

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ  
ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

**Б1.В.16 Информационные технологии**

**Направление подготовки 35.03.06. Агроинженерия**

**Профиль образовательной программы «Электрооборудование и электротехнологии»**

**Форма обучения очная**

## СОДЕРЖАНИЕ

<b>1. Конспект лекций.....</b>	<b>3</b>
<b>1.1 Лекция № 1 Информация и ее свойства.....</b>	<b>3</b>
<b>1.2 Лекция № 2 Понятие информационная технология, ее свойства.....</b>	<b>21</b>
<b>1.3 Лекция № 3 Этапы развития информационных технологий. Классификация информационных технологий.....</b>	<b>25</b>
<b>1.4 Лекция № 4 Автоматизированное рабочее место (АРМ).....</b>	<b>33</b>
<b>1.5 Лекция № 5 Пользовательский интерфейс и его виды.....</b>	<b>39</b>
<b>1.6 Лекция № 6 Корпоративные информационные системы.....</b>	<b>55</b>
<b>1.7 Лекция № 7 Глобальные компьютерные сети.....</b>	<b>82</b>
<b>1.8 Лекция № 8 Понятие базы данных.....</b>	<b>90</b>
<b>1.9 Лекция № 9 Функциональные возможности СУБД.....</b>	<b>99</b>
<b>2. Методические материалы по выполнению лабораторных работ.....</b>	<b>105</b>
<b>2.1 Лабораторная работа № 1 Операционная система WINDOWS.....</b>	<b>105</b>
<b>2.2 Лабораторная работа № 2 Оболочка Norton Commander.....</b>	<b>111</b>
<b>2.3 Лабораторная работа № 3 Стандартные приложения WINDOWS.....</b>	<b>121</b>
<b>2.4 Лабораторная работа № 4 Общее управление MS WORD.....</b>	<b>125</b>
<b>2.5 Лабораторная работа № 5 Структура электронной таблицы.....</b>	<b>130</b>
<b>2.6 Лабораторная работа № 6 Создание и заполнение таблицы.....</b>	<b>133</b>
<b>2.7 Лабораторная работа № 7 Обеспечение АРМ.....</b>	<b>146</b>
<b>2.8 Лабораторная работа № 8 Электронный офис.....</b>	<b>148</b>
<b>2.9 Лабораторная работа № 9 Локальные вычислительные сети.....</b>	<b>150</b>
<b>2.10 Лабораторная работа № 10 Построение.....</b>	<b>155</b>
<b>2.11 Лабораторная работа № 11 Корпоративные информационные системы.....</b>	<b>167</b>
<b>2.12 Лабораторная работа № 12 Глобальные компьютерные сети.....</b>	<b>175</b>
<b>2.13 Лабораторная работа № 13 Обеспечение безопасности информации в информационных технологиях.....</b>	<b>182</b>
<b>2.14 Лабораторная работа № 14 Связанные таблицы.....</b>	<b>186</b>
<b>2.15 Лабораторная работа № 15 Создание базы данных.....</b>	<b>191</b>
<b>2.16 Лабораторная работа № 16 Формирование запросов, отчетов и формы для однотабличной базы данных.....</b>	<b>193</b>
<b>2.17 Лабораторная работа № 17 Обслуживание информационной системы средствами СУБД Access.....</b>	<b>200</b>

## **1. КОНСПЕКТ ЛЕКЦИЙ**

### **1.1 Лекция №1 (2 часа).**

**Тема:** «Информация и ее свойства»

#### **1.1.1 Вопросы лекции:**

1. Понятие информации и данных
2. Формы адекватности информации
3. Меры информации
4. Виды информации

#### **1.1.2 Краткое содержание вопросов:**

1. Понятие информации и данных

Всё, что нас окружает, и с чем мы ежедневно сталкиваемся, относится либо к физическим телам, либо к физическим полям. Известно, что все физические объекты находятся в состоянии непрерывного движения и изменения, которое сопровождается обменом энергией и её переходом из одной формы в другую. Все виды энергообмена сопровождаются появлением сигналов, то есть, все сигналы имеют в своей основе материальную энергетическую природу. При взаимодействии сигналов с физическими телами в физических телах возникают определённые изменения свойств – это явление называется регистрацией сигналов. Такие изменения можно наблюдать, измерять или фиксировать иными способами. При этом возникают и регистрируются новые сигналы, то есть, образуются данные.

**Итак, данные – это зарегистрированные сигналы.**

Обработка данных адекватными им методами создаёт новый продукт – информацию. Таким образом, информация возникает и существует в момент взаимодействия объективных данных и субъективных методов. Как и всякий объект, она обладает свойствами (объекты различимы по своим свойствам). Характерной особенностью информации, отличающей её от других объектов природы и общества, является то, что на свойства информации влияют как свойства данных, составляющих её содержательную часть, так и свойства методов, взаимодействующих с данными в ходе информационного процесса. По окончании процесса свойства информации переносятся на свойства новых данных, то есть свойства методов могут переходить на свойства данных.

**Итак, информация – это продукт взаимодействия данных и адекватных им методов.**

В ходе информационного процесса данные преобразуются из одного вида в другой с помощью методов. Обработка данных включает в себя множество различных операций.

В структуре возможных операций с данными можно выделить следующие операции:

- сбор данных – накопление информации с целью обеспечения достаточной полноты для принятия решения;
- форматизация данных – приведение данных, поступающих из разных источников, к одинаковой форме, чтобы сделать их сопоставимыми между собой, то есть повысить их уровень доступности;
- фильтрация данных – отсеивание тех данных, в которых нет необходимости для принятия решения; при этом должны возрастать достоверность и адекватность информации;
- сортировка данных – упорядочение данных по заданному признаку с целью удобства их использования; при этом должна повышаться доступность информации;
- архивация данных – организация хранения данных в удобной и легкодоступной форме; служит для снижения экономических затрат по хранению данных и повышает общую надёжность информационного процесса в целом;
- преобразование данных – перевод данных из одной формы в другую или из одной структуры в другую;

- защита данных – комплекс мер, направленных на предотвращение утраты, воспроизведения и модификации данных;
- транспортировка данных – приём и передача данных между удалёнными участниками информационного процесса.

Данные могут быть представлены следующими видами:

- целыми и действительными числами;
- текстом;
- мультимедийными (графическими объектами, звуковыми сигналами, цветными изображениями).

В зависимости от вида данных, они могут подразделяться на следующие типы:

- байтовый тип;
- целочисленные типы простой и двойной точности;
- типы действительных чисел простой и двойной точности;
- типы даты и времени;
- строковый тип;
- логический тип;
- тип объектов.
- Работа с большими наборами данных автоматизируется проще, когда данные упорядочены, то есть образуют заданную структуру. Существует три основных типа структур данных: **линейная, табличная и иерархическая**. При создании любой структуры данных необходимо обеспечить решение двух задач: как разделять элементы данных между собой и как разыскивать нужные элементы.

- Линейные структуры – это хорошо знакомые списки. Список – это простейшая структура данных, отличающаяся тем, что каждый элемент данных однозначно определяется своим уникальным номером в массиве (списке).

- Табличные структуры данных подразделяются на двумерные и многомерные.

- Двумерные табличные структуры данных (матрицы) – это упорядоченные структуры, в которых адрес элемента определяется номером столбца и номером строки, на пересечении которых находится ячейка, содержащая искомый элемент.

- Многомерные таблицы – это упорядоченные структуры данных, в которых адрес элемента определяется тремя и более измерениями. Для отыскания нужного элемента в таких таблицах необходимо знать параметры всех измерений (размерностей).

- Линейные и табличные структуры являются простыми. Ими легко пользоваться, поскольку адрес каждого элемента задаётся числом (для списка), двумя числами (для двумерной таблицы) или несколькими числами для многомерной таблицы. Они также легко упорядочиваются. Основным методом упорядочения таких данных является сортировка. Недостатком простых структур данных является трудность их обновления. При добавлении, например, произвольного элемента в упорядоченную структуру возникает необходимость изменения адресных данных у других элементов.

- Иерархические структуры – это структуры, объединяющие нерегулярные данные, которые трудно представить в виде списка или таблицы. В иерархической структуре адрес каждого элемента определяется маршрутом, ведущим от вершины структуры к данному элементу. Эти структуры по форме сложнее, чем линейные и табличные, но они не создают проблем с обновлением данных. Их легко развивать путём создания новых уровней. Недостатком иерархических структур является относительная трудоёмкость записи адреса элемента данных и сложность упорядочения. Поэтому для упорядочения в таких структурах применяется метод предварительной индексации. При этом каждому элементу данных присваивается свой уникальный индекс, который используется при поиске, сортировке и тому подобное. В качестве примера иерархической структуры может служить система почтовых адресов.

## 2. Формы адекватности информации

Для потребителя информации очень важной характеристикой является ее *адекватность*.

**Адекватность информации** - это соответствие создаваемого с помощью полученной информации образа реальному объекту, процессу, явлению и т.п.

В реальной жизни вряд ли возможна ситуация, когда вы сможете ориентироваться на полную адекватность информации. Всегда присутствует некоторая степень неопределенности. От степени адекватности информации реальному состоянию объекта или процесса зависит правильность принятия решений потребителем.

### **Пример 2**

Вы успешно окончили школу и хотите продолжить образование по экономическому направлению. Поговорив с друзьями, вы узнаете, что подобную подготовку можно получить в разных вузах. В результате таких бесед вы получаете весьма разноречивые сведения, которые не позволяют вам принять решение в пользу того или иного варианта, т.е. полученная информация неадекватна реальному состоянию дел.

Для того чтобы получить более достоверные сведения, вы покупаете справочник для поступающих в вузы, из которого получаете исчерпывающую информацию. В этом случае можно говорить, что информация, полученная вами из справочника, адекватно отражает направления обучения в вузах и помогает вам определиться в окончательном выборе.

Адекватность информации может выражаться в трех формах: *семантической, синтаксической, прагматической*.

### **Синтаксическая адекватность**

*Синтаксическая адекватность* отображает формально-структурные характеристики информации и не затрагивает 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, т.е. синтаксической стороны информации.

### **Семантическая (смысловая) адекватность**

*Семантическая адекватность* определяет степень соответствия образа объекта и самого объекта. Семантический аспект имеет в виду учет смыслового содержания информации. На этом уровне анализируются те сведения, которые отражает информация, рассматриваются смысловые связи. В информатике устанавливаются смысловые связи между кодами представления информации. Эта форма служит для формирования понятий и представлений, выявления смысла, содержания информации и ее обобщения.

### **Прагматическая (потребительская) адекватность**

*Прагматическая адекватность* отражает отношение информации и ее потребителя, соответствие информации цели управления, которое на ее основе реализуется. Проявляются прагматические свойства информации только при наличии единства информации (объекта), пользователя и цели управления. Прагматический аспект рассмотрения связан с ценностью, полезностью использования информации для выработки потребителем решения для достижения своей цели. С этой точки зрения анализируются потребительские свойства информации. Эта форма адекватности непосредственно связана с практическим использованием информации, с соответствием ее целевой функции деятельности системы.

### **3. Меры информации**

Для измерения информации вводятся два параметра:

- количество информации  $I$  ;
- объем данных  $V_d$  .

Эти параметры имеют разные выражения и интерпретацию в зависимости от рассматриваемой формы адекватности. Каждой форме адекватности соответствует своя мера количества информации и объема данных (рис. 1).

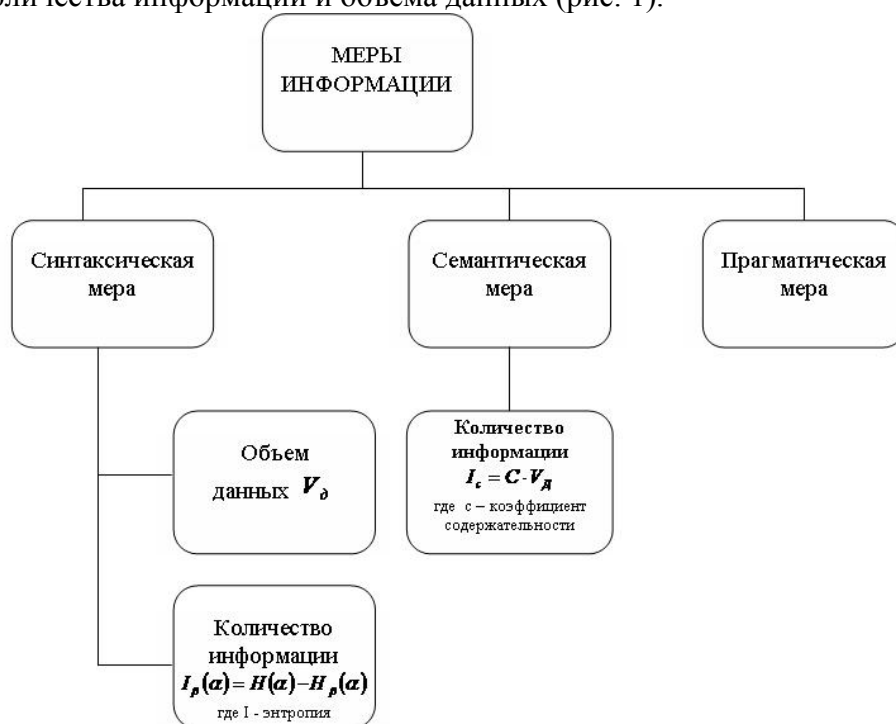


Рис. 1. Меры информации

### Синтаксические меры информации

Синтаксические меры количества информации имеют дело с обезличенной информацией, не выражающей смыслового отношения к объекту.

Объем данных  $V_d$  в сообщении измеряется количеством символов (разрядов) в этом сообщении. В различных системах счисления один разряд имеет различный вес, и соответственно меняется единица измерения данных:

- в двоичной системе счисления единица измерения - бит (*binary digit* - двоичный разряд). Наряду с этой единицей измерения широко используется укрупненная единица измерения “байт”, равная 8 бит.
- в десятичной системе счисления единица измерения - дит (десятичный разряд).

### Пример 3

Сообщение в двоичной системе в виде восьмиразрядного двоичного кода 10111011 имеет объем данных  $V_d = 8$  бит. Сообщение в десятичной системе в виде шестиразрядного числа 275903 имеет объем данных  $V_d = 6$  дит.

