Microsoft Word 2007**

Цель работы: научиться настраивать параметры документа Word.

### **Лабораторная работа №5 Форматирование документа Microsoft Word**

Цель работы: научиться форматировать документ Word.

### **Лабораторная работа №6 Создание таблиц и списков.**

Цель работы: научиться создавать таблицы и различные списки в Microsoft Word.

### **Лабораторная работа №7 Вставка объектов в документ Word**

Цель работы: научиться вставлять различные объекты и символы в Microsoft Word.

### **Лабораторная работа №8 Создание сносок, ссылок и оглавлений.**

Цель работы: создавать сноски, ссылки и оглавления в Microsoft Word.

### **Лабораторная работа №9 Знакомство с Excel**

Цель работы: познакомиться с основными приемами работы с табличным процессором.

### **Лабораторная работа №10 Абсолютная, относительная адресация. Работа с формулами. Вставка диаграмм**

Цель работы: научиться работать с формулами и абсолютными адресами. Научиться создавать и изменять диаграммы.

### **Лабораторная работа №11 Использование логических функций**

Цель работы: научиться работать с логическими функциями ЕСЛИ, ИЛИ, И.

### **Лабораторная работа №12 Решение матриц средствами Excel**

Цель работы: научиться решать матрицы с помощью табличного процессора Excel.

### **Лабораторная работа №13 Проведение расчётов в EXCEL. Защита данных**

Цель работы: научиться проводить расчёты с помощью табличного процессора Excel, защищать листы и рабочие книги.

### **Лабораторная работа №14 Работа со списками в Excel**

Цель работы: научиться работать со списками, фильтрами, автофильтрами, функциями баз данных.

**ЛАБОРАТОРНАЯ №15 Создание однотабличной базы данных. Ввод, редактирование и вывод данных.**

Цель работы: научиться создавать однотабличные БД, редактировать и выводить данные.

**Лабораторная работа №16 Формы. Отчеты**

Цель работы: научиться создавать формы и отчёты.

**Лабораторная работа №17 Обработка информации в однотабличной базе данных.**

**Поиск, фильтрация, сортировка. Создание запросов**

Цель работы: научиться создавать запросы, обрабатывать информацию.

**Лабораторная работа №18 Создание многотабличной базы данных и технология работы с ней.**

Цель работы: освоить основные приёмы работы с многотабличной БД.