Определение количества информации  $I$  на синтаксическом уровне невозможно без рассмотрения понятия неопределенности состояния системы (энтропии системы). Действительно, получение информации о какой-либо системе всегда связано с изменением степени неосведомленности получателя о состоянии этой системы.

Рассмотрим это понятие.

Пусть до получения информации потребитель имеет некоторые предварительные (априорные) сведения о системе  $a$ . Мерой его неосведомленности о системе является функция  $H(a)$ , которая в тоже время служит и мерой неопределенности состояния системы. Эта мера получила название *энтропия*. Если потребитель имеет полную информацию о системе, то энтропия равна 0. Если потребитель имеет полную неопределенность о какой-то системе, то энтропия является положительным числом. По мере получения новой информации энтропия уменьшается.

После получения некоторого сообщения  $b$  получатель приобрел некоторую дополнительную информацию  $I_b(\alpha)$ , уменьшившую его априорную неосведомленность так, что апостериорная (после получения сообщения  $b$ ) неопределенность состояния системы стала  $H_b(\alpha)$ .

Тогда количество информации  $I_b(\alpha)$  о системе, полученное в сообщении  $b$ , определится как  $I_b(\alpha) = H(\alpha) - H_b(\alpha)$ , т. е. количество информации измеряется изменением (уменьшением) неопределенности состояния системы.

Если конечная неопределенность  $H_b(\alpha)$  обратится в нуль, то первоначальное неполное знание заменится полным знанием и количество информации  $I_b(\alpha) = H(\alpha)$ . Иными словами, энтропия системы  $H(a)$  может рассматриваться как мера недостающей информации.

Энтропия системы  $H(a)$ , имеющая  $N$  возможных состояний, согласно формуле Шеннона, равна

$$H(\alpha) = - \sum_{i=1}^N P_i \cdot \log_2 P_i \quad (1)$$

где  $P_i$  - вероятность того, что система находится в  $i$ -м состоянии.

Для случая, когда все состояния системы равновероятны, т.е. их вероятности равны

$$P_i = \frac{1}{N},$$

ее энтропия определяется соотношением

$$H(\alpha) = - \sum_{i=1}^N \frac{1}{N} \cdot \log_2 \frac{1}{N} = \log_2 N \quad (2)$$

Энтропия системы в двоичной системе счисления измеряется в битах. Исходя из формулы (2) можно сказать, что в системе в равновероятными состояниями 1 бит равен количеству информации, которая уменьшает неопределенность знаний в два раза.

#### Пример 4

Система, которая описывает процесс бросания монеты, имеет два равновероятных состояния. Если вам нужно угадать, какая сторона выпала сверху, то вы сначала имеете полную неопределенность о состоянии системы. Что бы получить информацию о состоянии системы, вы задаете вопрос: "Это орел?". Этим вопросом вы пытаетесь отбросить половину неизвестных состояний, т.е. уменьшить неопределенность в 2 раза. Какой бы ответ ни последовал "Да" или "Нет", вы получите полную ясность о состоянии системы. Таким образом, ответ на вопрос содержит 1 бит информации. Поскольку после 1-го вопроса наступила полная ясность, то энтропия системы равна 1. Этот же ответ дает формула (2), т.к.  $\log_2 2 = 1$ .

#### Пример 5.

Игра "Отгадай число". Вам надо угадать задуманное число от 1 до 100. В начале отгадывания вы имеете полную неопределенность о состоянии системы. При отгадывании надо задавать вопросы не хаотично, а так, чтобы ответ уменьшал неопределенность знаний в 2 раза, получая таким образом примерно 1 бит информации после каждого вопроса. Например, сначала надо задать вопрос: "Число больше 50?". "Правильный" подход к отгадыванию дает возможность угадать число за 6-7 вопросов. Если применить формулу (2), то получится, что энтропия системы равна  $\log_2 100 = 6,64$ .

#### Пример 6.

Алфавит племени "тумбо-юмбо" содержит 32 различных символа. Какова энтропия системы? Другими словами надо определить, какое количество информации несет в себе

каждый символ. Если считать, что каждый символ встречается в словах с равной вероятностью, то энтропия  $\log_2 32 = 5$ .

Наиболее часто используются двоичные и десятичные логарифмы. Единицами измерения в этих случаях будут соответственно бит и дит.

*Коэффициент (степень) информативности* (лаконичность) сообщения определяется отношением количества информации к объему данных, т.е.

$$Y = \frac{I}{V_D},$$

причем  $0 < Y < 1$ .

Чем больше коэффициент информативности  $Y$ , тем меньше объем работы по преобразованию информации (данных) в системе. Поэтому стремятся к повышению информативности, для чего разрабатываются специальные методы оптимального кодирования информации.

### **Семантическая мера информации**

Для измерения смыслового содержания информации, т.е. ее количества на семантическом уровне наибольшее признание получила тезаурусная мера, предложенная Ю.И.Шнейдером. Он связывает семантические свойства информации прежде всего со способностью пользователя принимать поступившее сообщение. Для этого используется понятие "*тезаурус пользователя*".

**Тезаурус** - это совокупность сведений, которыми располагает пользователь или система.

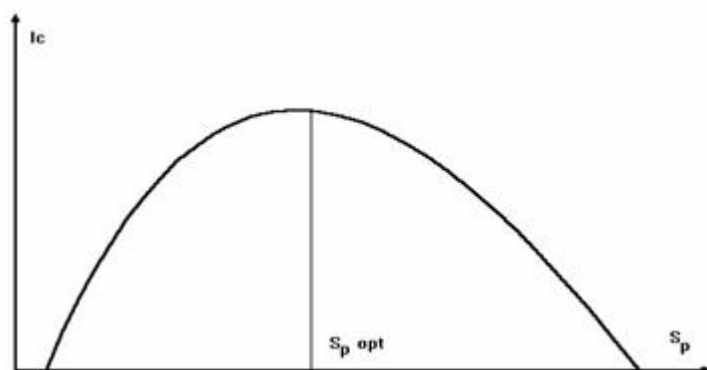
В зависимости от соотношений между смысловым содержанием информации  $S$  и тезаурусом пользователя  $Sp$  изменяется количество семантической информации  $I_c$ , воспринимаемой пользователем и включаемой им в дальнейшем в свой тезаурус. Характер такой зависимости показан на рис. 2. Рассмотрим два предельных случая, когда количество семантической информации  $I_c$  равно 0:

- при  $Sp \approx 0$  пользователь не воспринимает, не понимает поступающую информацию;
- при  $Sp \rightarrow \infty$  пользователь все знает, и поступающая информация ему не нужна.

Максимальное количество семантической информации  $I_c$  потребитель приобретает при согласовании ее смыслового содержания  $S$  со своим тезаурусом  $S_p = S_{порт}$ ; когда поступающая информация понятна пользователю и несет ему ранее не известные (отсутствующие в его тезаурусе) сведения.

Следовательно, количество семантической информации в сообщении, количество новых знаний, получаемых пользователем, является величиной относительной. Одно и то же сообщение может иметь смысловое содержание для компетентного пользователя и быть бессмысленным (семантический шум) для пользователя некомпетентного.





о.

Рис. 2. Зависимость количества семантической информации, воспринимаемой потребителем, от его тезауруса  $I_c = f(S_p)$

При оценке семантического (содержательного) аспекта информации надо стремиться к согласованию величин  $S$  и  $Sp$ .

Относительной мерой количества семантической информации может служить коэффициент содержательности  $C$ , который определяется как отношение количества семантической информации к ее объему

$$C = \frac{I_c}{V_d}$$

### Прагматическая мера информации

Прагматическая мера информации служит для определения ее *полезности* (ценности) для достижения пользователем поставленной цели. Эта мера также величина относительная, обусловленная особенностями использования этой информации в той или иной системе. Ценность информации целесообразно измерять в тех же самых единицах (или близких к ним), в которых измеряется целевая функция.

### Пример 7

В экономической системе прагматические свойства (ценность) информации можно определить приростом экономического эффекта функционирования, достигнутым благодаря использованию этой информации для управления системой:

$$I_{n\beta}(\gamma) = \Pi(\gamma / \beta) - \Pi(\gamma)$$

где  $I_{n\beta}(\gamma)$  - ценность информационного сообщения  $\beta$  для системы управления  $\gamma$ ;

$\Pi(\gamma)$  - априорный ожидаемый экономический эффект функционирования системы управления  $\gamma$ ;

$\Pi(\gamma / \beta)$  - ожидаемый эффект функционирования системы  $\gamma$  при условии, что для управления будет использована информация, содержащаяся в сообщении  $\beta$ .

Для сопоставления введенные меры информации представим в табл. 1.

Таблица 1. Единицы измерения информации и примеры

Меры информации	Единицы измерения	Примеры (для компьютерной области)
Синтаксическая: а) Шенноновский подход б) компьютерный подход	а) степень уменьшения неопределенности б) единицы представления информации	а) вероятность события б) бит, байт, Кбайт и т.д.
Семантическая	а) тезаурус	а) пакет прикладных программ,

кая	б) экономические показатели	персональный компьютер, компьютерные сети и т.д. б) рентабельность, производительность, коэффициент амортизации и т.д.
Прагматическая	Ценность использования	Емкость памяти, производительность компьютера, скорость передачи данных и т.д. Денежное выражение Время обработки информации и принятия решений

#### 4. Виды информации

Виды информации различают, как правило, по следующим основным характеристикам:

- ➔ степень упорядоченности;
- ➔ форма закрепления;
- ➔ доступность;
- ➔ содержание (сфера применения, ценность информации от времени, предназначение);
- ➔ затраты интеллектуального труда человека на создание информации;

По степени упорядоченности различают информацию и документированную информацию.

По форме закрепления информация подразделяется на:

- ➔ представленную в письменном виде (рукопись, машинопись, нотная запись и т.д.);
- ➔ в устной форме (публичное исполнение и т.д.);
- ➔ в виде звуковой или видеозаписи механической, магнитной, цифровой, оптической и т.д.);
- ➔ в виде изображения (рисунок, эскиз, картина, план, чертеж, кино-, теле-, видеоизображения, компьютерная графика и анимация, фотография и т.д.);
- ➔ в объемно-пространственной форме (скульптура, модель, макет, сооружение и т.д.).

По доступности информация разделяется на открытую (общедоступную) и ограниченного доступа.

По содержанию информация может быть классифицирована по сфере применения, ценности информации от времени, предназначению.

По сфере применения информация может быть разделена на:

- ➔ экономическую;
- ➔ социально значимую;
- ➔ служебную;
- ➔ частную;
- ➔ развлекательную и т.п.

По ценности от времени информация может быть разделена на:

- ➔ оперативную;
- ➔ неоперативную;
- ➔ памятники культуры.

Ценность оперативной информации существенно зависит от времени, прошедшего с момента ее возникновения. Ценность неоперативной информации в меньшей степени зависит от времени. Произведения художественной литературы или искусства могут сохранять свою ценность в течение нескольких лет. Если ценность этих произведений не уменьшается в течение десятков лет, они могут быть отнесены к памятникам культуры.

По предназначению информация в социальных системах подразделяется на:

- массовую;
- групповую;
- индивидуальную.

Массовая информация представляет собой сведения, предназначенные для неограниченного круга лиц. Как правило, форма представления массовой информации – это предназначенные для неограниченного круга лиц печатные, аудио-, аудиовизуальные и иные сообщения и материалы.

Групповая информация ориентирована на использование вполне определенным кругом лиц, имеющих отношение к содержащимся в ней сведениям.

Индивидуальная информация ориентирована на конкретного человека.

В коммуникативных процессах информация как таковая может быть двух типов: побудительная и констатирующая.

Побудительная информация проявляется в форме приказа, совета или просьбы. Она призвана стимулировать какое-то действие. Стимуляцию, в свою очередь, подразделяют на активизацию (побуждение к действию в заданном направлении), интердикцию (запрет нежелательных видов деятельности) и дестабилизацию (рассогласование или нарушение некоторых автономных форм поведения или деятельности).

Констатирующая информация проявляется в форме сообщения и не предполагает непосредственного изменения поведения.

По интеллектуальным затратам человека на создание информации она условно может быть разделена на:

- высокоинтеллектуальную;
- распорядительную;
- бытовую.

С точки зрения способа фиксации информации существуют различные виды документированной информации – т.е. зафиксированной на каком-либо материальном носителе информация с реквизитами, позволяющими ее идентифицировать.

Под носителем информации в общем случае понимается физическое лицо, или материальный объект, в том числе физическое поле, в котором информация находит свое отображение в виде символов, образов, сигналов, технических решений и процессов.

Документирование информации является обязательным условием включения информации в информационные ресурсы. По действующему законодательству Российской Федерации документированная информация с ограниченным доступом по условиям ее правового режима подразделяется на информацию, отнесенную к государственной тайне, и конфиденциальную.

Документированная информация при ее изучении и анализе называется также документальными источниками. Классификация документальных источников возможна по разным критериям: по способу фиксации информации, по целевому назначению, по степени персонификации, по статусу, по источнику информации и т.д.

Информация может иметь ряд ограничений по доступу к ней, различные условия ее правового режима. В этих случаях необходимо обеспечивать ее защиту от неправомерного доступа, искажений, модификации и прочих противоправных действий. Данный вид информации рассматривается как защищаемая – т.е. информация, являющаяся предметом собственности и подлежащая защите в соответствии с требованиями правовых документов или требованиями, устанавливаемыми собственником информации.

Одним из главных видов информации, защита которой гарантируется на уровне национального законодательства, является конфиденциальная.

Под конфиденциальной информацией понимается документированная информация, ознакомление с которой ограничивается ее собственником или в соответствии с законодательством. Ограничения (установление режима) использования

информации определяются законом или собственником информации, которые объявляют о степени (уровне) ее конфиденциальности.

В соответствии с законодательством Российской Федерации к конфиденциальной информации относятся, в частности, такие виды информации, как:

- государственная тайна – информация в области военной, внешнеполитической, экономической, разведывательной, контрразведывательной и оперативно-розыскной деятельности, распространение которой может нанести ущерб безопасности государства;

- служебная и коммерческая тайна – информация, имеющая действительную или потенциальную коммерческую ценность в силу неизвестности ее третьим лицам, если к ней нет законного доступа на законных (санкционированных) основаниях и обладатель такой информации принимает меры к охране ее конфиденциальности;

- банковская, личная и семейная тайна;

- тайна переписки, телефонных переговоров, почтовых, телеграфных или иных сообщений;

- тайна усыновления;

- информация, являющаяся объектом авторских и смежных прав;

- информация, непосредственно затрагивающая права и свободы гражданина, или персональные данные – сведения о фактах, событиях и обстоятельствах жизни гражданина, позволяющие идентифицировать его личность, как правило, включаемые в состав государственных информационных ресурсов, органов местного самоуправления, а также получаемые и собираемые негосударственными организациями.

Законодательством также обеспечивается защита такой информации, как частная – это информация, которая затрагивает ограниченный круг лиц и касается их частной жизни: сведения, раскрывающие реализацию гражданином своих конституционных прав на свободу мысли, совести, собраний, информационной деятельности, о мировоззрении, нравственных ценностях, отношении к религии и т.д.

Действующим законодательством Российской Федерации запрещено относить к информации с ограниченным доступом:

- законодательные и другие нормативные акты, устанавливающие правовой статус органов государственной власти, органов местного самоуправления, организаций, общественных объединений, а также права, свободы и обязанности граждан, порядок их реализации;

- документы, содержащие информацию о чрезвычайных ситуациях, экологическую, метеорологическую, демографическую, санитарно-эпидемиологическую и другую информацию, необходимую для обеспечения безопасного функционирования населенных пунктов, производственных объектов, безопасности граждан и населения в целом;

- документы, содержащие информацию о деятельности органов государственной власти и органов местного самоуправления, об использовании бюджетных средств и других государственных и местных ресурсов, о состоянии экономики и потребностях населения, за исключением сведений, отнесенных к государственной тайне;

- документы, накапливаемые в открытых фондах библиотек и архивов, информационных системах органов государственной власти, органов местного самоуправления, общественных объединений, организаций, представляющие общественный интерес или необходимые для реализации прав, свобод и обязанностей граждан.

Информация, не имеющая ограничений по доступу к ней всех заинтересованных лиц, – это открытая информация.

Если информация подверглась модификации, осуществленной в деструктивных целях, то такая информация определяется как искаженная. В социально-политическом аспекте искаженная информация – это информация, претерпевшая под действием определенных способов трансформации и искажения информационных потоков

качественные изменения, в результате которых она начинает оказывать негативное воздействие на функционирование политической системы общества. Искорженная информация, таким образом, является деструктивной разновидностью социально значимой информации, искажающей адекватность политической реальности. Трансформация и изменение информации или процессов ее функционирования используется некоторыми субъектами для оказания воздействия на психику людей и их поведение, для психологических манипуляций и оказания манипулятивного воздействия на личность и общество.

Выделяют следующие основные разновидности искаженной информации в сравнении с ее антиподами – "противоположными полюсами" информации, необходимой для нормального функционирования конкретных структурных элементов политической системы общества.

1. "Объективная – фальсифицированная информация". В политической борьбе объективная информация часто сознательно или непроизвольно "фальсифицируется" – подделывается, искажается, подлинное подменяется ложным, что порождает бредовые политические идеи".

2. "Системная – дезориентирующая информация". Под видом системной политической информации может функционировать ее подобие – "информация дезориентирующая", неправильно определяющая положение общества в историческом процессе: экономическом состоянии, отношениях с иными народами и государствами и т.д. Она может порождать сверхценные политические идеи, объективно не заслуживающие такого внимания, которое субъективно они вызывают.

3. "Организованная – деморализующая информация". Организованность информации способствует упорядочиванию нравственных принципов, экономических требований, поведения в нечто целое и гармоничное. Может происходить подмена организованной информации "деморализующей", которая трансформирует нравственные ценности и провоцирует "скачок политических идей", вытекающий из нарушения последовательности умозаключений, утверждений, хаотичности и незавершенности мыслей, непоследовательности и поведения.

4. "Достаточная – энтропийная политическая информация". Вместо краткой, убедительной и доказательной информации в средствах массовой коммуникации появляются всевозможные точки зрения от самых примитивных до самых фантастических и невероятных. Множество газет, политических партий, авторитетов и оракулов на тысячи ладов толкуют элементарные вещи, делая известное – неизвестным, понятное – непонятым, уродливое – красивым, нравственное – безнравственным, ясное – неясным. Причем доминируют наиболее примитивные, мистические и просто бредовые объяснения.

5. "Социализирующая политическая информация" – "политическая дезинформация". В "социализирующей политической информации" отражаются результаты трансформации исходных сведений о социально-политической ситуации и направлениях ее развития в форму ясных, понятных и убедительных для большинства населения новых алгоритмов и стереотипов социального поведения, адекватных социальной действительности. В то же время "социализирующая политическая информация" может подменяться "политической дезинформацией" – ложными сообщениями, которые вводят общественное мнение в заблуждение под видом истинных. Начинается с уничтожения книг и издания новых, с принципиально иными фактами и их трактовкой. В результате действия таких информационных процессов развивается так называемая массовая "бессвязность политического мышления", при котором возможно правильное восприятие частных фактов, но с утратой способности к логическим выводам и синтезированию частных фактов в целом. Известные представления, понятия, мысли утрачивают их истинную историческую связь, разрываются. Информационная ткань распадается на бессмысленные фрагменты, из которых составляется новая деформированная картина мира.

6. "Конкретная политическая информация" – "дезорганизирующая политическая информация", т.е. расстраивающая систему административного управления, разрушающая общественный порядок и приводящая к развалу хозяйственной и финансовой деятельности. Осуществляется методами "политического резонерства", основными признаками которых являются абсолютно бессодержательные, бедные мыслью программные политические выступления, облеченные во внешне правильную и привлекательную форму.

7. "Практичная информация" – "развращающая политическая информация" побуждает к запрещенным действиям: цель становится всем – неблагое средства используются для достижения результата, право истины замещается правом силы и т.д. Распространение развращающей политической информации порождает феномен "разорванности политического мышления", что с точки зрения психологии является признаком патологии мышления. В этом случае понятия и представления сочетаются друг с другом на основе случайных или формальных признаков.

8. "Необходимая информация" – "дизинтегрирующая политическая информация", под воздействием которой разъединяется целостное общество на соперничающие, противоборствующие части. Люди вдруг обнаруживают под влиянием такой информации неразрешимые противоречия в отношениях со своими недавними соратниками, коллегами, даже родственниками. Крайним следствием результатов воздействия этого вида информации может выступать гражданская война ("брат против брата", "сын против отца" и т.д.).

С появлением электронно-вычислительных машин возник такой вид информации, как информация компьютерная – идентифицируемый, имеющий собственника элемент информационной системы.

Одно из первых определений компьютерной информации принадлежит Н. Винеру. Компьютерная информация – это информация, которая передается, обрабатывается и хранится с использованием электронно-вычислительной техники. Она может быть перенесена в пространстве, сохранена во времени, передана другому субъекту или техническому устройству, подвергнута иным операциям.

Существуют и другие определения для данного термина. Ниже перечислены некоторые из них.

Компьютерная информация это:

- сведения (данные) о лицах, предметах, фактах, событиях, явлениях и процессах, которые представлены в специальном, понятном машине, виде, пригодном для их автоматизированной обработки, хранения и передачи, находятся на машинном носителе, в памяти ЭВМ или циркулируют по каналам связи, объединяющим ЭВМ в единую систему или сеть, и имеют собственника;

- сведения, знания или набор команд для использования в ЭВМ или управления ею, находящиеся в ЭВМ или на машинных носителях, идентифицируемый элемент информационной системы, имеющий собственника, установившего правила ее использования;

- сведения (данные) о лицах, предметах, фактах, событиях, явлениях и процессах, подлежащие вводу в ЭВМ, хранимые в ее памяти, обрабатываемые на ЭВМ и выдаваемые пользователям;

- информация, представленная в специальном (машинном) виде, предназначенном и пригодном для ее автоматизированной обработки, хранения и передачи, находящаяся на материальном носителе и имеющая собственника, установившего порядок ее создания (генерации), обработки, передачи и уничтожения;

- информация, зафиксированная на машинном носителе или передаваемая по телекоммуникационным каналам в форме, доступной восприятию ЭВМ;

- документированная информация, хранящаяся в памяти ЭВМ или управляющая ею;

- ✦ информация на машинном носителе, в ЭВМ, системе или сети ЭВМ;
- ✦ информация, которая передается, обрабатывается и хранится с использованием электронно-вычислительной техники.

По мнению ряда специалистов, ограничивать место содержания компьютерной информации только ЭВМ или машинными носителями не оправданно, так как в настоящее время находят широкое распространение автоматизированные информационные системы, построенные на сети ЭВМ, объединённых по радиоканалу.

Компьютерная информация в общем случае может содержаться (находиться) как на машинном носителе, так и в памяти ЭВМ, циркулировать по каналам связи, объединяющим ЭВМ в единую систему или сеть.

В ряде случаев вместо термина "компьютерная информация" используется термин "машинная информация", под которой, как правило, понимают информацию, циркулирующую в вычислительной среде, зафиксированную на физическом носителе в форме, доступной восприятию ЭВМ, или передающуюся по телекоммуникационным каналам; сформированную в вычислительной среде и пересылаемую посредством электромагнитных сигналов из одной ЭВМ в другую, из ЭВМ на периферийное устройство, либо на управляющий датчик оборудования.

## **1. 2 Лекция №2 (2 часа).**

**Тема:** «Понятие информационная технология, ее свойства»

### **1.2.1 Вопросы лекции:**

1. Понятие информационная технология.
2. Свойства информации.

### **1.2.2 Краткое содержание вопросов:**

1. Понятие информационная технология.

Информационные технологии (ИТ, от англ. information technology, IT) — широкий класс дисциплин и областей деятельности, относящихся к технологиям управления и обработки данных, в том числе, с применением вычислительной техники.

В последнее время под информационными технологиями чаще всего понимают компьютерные технологии. В частности, ИТ имеют дело с использованием компьютеров и программного обеспечения для хранения, преобразования, защиты, обработки, передачи и получения информации. Спецов по компьютерной технике и программированию часто называют ИТ-специалистами.

Информационные технологии во всём мире всё чаще становятся средствами, используемые человечеством для расширения и развития своих способностей.

Следует заметить: существительное “технология”, выражающее существо данного учебного предмета, ныне столь часто употребляется, что оно практически превратилось в прилагательное, в большей степени связанное с различными предметными областями и средствами. При этом технология – это всегда метод или способ выполнения определённых операций и процессов, связанных с изменением качества или первоначального состояния материала, объекта и т.п. Так, например, технология материального производства подразумевает процесс, заключающийся в изготовлении, обработке, изменении состояния, свойств и формы сырья или материала.

Особенностью информационных технологий (ИТ) является то, что они связаны с одним или несколькими информационными процессами: создания, получения (сбора), хранения, обработки, передачи и распространения информации.

Кроме того, они не могут рассматриваться изолированно, вне материальной сферы. Информация является неотъемлемой и, часто, определяющей компонентой практически всех материальных процессов, инициируемых человеком или в которых он участвует.

Сбор данных (информации) представляет процесс регистрации, фиксации, записи данных о событиях, объектах (реальных и абстрактных), связях, признаках и соответствующих действиях. Более сложным является понятие “сбор информации” – это процесс отбора из различных источников данных, необходимых для решения целевой задачи основной деятельности, их идентификации и представления в форме, необходимой для ввода в ЭВМ, последующего поиска и функциональной обработки.

Обработка данных включает взаимосвязанные операции поиска, выборки, сортировки, слияния, проведения расчётов и т. д. Она представляет процесс управления данными, по возможности без учёта смысла, заложенного в эти данные.

Обработка информации представляет процессы переработки данных, реализующие содержательное (функциональное) преобразование информации какого-либо определённого типа (текстовой, цифровой, графической и др.). Причём, зачастую тип данных (как форма существования информации) определяется характером содержания, т.е., обработка информации – это преобразование данных с учётом их содержания.

Хотя информационные технологии в значительной степени ориентированы на решение промышленных задач, их следует рассматривать не только как инструмент, умножающий возможности человека, но в концептуальном плане, как методологическую платформу, обладающую универсальными парадигмами, моделями, методами, языками для представления, формализации, моделирования, систематизации, обработки прикладных знаний.



Другая важная особенность ИТ предопределена естественным, особенно для больших систем, требованием надёжности и устойчивости их функционирования и развития, а также возможности интеллектуального (человеческого) контроля в условиях большой сложности. Это означает, что сфера информационных технологий распространяется практически на все этапы жизненного цикла различных продуктов и услуг.

Третья особенность – это непосредственное или опосредованное участие человека в технологических процессах. Любой автоматизированный и даже автоматический процесс связан с необходимостью представления (или получения) информации в форме, удобной для человека.

Современные индивиды должны знать возможности применения ИТ в повседневной жизни, а значит и основы информатики с тем, чтобы успешно использовать свои знания при решении личностных и производственных задач.

#### **Структура отрасли:**

1. Информатика
2. Программирование
3. Интернет и Всемирная паутина
4. Веб-разработка
5. Управление данными
6. Обработка данных
7. RFID
8. Добыча данных
9. Хранение данных
10. Базы данных
11. Информационная архитектура
12. Информационная безопасность
13. Криптография
14. Системная интеграция
15. Искусственный интеллект

#### **2. Свойства информации.**

Как и всякий объект, информация обладает свойствами. Характерной отличительной особенностью информации от других объектов природы и общества, является дуализм: на свойства информации влияют как свойства исходных данных, составляющих ее содержательную часть, так и свойства методов, фиксирующих эту информацию.

С точки зрения информатики наиболее важными представляются следующие общие качественные свойства: объективность, достоверность, полнота, точность, актуальность, полезность, ценность, своевременность, понятность, доступность, краткость и пр.

**Объективность информации.** Объективный – существующий вне и независимо от человеческого сознания. Информация – это отражение внешнего объективного мира.

Информация объективна, если она не зависит от методов ее фиксации, чьего-либо мнения, суждения.

Пример. Сообщение «На улице тепло» несет субъективную информацию, а сообщение «На улице 22°C» – объективную, но с точностью, зависящей от погрешности средства измерения.

Объективную информацию можно получить с помощью исправных датчиков, измерительных приборов. Отражаясь в сознании конкретного человека, информация перестает быть объективной, так как, преобразовывается (в большей или меньшей степени) в зависимости от мнения, суждения, опыта, знаний конкретного субъекта.

Достоверность информации. Информация достоверна, если она отражает истинное положение дел. Объективная информация всегда достоверна, но достоверная информация может быть как объективной, так и субъективной. Достоверная информация помогает принять нам правильное решение. Недостоверной информация может быть по следующим причинам:

- преднамеренное искажение (дезинформация) или непреднамеренное искажение субъективного свойства;
- искажение в результате воздействия помех («испорченный телефон») и недостаточно точных средств ее фиксации.

Полнота информации. Информацию можно назвать полной, если ее достаточно для понимания и принятия решений. Неполная информация может привести к ошибочному выводу или решению.

Точность информации определяется степенью ее близости к реальному состоянию объекта, процесса, явления и т. п.

Актуальность информации – важность для настоящего времени, злободневность, насущность. Только вовремя полученная информация может быть полезна.

Полезность (ценность) информации. Полезность может быть оценена применительно к нуждам конкретных ее потребителей и оценивается по тем задачам, которые можно решить с ее помощью.

Самая ценная информация – объективная, достоверная, полная, и актуальная. При этом следует учитывать, что и необъективная, недостоверная информация (например, художественная литература), имеет большую значимость для человека. Социальная (общественная) информация обладает еще и дополнительными свойствами: имеет семантический (смысловой) характер, т. е. понятийный, так как именно в понятиях обобщаются наиболее существенные признаки предметов, процессов и явлений окружающего мира имеет языковую природу (кроме некоторых видов эстетической информации, например изобразительного искусства). Одно и то же содержание может быть выражено на разных естественных (разговорных) языках, записано в виде математических формул и т. д.

С течением времени количество информации растет, информация накапливается, происходит ее систематизация, оценка и обобщение. Это свойство назвали ростом и кумулированием информации. (Кумуляция – от лат. *cumulatio* – увеличение, скопление).

Старение информации заключается в уменьшении ее ценности с течением времени. Старит информацию не само время, а появление новой информации, которая уточняет, дополняет или отвергает полностью или частично более раннюю. Научно-техническая информация стареет быстрее, эстетическая (произведения искусства) – медленнее.

Логичность, компактность, удобная форма представления облегчает понимание и усвоение информации.

### **1.3 Лекция №3 (2 часа).**

**Тема:** «Этапы развития информационных технологий. Классификация информационных технологий»

#### **1.3.1 Вопросы лекции:**

1. Этапы развития информационных технологий.
2. Виды информационных технологий

#### **1.3.2 Краткое содержание вопросов:**

1. Этапы развития информационных технологий.

Существует несколько точек зрения на развитие информационных технологий с использованием компьютеров, которые определяются различными признаками деления.

#### ***Признак деления — вид задач и процессов обработки информации***

1-й этап (60 - 70-е гг.) — обработка данных в вычислительных центрах в режиме коллективного пользования. Основным направлением развития информационной технологии являлась автоматизация операционных рутинных действий человека.

2-й этап (с 80-х гг.) — создание информационных технологий, направленных на решение стратегических задач.

#### ***Признак деления — проблемы, стоящие на пути информатизации общества***

1-й этап (до конца 60-х гг.) характеризуется проблемой обработки больших объемов данных в условиях ограниченных возможностей аппаратных средств.

2-й этап (до конца 70-х гг.) связывается с распространением ЭВМ серии IBM/360. Проблема этого этапа — отставание программного обеспечения от уровня развития аппаратных средств.

3-й этап (с начала 80-х гг.) — компьютер становится инструментом непрофессионального пользователя, а информационные системы — средством поддержки принятия его решений. Проблемы — максимальное удовлетворение потребностей пользователя и создание соответствующего интерфейса работы в компьютерной среде.

4-й этап (с начала 90-х гг.) — создание современной технологии межорганизационных связей и информационных систем. Проблемы этого этапа весьма многочисленны. Наиболее существенными из них являются:

- выработка соглашений и установление стандартов, протоколов для компьютерной связи;
- организация доступа к стратегической информации;
- организация защиты и безопасности информации.

#### ***Признак деления — преимущество, которое приносит компьютерная технология***

1-й этап (с начала 60-х гг.) характеризуется довольно эффективной обработкой информации при выполнении рутинных операций с ориентацией на централизованное коллективное использование ресурсов вычислительных центров

2-й этап (с середины 70-х гг.) связан с появлением персональных компьютеров. Изменился подход к созданию информационных систем — ориентация смещается в сторону индивидуального пользователя для поддержки принимаемых им решений.

3-й этап (с начала 90-х гг.) связан с понятием анализа стратегических преимуществ в бизнесе и основан на достижениях телекоммуникационной технологии распределенной обработки информации.

#### ***Признак деления — виды инструментария технологии***

1-й этап (до второй половины XIX в.) — "ручная" информационная технология, инструментарий которой составляли: перо, чернильница, книга. Коммуникации осуществлялись ручным способом путем переправки через почту писем, пакетов, депеш. Основная цель технологии — представление информации в нужной форме.

2-й этап (с конца XIX в.) — "механическая" технология, инструментарий которой составляли: пишущая машинка, телефон, диктофон, оснащенная более совершенными

средствами доставки почта. Основная цель технологии — представление информации в нужной форме более удобными средствами.

3-й этап (40 — 60-е гг, XX в.) — *"электрическая"* технология, инструментарий которой составляли: большие ЭВМ и соответствующее программное обеспечение, электрические пишущие машинки, ксероксы, портативные диктофоны.

Изменяется цель технологии. Акцент в информационной технологии начинает перемещаться с формы представления информации на формирование ее содержания.

4-й этап (с начала 70-х гг.) — *"электронная"* технология, основным инструментарием которой становятся большие ЭВМ и создаваемые на их базе автоматизированные системы управления (АСУ) и информационно-поисковые системы (ИПС), оснащенные широким спектром базовых и специализированных программных комплексов. Центр тяжести технологии еще более смещается на формирование содержательной стороны информации для управленческой среды различных сфер общественной жизни, особенно на организацию аналитической работы.

5 - й этап (с середины 80-х гг.) — *"компьютерная"* ("новая") технология, основным инструментарием которой является персональный компьютер с широким спектром стандартных программных продуктов разного назначения. На этом этапе происходит процесс персонализации АСУ, который проявляется в создании систем поддержки принятия решений определенными специалистами.

## 2. Виды информационных технологий

### ***Существует несколько видов информационных технологий:***

- *Информационная технология обработки данных*
- *Информационная технология управления*
- *Автоматизация офиса*
- *Информационная технология поддержки принятия решений*
- *Информационная технология экспертных систем*

### ***Информационная технология обработки данных.***

Информационная технология обработки данных предназначена для решения хорошо структурированных задач, по которым имеются необходимые входные данные и известны алгоритмы и другие стандартные процедуры их обработки. Эта технология применяется на уровне операционной (исполнительской) деятельности (см. рис. 3.3) персонала невысокой квалификации в целях автоматизации некоторых рутинных постоянно повторяющихся операций управленческого труда.

#### *Основные компоненты:*

- *Сбор данных;*
- *Обработка данных;*
- *Хранение данных;*
- *Создание отчетов (документов.)*

### ***Информационная технология управления***

Целью информационной технологии управления является удовлетворение информационных потребностей всех без исключения сотрудников фирмы, имеющих дело с принятием решений.

Информационная технология управления направлена на создание различных видов отчетов:

*Регулярные* отчеты создаются в соответствии с установленным графиком, определяющим время их создания, например месячный анализ продаж компании.

*Специальные* отчеты создаются по запросам управленцев или когда в компании произошло что-то незапланированное.

И те, и другие виды отчетов могут иметь форму суммирующих, сравнительных и чрезвычайных отчетов.

В *суммирующих* отчетах данные объединены в отдельные группы, отсортированы и представлены в виде промежуточных и окончательных итогов по отдельным полям.

*Сравнительные* отчеты содержат данные, полученные из различных источников или классифицированные по различным признакам и используемые для целей сравнения.

*Чрезвычайные* отчеты содержат данные исключительного (чрезвычайного) характера.

#### ***Автоматизация офиса.***

Информационная технология автоматизированного офиса — это организация и поддержка коммуникационных процессов как внутри организации, так и с внешней средой на базе компьютерных сетей и других современных средств передачи и работы с информацией.

Информационная технология автоматизированного офиса включает в себя:

1. *База данных* (рассмотрим на отдельной лекции).
2. *Текстовый процессор*. Это вид прикладного программного обеспечения, предназначенный для создания и обработки текстовых документов.
3. *Электронная почта*. Электронная почта (E-mail), основываясь на сетевом использовании компьютеров, дает возможность пользователю получать, хранить и отправлять сообщения своим партнерам по сети.
4. *Аудиопочта*. Это почта для передачи сообщений голосом. Она напоминает электронную почту, за исключением того, что вместо набора сообщения на клавиатуре компьютера вы передаете его через телефон.
5. *Табличный процессор*. Он так же, как и текстовый процессор, является базовой составляющей информационной культуры любого сотрудника и автоматизированной офисной технологии.
6. *Электронный календарь*. Он предоставляет еще одну возможность использовать сетевой вариант компьютера для хранения и манипулирования рабочим расписанием управленцев и других работников организации.
7. *Компьютерные конференции и телеконференции*. Компьютерные конференции используют компьютерные сети для обмена информацией между участниками группы, решающей определенную проблему. Телеконференция включает в себя три типа конференций: аудио, видео и компьютерную.
8. *Видеотекст*. Он основан на использовании компьютера для получения отображения текстовых и графических данных на экране монитора.
9. *Хранение изображений*. Хранение изображений (imaging) является перспективной офисной технологией и основывается на использовании специального устройства — оптического распознавателя образов, позволяющего преобразовывать изображение документа или фильма в цифровой вид для дальнейшего хранения во внешней памяти компьютера.
10. *Аудиоконференции*. Они используют аудиосвязь для поддержания коммуникаций между территориально удаленными работниками или подразделениями фирмы. Наиболее простым техническим средством реализации аудиоконференций является телефонная связь, оснащенная дополнительными устройствами, дающими возможность участия в разговоре более чем двум участникам.
11. *Видеоконференции*. Они предназначены для тех же целей, что и аудиоконференций, но с применением видеоаппаратуры. Их проведение также не требует компьютера.
12. *Факсимильная связь*. Эта связь основана на использовании факс-аппарата, способного читать документ на одном конце коммуникационного канала и воспроизводить его изображение на другом.

#### ***Информационная технология поддержки принятия решений***

Главной особенностью *информационной технологии поддержки принятия решений* является качественно новый метод организации взаимодействия человека и компьютера. Выработка решения, что является основной целью этой технологии, происходит в результате итерационного процесса, в котором участвуют:

- система поддержки принятия решений в роли вычислительного звена и объекта управления;
- человек как управляющее звено, задающее входные данные и оценивающее полученный результат вычислений на компьютере.

*В состав системы поддержки принятия решений входят три главных компонента:* база данных, база моделей и программная подсистема, которая состоит из системы управления базой данных (СУБД), системы управления базой моделей (СУБМ) и системы управления интерфейсом между пользователем и компьютером.

#### *База данных*

*Система управления данными должна обладать следующими возможностями:*

- составление комбинаций данных, получаемых из различных источников, посредством использования процедур агрегирования и фильтрации;
- быстрое прибавление или исключение того или иного источника данных;
- построение логической структуры данных в терминах пользователя;
- использование и манипулирование неофициальными данными для экспериментальной проверки рабочих альтернатив пользователя;
- обеспечение полной логической независимости этой базы данных от других операционных баз данных, функционирующих в рамках фирмы

*База моделей.* Целью создания моделей являются описание и оптимизация некоторого объекта или процесса.

*Существует множество типов моделей и способов их классификации, например, по цели использования, области возможных приложений, способу оценки переменных и т. п.*

*По цели использования* модели подразделяются на *оптимизационные*, связанные с нахождением точек минимума или максимума некоторых показателей, *описательные*, описывающие поведение некоторой системы и не предназначенные для целей управления (оптимизации).

*По способу оценки*, модели классифицируются на *детерминистские*, использующие оценку переменных одним числом при конкретных значениях исходных данных, и *стохастические*, оценивающие переменные несколькими параметрами, так как исходные данные заданы вероятностными характеристиками.

*По области возможных приложений* модели разбиваются на *специализированные*, предназначенные для использования только одной системой, и *универсальные* - для использования несколькими системами.

#### *Составляющие части базы моделей:*

*Стратегические модели* используются на высших уровнях управления для установления целей организации, объемов ресурсов, необходимых для их достижения, а также политики приобретения и использования этих ресурсов. Эти модели обычно детерминистские, описательные, специализированные для использования на одной определенной фирме.

*Тактические модели* применяются управляющими среднего уровня для распределения и контроля использования имеющихся ресурсов. Обычно тактические модели реализуются как детерминистские, оптимизационные и универсальные.

*Оперативные модели* используются на низших уровнях управления для поддержки принятия оперативных решений с горизонтом, измеряемым днями и неделями. Они, как правило, детерминистские, оптимизационные и универсальные

*Математические модели* состоят из совокупности модельных блоков, модулей и процедур, реализующих математические методы. Сюда могут входить процедуры линейного программирования, статистического анализа временных рядов, регрессионного анализа и т.п. — от простейших процедур до сложных пакетов прикладных программ (ППП). Модельные блоки, модули и процедуры могут использоваться как поодиночке, так и комплексно для построения и поддержания моделей.

**Система управления интерфейсом.** Интерфейс определяет: язык пользователя; язык сообщений компьютера, организующий диалог на экране дисплея; знания пользователя.

***Информационная технология экспертных систем.***

Наибольший прогресс среди компьютерных информационных систем отмечен в области разработки *экспертных систем*, основанных на использовании искусственного интеллекта. Экспертные системы дают возможность менеджеру или специалисту получать консультации экспертов по любым проблемам, о которых этими системами накоплены знания.

Под *искусственным интеллектом* обычно понимают способности компьютерных систем к таким действиям, которые назывались бы интеллектуальными, если бы исходили от человека.

Сходство информационных технологий, используемых в экспертных системах и системах поддержки принятия решений, состоит в том, что обе они обеспечивают высокий уровень поддержки принятия решений. Однако имеются три существенных различия. Первое связано с тем, что решение проблемы в рамках систем поддержки принятия решений отражает уровень ее понимания пользователем и его возможности получить и осмыслить решение. Технология экспертных систем, наоборот, предлагает пользователю принять решение, превосходящее его возможности. Второе отличие указанных технологий выражается в способности экспертных систем пояснять свои рассуждения в процессе получения решения. Очень часто эти пояснения оказываются более важными для пользователя, чем само решение. Третье отличие связано с использованием нового компонента информационной технологии – знаний.

*Основными компонентами информационной технологии, используемой в экспертных системах, являются:*

- интерфейс пользователя;
- база знаний;
- интерпретатор;
- модуль создания системы.

***Интерфейс пользователя.*** Менеджер (специалист) использует интерфейс для ввода информации и команд в экспертную систему и получения выходной информации из нее.

***База знаний.*** Она содержит факты, описывающие проблемную область, а также логическую взаимосвязь этих фактов. Центральное место в базе знаний принадлежит правилам. *Правило* определяет, что следует делать в данной конкретной ситуации, и состоит из двух частей: условия, которое может выполняться или нет, и действия, которое следует произвести, если условие выполняется.

***Интерпретатор.*** Это часть экспертной системы, производящая в определенном порядке обработку знаний (мышление), находящихся в базе знаний. Технология работы интерпретатора сводится к последовательному рассмотрению совокупности правил (правило за правилом). Если условие, содержащееся в правиле, соблюдается, выполняется определенное действие, и пользователю предоставляется вариант решения его проблемы.

***Модуль создания системы.*** Он служит для создания набора (иерархии) правил. Существуют два подхода, которые могут быть положены в основу модуля создания системы: «пользование алгоритмических языков программирования и использование оболочек экспертных систем».

***Оболочка экспертных систем*** представляет собой готовую программную среду, которая может быть приспособлена к решению определенной проблемы путем создания соответствующей базы знаний. В большинстве случаев использование оболочек позволяет создавать экспертные системы быстрее и легче в сравнении с программированием.

#### **1. 4 Лекция №4 (2 часа).**

**Тема:** «Автоматизированное рабочее место (АРМ)»

##### **1.4.1 Вопросы лекции:**

1. Принципы организации АРМ
2. Требования к построению АРМ
3. Режимы работы АРМ
4. Виды АРМ

##### **1.4.2 Краткое содержание вопросов:**

1. Принципы организации АРМ

При покупке, организации, эксплуатации или модернизации АРМ силами фирмы следует руководствоваться следующими принципами:

1. Максимальная ориентация на конечного пользователя. Это достигается созданием специальных средств адаптации АРМ к уровню подготовки пользователя и возможностью его обучения, в т.ч. самообучения непосредственно на данном АРМ.

2. Проблемная ориентация. Специализация АРМ на решении определенного класса задач, объединенных общей технологией обработки данных, единством режимов работы и эксплуатации (ориентация на автоматизацию некоторой группы функций, постоянно выполняемых работниками сферы организационного управления).

3. Соответствие информационных потребностей пользователей используемым техническим средствам. Необходимо тщательно установить информационные потребности пользователя, обеспечивающих выполнение возложенных на него функций, а затем приступить к определению состава АРМ.

4. Принцип минимальных затрат ресурсов пользователя (временных, физических, интеллектуальных, психологических и др.) на решение задачи.

5. Принцип минимального объема занятой «оперативной» памяти у пользователя. Так как скорость переработки информации у пользователя и его пропускная способность существенно ограничены, то необходимо, чтобы в процессе работы с АРМ пользователь запоминал как можно меньше информации.

6. Принцип максимального контроля работы АРМ со стороны пользователя на всех этапах решения задачи.

7. Контакт разработчиков АРМ и их потенциальных пользователей. При выполнении такого принципа требования к АРМ формируются и уточняются параллельно с созданием системы.

Соблюдение названных принципов позволит эффективно организовывать и эксплуатировать АРМ различного назначения.

##### **2. Требования к построению АРМ**

1) Обеспечение быстродействия работы машины по формированию ответа на запросы.

2) Требование к объему запоминающего устройства.

3) Простота в изучении АРМ.

4) Возможность работы АРМ в вычислительных сетях.

АРМ выполняют децентрализованную одновременную обработку экономической информации на рабочих местах исполнителей в составе распределенной базы данных (БД). При этом они имеют выход через системное устройство и каналы связи в ПЭВМ и БД других пользователей, обеспечивая таким образом совместное функционирование ПЭВМ в процессе коллективной обработки.

В настоящее время выделяют три режима работы АРМ:

1) *Одиночный* – АРМ устанавливается на обособленном компьютере, все ресурсы которого используются одним специалистом. Оно ориентировано на индивидуальное решение задач.

2) *Групповой* на базе одного компьютера автоматизируется несколько рабочих мест.



3) *Сетевой* – каждое АРМ разрабатывается на базе одного компьютера и может работать как автономно, так и при подключении вычислительных ресурсов других компьютеров.

### 3. Режимы работы АРМ

В настоящее время выделяют три режима работы АРМ:

1) *Одиночный* - АРМ устанавливается на обособленном компьютере, все ресурсы которого используются одним специалистом. Оно ориентировано на индивидуальное решение задач.

2) *Групповой* на базе одного компьютера автоматизируется несколько рабочих мест.

3) *Сетевой* - каждое АРМ разрабатывается на базе одного компьютера и может работать как автономно, так и при подключении вычислительных ресурсов других компьютеров.

АРМ, созданные на базе персональных компьютеров, – наиболее простой и распространенный вариант автоматизированного рабочего места для работников сферы организационного управления. Такое АРМ рассматривается как система, которая в интерактивном режиме работы предоставляет конкретному работнику (пользователю) все виды обеспечения монополю на весь сеанс работы.

Создание АРМ на базе персональных компьютеров обеспечивает:

- простоту и удобство по отношению к пользователю;
- простоту адаптации к конкретным функциям пользователя;
- компактность размещения и невысокие требования к условиям эксплуатации;
- высокую надежность;
- сравнительно простую организацию технического обслуживания.

Эффективным режимом работы АРМ является его функционирование в рамках локальной вычислительной сети в качестве рабочей станции. Особенно целесообразен такой вариант, когда требуется распределить информационно-вычислительные ресурсы между несколькими пользователями.

Более сложной формой является АРМ с использованием интеллектуального терминала, а также с удаленным доступом к ресурсам центральной (главной) ЭВМ или внешней сети. В данном случае несколько ПЭВМ подключаются по каналам связи к главной ЭВМ, при этом каждая ПЭВМ может работать и как самостоятельное терминальное устройство.

В наиболее сложных системах АРМ могут через специальное оборудование подключаться не только к ресурсам главной ЭВМ сети, но и к различным информационным службам и системам различного назначения (службам новостей, национальным информационно-поисковым системам, базам данных и знаний, библиотечным системам и т.п.)

### 4. Виды АРМ

Основным предназначением рабочих мест управленческих и иных работников является обеспечение условий для комфортной, высокопроизводительной и качественной работы. Для этой цели рабочее место специалиста должно быть удобно спланировано, оснащено всем необходимым и обеспечивающим его бесперебойную информационную поддержку и рациональную организацию труда на этом месте. Применение АРМ не должно нарушать привычный ритм работы пользователя и должно концентрировать внимание пользователя на логике решаемых задач. Однако если заданное действие пользователя не выполняется, пользователь должен знать причину и информация об этом должна выдаваться пользователю.

Под *автоматизированным рабочим местом (АРМ)* понимается совокупность методических, языковых (лингвистических), аппаратных и программных средств, обеспечивающих автоматизацию функций пользователя в некоторой предметной области

и позволяющих оперативно удовлетворять его функциональные и вычислительные запросы.

Таким образом АРМ имеет проблемно-профессиональную ориентацию на конкретную предметную область. Создание автоматизированных рабочих мест предполагает, что основные операции по накоплению, хранению и переработке информации возлагаются на вычислительную технику, специалист же выполняет определенную часть ручных операций и операций, требующих творческого подхода при подготовке управленческих решений. ПК при этом работает в тесном взаимодействии с пользователем, который контролирует его действия, меняет значения отдельных параметров в ходе решения задачи, а также вводит исходные данные для решения задач и функций управления.

АРМ, как инструмент интенсификации управленческой деятельности, должен обеспечивать выполнение некоторой группы функций. Наиболее простой функцией АРМ является информационно-справочное обслуживание, причем эта функция в той или иной степени присуща любому АРМ, однако особенности ее реализации существенно зависят от категории пользователя.

Множество известных АРМ может быть классифицировано на основе следующих признаков:

- функциональная сфера использования (научная деятельность, проектирование, производственно-технологические процессы, организационное управление);

- тип используемой ЭВМ (микро-, мини-, макро ЭВМ);

- режим эксплуатации (индивидуальный, групповой, сетевой);

- квалификация пользователей (профессиональные и непрофессиональные).

Внутри каждой из выделенных групп АРМ может быть проведена более детальная классификация.

Например, АРМы организационного управления могут быть разделены на АРМ руководителей организаций и подразделений, АРМ работников плановых отделов, АРМ работников материально-технического снабжения, АРМ бухгалтеров и др. Условно все эти АРМ можно назвать *АРМ - экономиста*.

По уровню принимаемых работником решений можно выделить следующие виды АРМ:

- АРМ руководителя;

- АРМ специалиста;

- АРМ технического работника.

Особенности АРМ каждого вида определяются основными функциями и задачами работников

Основными функциями руководителя являются оперативное управление и принятие решений. Именно этими функциями определяются общие требования к *АРМ руководителя* :

- наличие достаточно развитой базы, постоянно пополняемой оперативной и достоверной информацией, причем к части этой базы, как правило, имеет доступ ограниченное число сотрудников, а к отдельным ее фрагментам – только сам руководитель;

- обеспечение возможности оперативного поиска информации;

- возможность представления информации в требуемой удобной и интегрированной форме (в том числе поступающей из разных источников);

- наличие программных средств обеспечивающих принятие решений и средств, регулирующих организаторскую и административную деятельность;

- обеспечение оперативной связи с подчиненными и различными источниками информации.

Профессиональная ориентация специалиста определяет требования к *АРМ специалиста*, однако АРМ любого специалиста должно предоставлять возможность

проводить аналитическую работу. Поэтому к общим требованиям АРМ специалиста следует отнести:

возможность работы с корпоративными БД, учрежденческими и персональными БД;

возможность моделирования анализируемых процессов;

обеспечение многофункциональности и гибкости системы.

К техническим работникам относятся секретари, операторы, машинистки, инспекторы и др. Ими, как правило, выполняется рутинная работа, требующая определенных профессиональных навыков. Это ввод информации, ведение картотек и архивов, контроль ежедневного личного плана руководителя, обработки входящей и исходящей документации и т.п. *АРМ технического работника* должно обеспечить автоматизацию этих функций.

Типовыми формами *АРМ технического работника* являются:

АРМ инспектора отдела писем;

АРМ инспектора табельного учета;

АРМ оператора обработки текстовых документов;

АРМ архивариуса.

Первые АРМ создавались на базе больших универсальных ЭВМ, позднее строились на базе малых ЭВМ. С развитием ПЭВМ для построения АРМ в основном используются технические средства ПК.

На современном этапе различным видам АРМ соответствуют различные архитектурно-технологические решения:

*АРМ на базе ПК* – наиболее простой и распространенный вариант АРМ работников сферы организационного управления. Такое АРМ представляет собой систему, которая в интерактивном режиме работы предоставляет пользователю монополию на все виды обеспечения на весь сеанс работы. Пользователь сам выполняет все функциональные обязанности по преобразованию информации. Такие АРМ обеспечивают:

простоту, удобство и дружелюбность по отношению к пользователю;

простоту адаптации к конкретным функциям пользователя;

невысокие требования к условиям эксплуатации;

высокую надежность;

простую организацию технического обслуживания.

Концептуальное отличие АРМ на базе ПЭВМ состоит в том, что АРМ открытая архитектура. Функционально, физически и эргономически настраивается на конкретного пользователя (персональный АРМ) или группу пользователей (групповой АРМ).

*АРМ как рабочая станция* обеспечивает эффективный режим работы в рамках локальной вычислительной сети. Этот режим целесообразен в условиях необходимости распределения информационно-вычислительных ресурсов между несколькими пользователями.

*АРМ с использованием ПК в качестве интеллектуального терминала* предполагает удаленный доступ к ресурсам центральной (главной) ЭВМ или внешней сети. Несколько ПЭВМ подключаются по каналам связи к главной ЭВМ, при этом каждый ПК может работать в автономном режиме. Кроме того, АРМ могут подключаться к различным информационным службам и системам общего назначения (службам новостей, информационным поисковым системам, базам данных и знаний, библиотечным системам и т.п.).

## **1. 5 Лекция №5 (2 часа).**

**Тема:** «Пользовательский интерфейс и его виды»

### **1.5.1 Вопросы лекции:**

1. Понятие пользовательского интерфейса прикладных информационных технологий

2. Развитие пользовательского интерфейса

3. Классификация пользовательского интерфейса

4. Возможности графического интерфейса

### **1.5.2 Краткое содержание вопросов:**

1. Понятие пользовательского интерфейса прикладных информационных технологий

Пользовательский интерфейс - это программных и аппаратных средств, обеспечивающих взаимодействие пользователя с компьютером. Основу такого взаимодействия составляют диалоги. Под диалогом в данном случае понимают регламентированный обмен информацией между человеком и компьютером, осуществляемый в реальном масштабе времени и направленный на совместное решение конкретной задачи. Каждый диалог состоит из отдельных процессов ввода/вывода, которые физически обеспечивают связь пользователя и компьютера. Обмен информацией осуществляется передачей сообщения.

В основном пользователь генерирует сообщения следующих типов:

-запрос информации

-запрос помощи

-запрос операции или функции

-ввод или изменение информации

В ответ пользователь получает подсказки или справки; информационные сообщения, требующие ответа; приказы, требующие действия; сообщения об ошибках и другую информацию.

Типы интерфейсов:

Интерфейсы пользователя бывают двух типов:

1)процедурно-ориентированные:

-примитивные

-меню

-со свободной навигацией

2)объектно-ориентированные:

-прямого манипулирования.

Процедурно ориентированный интерфейс использует традиционную модель взаимодействия с пользователем, основанную на понятиях «процедура» и «операция». В рамках этой модели программное обеспечение предоставляет пользователю возможность выполнения некоторых действий, для которых пользователь определяет соответствие данных и следствием выполнения которых является получение желаемого результата.

Объектно-ориентированные интерфейсы используют модель взаимодействия с пользователем, ориентированную на манипулирование объектами предметной области. В рамках этой модели пользователю предоставляется возможность напрямую взаимодействовать с каждым объектом и инициировать выполнение операций, в процессе которых взаимодействуют несколько объектов. Задача пользователя формулируется как целенаправленное изменение некоторого объекта. Объект понимается в широком смысле слова - модель БД, системы и т.д.

Процедурно-ориентированные интерфейсы:

1)Обеспечивает пользователю функции, необходимые для выполнения задач;

2)Акцент делается на задачи;

3)Пиктограммы представляют приложения, окна или операции;

4)Содержание папок и справочников отражается с помощью таблицы-списка.

Объектно-ориентированные интерфейсы:

- 1) Обеспечивает пользователю возможность взаимодействия с объектами;
- 2) Акцент делается на входные данные и результаты;
- 3) Пиктограммы представляют объекты;
- 4) Папки и справочники являются визуальными контейнерами объектов.

Примитивным называется интерфейс, который организует взаимодействие с пользователем и используется в консольном режиме. Единственное отклонение от последовательного процесса, который обеспечивается данными, заключается в организации цикла для обработки нескольких наборов данных.

Интерфейс Меню. В отличие от примитивного интерфейса, позволяет пользователю выбирать операцию из специального списка, выводимого ему программой. Эти интерфейсы предполагают реализацию множества сценариев работы, последовательность действий в которых определяется пользователями. Древовидная организация меню предполагает строго ограниченную реализацию. При этом возможны два варианта организации меню:

- каждое окно меню занимает весь экран
- на экране одновременно присутствуют несколько разноуровневых меню (Windows).

В условиях ограниченной навигации, независимо от варианта реализации, поиск пункта более чем двух уровней меню оказывается довольно сложной задачей.

Интерфейс со свободной навигацией (графический интерфейс). Поддерживает концепцию интерактивного взаимодействия с ПО, визуальную обратную связь с пользователем и возможность прямого манипулирования объектом (кнопки, индикаторы, строки состояния). В отличие от интерфейса Меню, интерфейс со свободной навигацией обеспечивает возможность осуществления любых допустимых в конкретном состоянии операций, доступ к которым возможен через различные интерфейсные компоненты («горячие» клавиши и т.д.). Интерфейс со свободной навигацией реализуется с использованием событийного программирования, что предполагает применение визуальных средств разработки (посредством сообщений).

Объектно-ориентированный интерфейс предполагает, что взаимодействие с пользователем осуществляется посредством выбора и перемещения пиктограмм соответствующей объектно-ориентированной области.

Различают одно-документные (SDI) и многодокументные (MDI) интерфейсы.

Как уже указывалось выше, интерфейс - это, прежде всего, набор правил. Как любые правила, их можно обобщить, собрать в "кодексы", сгруппировать по общему признаку. Таким образом, мы пришли к понятию "вид интерфейса" как объединение по схожести способов взаимодействия человека и компьютеров. Вкратце можно предложить следующую схематическую классификацию различных интерфейсов общения человека и компьютера.

Современными видами интерфейсов являются:

1) *Командный интерфейс*. Командный интерфейс называется так по тому, что в этом виде интерфейса человек подает "команды" компьютеру, а компьютер их выполняет и выдает результат человеку. Командный интерфейс реализован в виде пакетной технологии и технологии командной строки.

2) *WIMP - интерфейс* (Window - окно, Image - образ, Menu - меню, Pointer - указатель). Характерной особенностью этого вида интерфейса является то, что диалог с пользователем ведется не с помощью команд, а с помощью графических образов - меню, окон, других элементов. Хотя и в этом интерфейсе подаются команды машине, но это делается "опосредованно", через графические образы. Этот вид интерфейса реализован на двух уровнях технологий: простой графический интерфейс и "чистый" WIMP - интерфейс.

3) *SILK* - интерфейс (Speech - речь, Image - образ, Language - язык, Knowledge - знание). Этот вид интерфейса наиболее приближен к обычной, человеческой форме общения. В рамках этого интерфейса идет обычный "разговор" человека и компьютера. При этом компьютер находит для себя команды, анализируя человеческую речь и находя в ней ключевые фразы. Результат выполнения команд он также преобразует в понятную человеку форму. Этот вид интерфейса наиболее требователен к аппаратным ресурсам компьютера, и поэтому его применяют в основном для военных целей.

1. *Общественный интерфейс* - основан на семантических сетях.

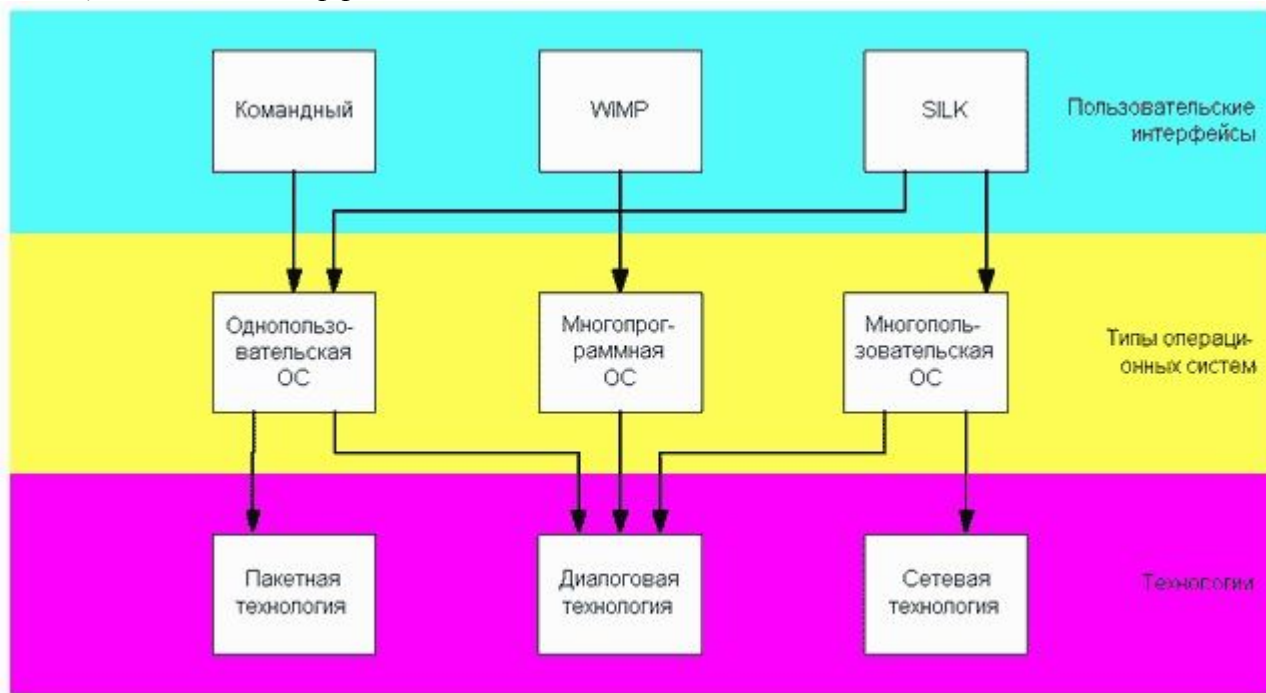


Рис. А.1. Взаимодействие типов операционных систем, пользовательских интерфейсов и технологий

## 2. Развитие пользовательского интерфейса

История графического интерфейса пользователя охватывает промежуток в пять десятилетий. За это время его основные принципы претерпели большие изменения. Впервые концепт многооконного управления системой был применен для управления системой «SAGE» 1954 г. (Semi - Automatic Ground Environment Computer), предназначенным для обнаружения военных угроз, и электронным блокнотом Ивана Сазерленда 1965 г. (США).

В 1960 году Дуглас Энгельбарт (Калифорния, США) в своем проекте «Augmentation of Human Intellect» («Прирост человеческого интеллекта») разработал первую онлайн систему (NLS). Этот компьютер уже включал в себя компьютерную мышь с курсором и несколькими «окнами», используемых для работы над гипертекстом (текстом «ветвящимся или выполняющего действия по запросу»).

Позже, в 1973 году, Энгельбарт, работая уже в компании «Xerox PARC», разработал персональный компьютер (personal computer - PC) «Alto», имеющий свой графический интерфейс, разработанный Аланом Кеем, Ларри Теслером, Дэном Инглассом и рядом других разработчиков. Он имел битовый (растровый) экран и был первым, который использовал метафору рабочего стола и графического интерфейса пользователя (GUI). Он создавался как некоммерческий продукт, однако были выпущены несколько тысяч единиц, которые активно использовались в течение многих лет. В 1975 году инженеры «Xerox PARC» продемонстрировали графический интерфейс пользователя, включающий в себя иконки (графические символы) с использованием всплывающих меню.

А в 1981 году «Xerox PARC» предоставила новаторский продукт «Star», включавший в себя многие новые разработки, в том числе уникальный интерфейс. Хотя этот продукт не был коммерчески успешным, находки, примененные здесь в пользовательском интерфейсе, оказали большое влияние на будущее таких компьютерных гигантов, как «Apple», «Microsoft» и «Sun Microsystems».

С тех пор некоторые производители создали свои собственные интерфейсы управления типа «окон» на основе независимого кода, но несущие в себе все те же основные элементы интерфейса, которые мы используем и сейчас – это WIMP – window, icon, menu, pointing device (окно, иконка, меню и кнопки управления). Широкое внедрение РС, появление многоцветных мониторов и удешевление телекоммуникационных сетей способствовало популяризации и распространению деятельности по разработке пользовательских интерфейсов и их начали рассматривать с точки зрения эстетики.

В 1995 году компания «Microsoft» под лозунгом «Компьютер в каждый дом» выпустила операционную систему «Windows 95», объединявшую в себе ранее отдельные системы «MS-DOS» и «Windows». Эта операционная система получила значительное изменение дизайна графического интерфейса, получивший название «Cairo». Главным дизайнерским ходом стали объем и тени – все элементы выглядят более реалистично и осязаемо, чем было прежде. На рынке система «Windows 95» получила безоговорочный успех и стала самой успешной операционной системой, которую когда-либо до этого разрабатывали.

Параллельно с «Microsoft» успешно развивали и совершенствовали свою среду графического интерфейса в «объеме» такие крупные компании как «Apple», «Sun Microsystems», «Commodore International» и другие.

С течением времени разработчики интерфейсов управления стремились все к большей реалистичности элементов (3D), добиваясь при этом поразительных результатов. Эти результаты как же провоцировали совершенствование выразительности и разработку визуальных и звуковых эффектов, типа сворачивания окон, перелистывания страниц, всплывания окон, появления меню и т. п.

Новое тысячелетие характеризуется бумом портативных карманных устройств, таких как смартфоны, MP3-плееры, планшеты и т. д. С середины 2000-х годов подавляющее большинство портативных устройств двигались в сторону увеличения размеров экрана. Появились сенсорно управляемые устройства, практически полностью вытеснившие устройства, управляемые нажатиями кнопок на корпусе. Все эти новшества определили новые регламенты разработки интерфейсов с учетом использования их одной рукой и жестами многозадачности. Никакие компании не могут игнорировать изменения в IT-рынке, поскольку это означало бы их заблаговременную сдачу позиций на рынке.

Сегодня наиболее распространенными операционными системами для портативных устройств являются «iOS» от «Apple», «Android» от «Google» и «Windows 8» от «Microsoft». Появление стиля интерфейса «Metro», в котором были выполнены операционные системы «Windows Phone 7» и «Windows 8» (2012 г.), произвело настоящий фурор среди графических и «digital» дизайнеров. Здесь утратился тот объем, те тени, и общая реалистичность к которым столько лет стремились разработчики интерфейсов. Дизайн элементов вернулся к плоской эстетике, применяемой в самом начале истории пользовательского интерфейса – «минимум деталей, максимум информации», простота и чистота (2D элементы). Сегодня применение этого стиля можно встретить практически везде начиная от интерфейсов, заканчивая визитками и корпоративной айдентикой.

### 3. Классификация пользовательского интерфейса

#### Структура и классификация пользовательских интерфейсов

В пользовательском интерфейсе можно условно выделить декоративную и активную составляющие. К первой относятся элементы, отвечающие за эстетическую привлекательность программного изделия.

Активные элементы подразделяются на операционные и информационные образы моделей вычислений и управляющие средства пользовательского интерфейса, посредством которых пользователь управляет программой. Управляющие средства различных классов программных изделий могут значительно различаться.

В основе управляющих средств пользовательского интерфейса лежит тот или иной интерфейсный язык. При этом роль синтаксиса играют используемые графические образы и их динамические свойства. О типах управляющих средств пользовательского интерфейса мы будем говорить, имея в виду различные формы (элементы дизайна) типизированных управляющих элементов пользовательского интерфейса определенного подкласса.

Дизайн конкретных реализаций интерфейса может включать композицию различных типов управляющих средств, информационные образы предметной области и декоративные элементы. Компоненты дизайна пользовательского интерфейса не задаются произвольно, а образуют некоторое стилевое единство.

Можно выделить поколения пользовательского интерфейса, которые характеризуются четырьмя интерфейсными стилями.

В первый период (50-е и начало 60-х гг.) компьютеры работали в основном в пакетном режиме.

Во втором периоде (с начала 60-х до начала 80-х гг.) пользователи могли взаимодействовать с компьютером путем ввода с клавиатуры команд с параметрами.

В третьем периоде (80-е годы) были созданы графические интерфейсы пользователя (GUI). Эти интерфейсы принято обозначать аббревиатурой WIMP (Windows-Icons-Menus-Pointing device), что предполагает использование окон, пиктограмм, меню и позиционирующего устройства (обычно мышь).

В настоящее время используются новые формы интерфейсов четвертого поколения, которые называют *post-WIMP*-интерфейсы. Они не используют меню, формы и панели инструментов, вместо них при задании спецификаций операций и операндов упор идет на обучающие примеры, жесты и распознавание речи.

В пользовательских интерфейсах можно выделить классы и подклассы, происхождение которых связано с используемыми базовыми техническими средствами человеко-машинного взаимодействия

#### 4. Возможности графического интерфейса

Хотя многие профессиональные программисты, работающие в среде ОС UNIX, и сегодня предпочитают пользоваться традиционными строчными средствами взаимодействия с системой, широкое распространение относительно недорогих цветных графических терминалов с высоким качеством разрешения привело к тому, что во всех современных вариантах ОС UNIX поддерживаются графические интерфейсы пользователя с системой, а пользователям предоставляются инструментальные средства для разработки графических интерфейсов с разрабатываемыми ими программами. С точки зрения конечного пользователя средства графического интерфейса, поддерживаемого в разных вариантах ОС UNIX, да и в других системах (например, MS Windows или Windows NT), примерно одинаковы по своему стилю.

*Во-первых*, во всех случаях поддерживается многооконный режим работы с экраном терминала. В любой момент времени пользователь может образовать новое окно и связать его с нужной программой, которая работает с этим окном как с отдельным терминалом. Окна можно перемещать, изменять их размер, временно закрывать и т.д.

*Во-вторых*, во всех современных разновидностях графического интерфейса поддерживается управление мышью. В случае ОС UNIX часто оказывается, что обычной клавиатурой терминала пользуются только при переходе к традиционному строчному интерфейсу (хотя в большинстве случаев по крайней мере в одном окне терминала работает один из командных интерпретаторов семейства shell).



*В-третьих*, такое распространение "мышинного" стиля работы оказывается возможным за счет использования интерфейсных средств, основанных на пиктограммах (icons) и меню. В большинстве случаев, программа, работающая в некотором окне, предлагает пользователю выбрать какую-либо выполняемую ей функцию либо путем отображения в окне набора символических образов возможных функций (пиктограмм), либо посредством предложения многоуровневого меню. В любом случае для дальнейшего выбора оказывается достаточным управления курсором соответствующего окна с помощью мыши.

Наконец, современные графические интерфейсы обладают "дружественностью по отношению к пользователю", обеспечивая возможность немедленного получения интерактивной подсказки по любому поводу. (Наверное, правильнее было бы сказать, что хорошим стилем программирования с использованием графических интерфейсов является стиль, при котором такие подсказки реально обеспечиваются.)

После перечисления всех этих общих свойств современных средств графического интерфейса может возникнуть естественный вопрос: Если в области графических интерфейсов существует такое единообразие, что особенное может быть сказано по поводу графических интерфейсов в среде ОС UNIX? Ответ достаточно прост. Да, конечный пользователь действительно в любой сегодняшней системе имеет дело примерно с одним и тем же набором интерфейсных возможностей, но в разных системах эти возможности достигаются по-разному. Как обычно, преимуществом ОС UNIX является наличие стандартизованных технологий, позволяющих создавать мобильные приложения с графическими интерфейсами.

В этой части курса мы рассмотрим базовый механизм ОС UNIX для организации оконных графических интерфейсов на основе использования разнообразных графических терминалов (оконную систему X), а также два распространенных инструментальных пакета, предназначенных для облегчения разработки графического интерфейса с прикладной программой (индустриальный, используемый сегодня практически во всех вариантах ОС UNIX пакет Motif и свободно распространяемый пакет Tcl/Tk). Каждая из упомянутых тем настолько обширна, что полная информация занимает несколько толстых книг; по поводу каждой из тем можно было бы провести отдельный курс лекций. Поэтому здесь мы приводим только краткий обзор, целью которого является создание общего представления о методах и средствах организации графического интерфейса в среде ОС UNIX.

## **Оконная система X как базовое средство графических интерфейсов в среде ОС UNIX**

Понятно, что для нормальной организации работы пользовательских программ с графическими терминалами (если учитывать отмеченные выше стандартные требования к графическому интерфейсу) требуется наличие некоторого базового слоя программного обеспечения, скрывающего аппаратные особенности терминала; обеспечивающего создание окон на экране терминала, управление этими окнами и работу с ними со стороны пользовательской программы; дающего возможность пользовательской программе реагировать на события, происходящие в соответствующем окне (ввод с клавиатуры, движение курсора, нажатие клавиш мыши и т.д.). Такой базовый слой графического программного обеспечения принято называть оконной системой.

В мире ОС UNIX предпринималось несколько попыток создания оконных систем, и большинство из них успешно использовалось практически (упомянем, например, оконную систему NeWS компании Sun Microsystems, интерфейс которой основывался на использовании языка Postscript). Однако ни одна из этих систем не выходила за пределы ведомственного использования, что, естественно, резко ограничивало мобильность программ, обладающих графическим интерфейсом. Успеха удалось добиться группе молодых исследователей и программистов из Масачусетского технологического института, которые создали оконную систему под кратким и предельно скромным

названием X (кстати, именно так правильно называть систему; по-английски ее грамотно называют не X-Window, а X window system, т.е. "оконная система X"). В настоящее время оконная система X является фактическим стандартом опорных средств графического интерфейса. Система X, дополнительные библиотеки, а также ряд готовых интерфейсных средств распространяются MIT бесплатно (относясь к категории public domain). В то же время сегодня именно оконная система X является базовым механизмом организации графических интерфейсов пользователя в большинстве UNIX-систем.

### **Общая организация X-Window**

Как кажется, оконная система X победила потому, что организация системы очень точно соответствует общей идеологии ОС UNIX. UNIX - это традиционно сетевая операционная система. Девиз Билла Джоя и всей компании Sun Microsystems "The Network is the Computer - Сеть - это компьютер" - в полной мере относится к направлению ОС UNIX в целом. Популярная ныне архитектура организации программно-аппаратных средств "клиент-сервер" всегда была совершенно естественной в мире UNIX. Специализация и разделение функций в сети - это и значит, что для пользователя компьютер и сеть неразличимы.

На этих идеях построена и оконная система X. Поскольку ОС UNIX является интерактивной операционной системой, то каждый работающий в системе пользователь взаимодействует с системой через предоставленный ему терминал (скорее всего, рабочую станцию) и, вообще говоря, вызывает программы, которые будут выполняться на других компьютерах локальной или территориально распределенной сети. Именно эти программы обеспечивают интерфейс с данным пользователем, т.е. они должны иметь возможность работать с терминалом пользователя вне зависимости от того, где они выполняются. Более того, чтобы возможности графического интерфейса этих программ удовлетворяли общим требованиям, нужно, чтобы программы могли не заботиться о таких деталях, как многооконная организация экрана, текущее расположение окон, управление мышью и т.д.

Оконная система X предоставляет в точности требуемые возможности. На стороне пользовательского терминала находится сервер системы X, обеспечивающий единообразное управление графическим терминалом вне зависимости от его специфических аппаратных характеристик. В других компьютерах сети (которые, фактически, являются серверами с точки зрения организации вычислительного процесса) установлены клиентские части системы X, создающие впечатление у выполняемой программы, что она взаимодействует с локальным терминалом, а на самом деле поддерживающие точно специфицированный протокол взаимодействий с сервером системы X.

### **Клиентская и серверная части**

Как видно из материала предыдущего пункта, клиентская и серверная части оконной системы X, хотя в целом соответствуют идеологии архитектуры "клиент-сервер", обладают тем своеобразием, что серверная часть системы находится вблизи пользователя (т.е. основного клиента вычислительной сети), а клиентская часть системы базируется на мощных серверах сети. Конечно, система X обладает достаточной гибкостью, чтобы допустить расположение серверной и клиентской частей системы в одном компьютере, в разных компьютерах одной локальной сети и удаленных компьютерах, входящих в состав территориально распределенной сети. В зависимости от конфигурации системы X-сервер может обслуживать один или несколько графических экранов, клавиатуру и мышь, реально представляя собой процесс, группу процессов или выделенное компьютерное устройство (X-терминал).

Для обеспечения требуемой гибкости, взаимодействия клиентской и пользовательской частей системы X по мере возможностей не должны были зависеть от используемых сетевой среды передачи данных и сетевых протоколов. Конечно, полная независимость - это теоретически недостижимая цель (всегда и везде), но что касается

системы X, то она действительно умеет работать в большинстве распространенных сетевых сред (в том числе, естественно, в стандартных для ОС UNIX сетях, основанных на семействе протоколов TCP/IP).

Основой взаимодействия между клиентом и сервером оконной системы X является так называемый X-протокол, представляющий собой точную спецификацию допустимых запросов от клиента к серверу и допустимых ответов сервера к клиенту. Как указывается в документации оконной системы X, X-протокол обладает следующими особо привлекательными качествами:

- При использовании этого протокола обработка взаимодействий клиента и сервера ведется единообразно, независимо от того, основана она на внутренних механизмах IPC или на реальных сетевых обменах; это позволяет добиться прозрачности сетевой среды как с точки зрения конечного пользователя, так и с точки зрения разработчика прикладных программ.

- За счет наличия строгой и не зависящей от окружения спецификации X-протокол может быть реализован на различных языках в различных операционных средах.

- X-протокол может быть реализован на основе любого надежно поддерживаемого потока байтов (обеспечиваемого внутренними механизмами IPC или внешними сетевыми механизмами); многие из пригодных механизмов являются стандартными и реализованы в большинстве архитектур.

- Для большинства (хотя и не для всех) приложений X-протокол не порождает существенных задержек при работе с графическими терминалами. Обычно задержки вызываются скорее временными потребностями самих терминалов, а не расходами на протокольные взаимодействия клиента и сервера.

Одним из клиентов оконной системы обычно является так называемый "оконный менеджер" (window manager). Это специально выделенный клиент оконной системы, обладающий полномочиями на управление расположением окон на экране терминала. Некоторые из возможностей X-протокола (связанные, например, с перемещением окон) доступны только клиентам с полномочиями оконного менеджера. Во всем остальном оконный менеджер является обычным клиентом.

#### **Базовые библиотеки**

Понятно, что теоретически любая прикладная программа, которой требуется взаимодействовать с X-сервером, могла бы работать с ним, обмениваясь сообщениями в соответствии с X-протоколом. Однако, конечно же, это неудобно. Для выполнения любого, самого простого действия с терминалом клиенту требуется обменяться несколькими сообщениями с X-сервером, причем для наиболее распространенных действий последовательность таких сообщений предопределена.

Библиотека Си-функций, которая поставляется вместе с оконной системой X и облегчает взаимодействие Си-программы с X-сервером в соответствии с X-протоколом, называется XLib. Сам X-протокол достаточно компактен, поскольку в нем специфицированы мелкие сообщения, которые, как правило, можно реально использовать только в некоторых комбинациях. XLib - это уже довольно большая библиотека (в документации системы X ее описание занимает отдельный солидный том). Это потому, что каждая функция библиотеки XLib основана на использовании нескольких протокольных сообщений (а после этого общее количество функций XLib определяется законами комбинаторики). Вместе с тем, XLib - это всего-навсего интерфейсная библиотека над X-протоколом. Если не подниматься над уровнем XLib, то для создания любого графического объекта или сценария графического протокола в каждой прикладной программе придется повторять примерно одни и те же последовательности вызовов функций XLib.

Уровнем, который позволяет использовать ранее созданные графические образы и/или заготовки интерфейсов, является библиотека Xt (X Toolkit) Intrinsics. Эта библиотека служит для создания и использования уже существующих элементов

пользовательского интерфейса, называемых виджетами (widgets). *Виджет* - это параметризуемая заготовка части пользовательского интерфейса (кнопка, часть меню, пиктограмма и т.д.), привязываемая к окну экрана терминала. Библиотека Xt Intrinsics выполнена в объектно-ориентированном стиле, так что каждый виджет представляет собой класс, который может использоваться для порождения новых классов, представляющих собой комбинированные виджеты и т.д.

По своим идеям Xt Intrinsics является мощным средством разработки пользовательских графических интерфейсов. Однако эта библиотека разрабатывалась в MIT и в основном являлась исследовательским прототипом. Хотя несколько компаний представили в public domain собственные наборы виджетов, их общее количество оказывается недостаточным для быстрого и качественного производства графических пользовательских интерфейсов. Это позволило занять лидирующее положение на коммерческом рынке инструментальному пакету консорциума Open Software Foundation (OSF) Motif.

### **Средства разработки графических интерфейсов**

Основное назначение тех средств разработки пользовательских графических интерфейсов, которые разрабатываются и поставляются отдельно от оконной системы, является облегчение создания нового графического интерфейса за счет использования существующих параметризованных заготовок. Как видно, в принципе это те же самые идеи, на которых основана объектно-ориентированная библиотека оконной системы X Xt Intrinsics.

И действительно, наиболее распространенный пакет, предназначенный для быстрой и качественной разработки графических пользовательских интерфейсов, Motif, который был спроектирован и разработан в северо-американском консорциуме OSF, в основном является развитием идей Xt Intrinsics. Motif является сугубо коммерческим продуктом. Дело дошло до того, что компания OSF запатентовала внешний интерфейс продуктов, входящих в состав Motif, чтобы не дать кому-нибудь возможность воспроизвести этот интерфейс.

Это привело к настоящему скандалу в сообществе американских программистов, потому что создало опасный прецедент патентования интерфейсов. Если можно закрыть своим авторским правом возможность повторения графического интерфейса, то почему нельзя запатентовать синтаксис языка программирования, интерфейс операционной системы и т.д.? Однако строгость американских законов не оправдывается необязательностью их исполнения, и поэтому недовольные свободолюбивые американские программисты ропщат, но терпят, а тем временем пытаются сагитировать восточно-европейских программистов (не так сильно зависящих от американских законов) на нелегальную свободно доступную реализацию интерфейсов Motif.

С другой стороны, сравнительно недавно (4-5 лет тому назад) в Калифорнийском университете г. Беркли был создан альтернативный механизм под названием Tcl/Tk. Этот механизм основан на наличии специализированного командного языка, предназначенного для описания графических пользовательских интерфейсов, соответствующего интерпретатора и библиотеки ранее разработанных заготовок интерфейсов. Пакет Tcl/Tk распространяется (вместе с полной документацией) свободно, и многие профессиональные программисты находят его более удобным, чем Motif.

### **Пакет Motif**

*Motif* (официальное название этого продукта - OSF/Motif) представляет собой программный пакет, включающий оконный менеджер, набор вспомогательных утилит, а также библиотеку классов, построенных на основе Xt Intrinsics. Для конечных пользователей оконных систем, опирающихся на Motif, основной интерес представляет менеджер окон, хотя, скорее всего, вы не сможете определить, применяется ли оконный менеджер Motif в используемой вами установке.

Для разработчиков же графических интерфейсов важны все три компонента Motif. Новый интерфейс разрабатывается в графическом же режиме с использованием оконного менеджера. При этом полезно использование утилит Motif и необходимо использование библиотеки классов Motif.

Библиотека классов Motif является расширением библиотеки Xt Intrinsics с целью придания этой библиотеке практического смысла (по-другому можно сказать, что Motif - это то, чем должен был бы быть Xt, если бы при его создании ставились коммерческие цели). Все графические объекты (правильнее сказать, классы) Xt Intrinsics включаются в библиотеку классов Motif, хотя в ней используются другие имена.

Но Motif существенно расширяет возможности Xt Intrinsics. В его библиотеке поддерживается большое число классов, позволяющих создавать меню, "нажимаемые" кнопки и т.д. Основное назначение этих классов - определение новых виджетов, связанных с окнами.

Однако в Motif поддерживается и новый вид графических объектов (их классов) - так называемые гаджеты (gadgets). *Гаджет* отличается от виджета тем, что соответствующий класс также может использоваться для создания элементов интерфейса, но графический объект не привязывается к определенному окну. При отображении на экран гаджета используется окно объекта, относящегося к суперклассу класса гаджета.

Понятно, что здесь мы не можем привести подробное описание Motif (еще раз повторим, что соответствующий материал содержится в нескольких солидных книгах). Однако основная идея должна быть понятна: развитая библиотека классов языка Си++, возможности применения этих классов при использовании обычного стиля программирования и поддержка визуального программирования с немедленным отображением получающихся графических объектов.

### **Язык и интерпретатор Tcl/Tk**

Продукт Tcl/Tk в действительности представляет собой два связанных программных пакета, которые совместно обеспечивают возможность разработки и использования приложений с развитым графическим пользовательским интерфейсом. Название Tcl относится к "командному языку инструментальных средств - tool command language", и, как не странно, его рекомендуется произносить "тикл". Это простой командный язык для управления приложениями и расширения их возможностей. Язык Tcl является "встраиваемым": его интерпретатор реализован в виде библиотеки функций языка Си, так что интерпретатор может быть легко пристыкован к любой прикладной программе, написанной на языке Си.

Tk (рекомендуемое произношение - "ти-кей") является библиотекой Си-функций, ориентированной на облегчение создания пользовательских графических интерфейсов в среде оконной системы X (т.е., по сути дела, некоторый аналог Xt Intrinsics). С другой стороны, аналогично тому, как это делается в командных языках семейства shell, функции библиотеки Tk являются командами языка Tcl, так что любой программист может расширить командный репертуар языка Tcl путем написания новой функции на языке Си.

Совместно, Tcl и Tk обеспечивают четыре преимущества для разработчиков приложений и пользователей (мы используем здесь авторские тексты разработчиков). Во-первых, наличие командного языка Tcl дает возможность в каждом приложении использовать мощный командный язык. Все, что требуется от разработчика приложения, чтобы удовлетворить его/ее специфические потребности, - это создать несколько новых команд Tcl, требующихся приложению (и, возможно, другим приложениям - явно традиционный стиль командного программирования в ОС UNIX). После этого нужно связать прикладную программу с интерпретатором Tcl и пользоваться полными возможностями командного языка.

Вторым преимуществом использования Tcl/Tk является возможность быстрой разработки графических интерфейсов. Многие интересные оконные приложения могут быть написаны в виде скриптов языка Tcl без привлечения языков Си или Си++ (а Tcl

позволяет скрыть многие несущественные детали). Как утверждают разработчики Tcl/Tk, пользователи оказываются способными к созданию новых графических интерфейсов уже после нескольких часов знакомства с продуктом. Другой особенностью языка Tcl, способствующей быстрой разработке оконных приложений, является то, что язык является интерпретируемым. Можно опробовать новую идею интерфейса, выражающуюся в сотнях или тысячах строк кода на языке Tcl, без потребности вызова новых программных средств, путем простого нажатия на клавишу мыши (не наблюдая существенных задержек при использовании современных рабочих станций).

Третьим преимуществом языка Tcl является то, что его можно применять в качестве языка "склейки" приложений. Например, любое основанное на Tcl и использующее Tk оконное приложение может направить свой скрипт любому другому аналогично ориентированному приложению. С использованием Tcl/Tk можно создавать приложения, работающие в стиле мультимедиа, и опять же они смогут обмениваться скриптами, поскольку пользуются общим интерпретатором командного языка Tcl и общей внешней библиотекой Tk.

Наконец, четвертым удобством интегрированного пакета Tcl/Tk является удобство пользователей. Для написания нового приложения в среде Tcl/Tk достаточно выучить несколько совершенно необходимых команд, и этого окажется достаточно. Другими словами оказывается возможным инкрементальный (пошаговый) стиль погружения в предмет. Такая возможность всегда радует сердце и греет душу.

Впрочем, заметим, что далеко не все программисты разделяют выраженное выше глубоко радостное отношение разработчиков Tcl/Tk к своему продукту.

## **1. 6 Лекция №6 (2 часа).**

**Тема:** «Корпоративные информационные системы»

### **1.6.1 Вопросы лекции:**

1. Технология групповой работы
2. Понятие корпоративной системы и сети
3. Структура корпоративной сети
4. Пример построения корпоративной сети

### **1.6.2 Краткое содержание вопросов:**

1. Технология групповой работы

История и терминология корпоративных сетей тесно связана с историей зарождения Интернет и World Wide Web. Поэтому не мешает вспомнить, как появились самые первые сетевые технологии, которые привели к созданию современных корпоративных (ведомственных), территориальных и глобальных сетей.

Интернет начинался в 60-х годах как проект Министерства Обороны США. Возросшая роль компьютера вызвала к жизни потребности как разделения информации между разными зданиями и локальными сетями, так и поддержания общей работоспособности системы при выходе из строя отдельных компонентов. Интернет базируется на основе набора протоколов, которые позволяют распределенным сетям направлять и передавать информацию друг другу независимо; если один узел сети по какой-то причине недоступен, информация достигает конечного пункта назначения через другие узлы, которые в данный момент в рабочем состоянии. Разработанный для этой цели протокол получил название Internetworking Protocol (IP). (То же самое означает акроним TCP/IP.)

С тех пор IP протокол стал общепринятым в военных ведомствах как способ сделать информацию общедоступной. Так как множество проектов этих ведомств выполнялось в различных исследовательских группах в университетах по всей стране, а способ обмена информацией между гетерогенными сетями оказался весьма эффективным, применение этого протокола быстро вышло за пределы военных ведомств. Его начали использовать и в исследовательских институтах НАТО и в университетах Европы. Сегодня протокол IP, а следовательно, и Интернет являются всеобщим мировым стандартом.

В конце восьмидесятых перед Интернетом встала новая проблема. Сначала информация представляла собой либо электронные письма, либо простые файлы данных. Для передачи их были выработаны соответствующие протоколы. Теперь же возник целый ряд файлов нового типа, объединяемых обычно названием multimedia, содержащие как изображения и звуки, так и гиперссылки, позволяющие пользователям перемещаться как внутри одного документа, так и между разными документами, содержащими связанную между собой информацию.

В 1989 году Лаборатория Физики Элементарных Частиц Европейского Центра Ядерных Исследований (CERN) успешно стартовала новый проект, целью которого являлось создание стандарта передачи такого рода информации через Интернет. Основными компонентами этого стандарта были форматы файлов multimedia, гипертекстовых файлов а также протокол получения таких файлов по сети. Формат файлов был назван HyperText Markup Language (HTML). Он являлся упрощенным вариантом более общего стандарта Standard General Markup Language (SGML). Протокол обслуживания запросов получил название HyperText Transfer Protocol (HTTP). В целом это выглядит следующим образом: сервер, на котором работает программа, обслуживающая HTTP протокол (HTTP demon), посылает HTML файлы по запросу клиентов Интернет. Эти два стандарта составили основу для принципиально нового типа досупа к компьютерной информации. Стандартные multimedia файлы теперь могут быть не только получены по запросу пользователя, но и существовать и отображаться как часть другого документа. Так как файл содержит гиперссылки на другие документы, которые могут

находиться на других компьютерах, пользователь может добраться до этой информации легким нажатием кнопки мыши. Это принципиально снимает сложность обращения к информации в распределенной системе. Файлы multimedia в этой технологии традиционно называются страницами. Страницей также называется информация, которая пересылается клиентской машине в ответ на каждый запрос. Причина этого в том, что документ обычно состоит из множества отдельных частей, связанных между собой гиперссылками. Такое разбиение позволяет пользователю самому решать, какие именно части хочет он видеть перед собой, позволяет сэкономить его время и уменьшить сетевой трафик. Программный продукт, который использует непосредственно пользователь, обычно называется браузером (от слова browse — пастись) или навигатором. Большая часть из них позволяет автоматически получить и отобразить определенную страницу, на которой размещены ссылки на документы, к которым пользователь обращается наиболее часто. Эта страница называется home page (домашняя), для доступа к ней обычно предусматривается отдельная кнопка. Каждый нетривиальный документ обычно снабжается специальной страницей, аналогичной разделу «Содержание» в книге. С нее обычно начинается изучение документа, поэтому она также часто называется домашней страницей. Поэтому в общем под домашней страницей понимается некоторый индекс, входная точка в информацию определенного вида. Обычно в само название входит определение этого раздела, например, Домашняя Страница компании Микрософт. С другой стороны, каждый документ может быть доступен из множества других документов. Все пространство ссылающихся друг на друга документов в Интернет получило название World Wide Web (мировая паутина) аббревиатуры WWW или W3). Система документов полностью распределена, а автор не имеет даже возможности проследить все ссылки на свой документ, существующие в Интернете. Сервер, предоставляющий доступ к этим страницам, может регистрировать всех тех, кто читает такой документ, но не тех, кто ссылается на него. Ситуация обратная существующей в мире печатной продукции. Во многих исследовательских областях существуют периодически издаваемые индексы статей на какую-то тему, однако невозможно проследить всех тех, кто читает тот или иной документ. Здесь же мы знаем тех, кто читал (имел доступ) к документу, но не знаем, кто ссылался на него. Другая интересная особенность состоит в том, что при такой технологии становится невозможно следить за всей информацией, доступной через WWW. Информация появляется и исчезает непрерывно, при отсутствии какого-то ни было центрального управления. Однако этого не стоит пугаться, то же происходит и в мире печатной продукции. Мы не пытаемся копить старые газеты, если имеем каждый день свежие, причем усилия при этом ничтожны.

Клиентские программные продукты, получающие и отображающие файлы HTML, называются браузерами. Первым из графических браузеров назывался Mosaic, и сделан он был в Университете Иллинойса (University of Illinois). Многие из современных браузеров базируются на этом продукте. Однако в силу стандартизации протоколов и форматов, можно использовать любой совместимый программный продукт. Системы просмотра существуют в большинстве основных клиентских систем, способных поддерживать интеллектуальные окна. Здесь можно назвать MS/Windows, Macintosh, системы X-Window и OS/2. Есть также системы просмотра для тех ОС, где окна не используются — они выводят на экран текстовые фрагменты документов, к которым осуществляется доступ.

Присутствие систем просмотра на таких разнородных платформах имеет большое значение. Операционные среды на машине автора, сервере и клиенте не зависят друг от друга. Любой клиент может получить доступ и просмотреть документы, созданные с использованием HTML и соответствующих стандартов, и передаваемые через HTTP-сервер, вне зависимости от того, в какой операционной среде они были созданы или откуда поступили. HTML также поддерживает разработку форм и функции обратной связи. Это означает, что пользовательский интерфейс и при запросе, и при получении данных позволяет выходить за пределы принципа «укажи и щелкни».



Многие станции, в том числе Amdahl, написали интерфейсы для взаимодействия HTML-форм и старых приложений, создав для последних универсальный клиентский пользовательский интерфейс. Это дает возможность писать клиент-серверные приложения, не думая о кодировании на уровне клиента. В сущности, уже появляются программы, в которых клиент рассматривается как система просмотра. В качестве примера можно привести интерфейс WOW корпорации Oracle, который заменяет собой Oracle Forms и Oracle Reports. Хотя эта технология еще очень молода, она уже способна изменить ситуацию в области управления информацией настолько, насколько в свое время использование полупроводников и микропроцессоров изменило мир компьютеров. Она позволяет превращать функции в отдельные модули и упрощать приложения, поднимая нас на новый уровень интеграции, который больше соответствует бизнес-функциям работы предприятия.

Информационная перегрузка — проклятие нашего времени. Технологии, которые создавались, чтобы облегчить эту проблему, только усугубили ее. Это неудивительно: стоит взглянуть на содержимое мусорных корзин (обычных или электронных) рядового сотрудника, имеющего дело с информацией. Даже если не считать кучи неизбежного рекламного «мусора» в почте, большая часть информации отправляется такому сотруднику просто «на тот случай», что она ему понадобится. Добавьте к этому «несвоевременную» информацию, которая скорее всего понадобится, но позже — и вот вам основное содержимое мусорной корзины. Сотрудник скорее всего будет хранить половину информации, которая «может понадобиться» и всю информацию, которая наверняка понадобится в будущем. Когда в ней возникнет необходимость, ему придется иметь дело с громоздким, плохо структурированным архивом персональной информации, и на этом этапе могут возникнуть дополнительные сложности из-за того, что она хранится в файлах разных форматов на разных носителях. Появление ксероксов сделало ситуацию с информацией, «которая может вдруг потребоваться», еще хуже. Количество копий вместо того, чтобы уменьшаться, только увеличивается. Электронная почта только усугубила проблему. Сегодня «публикатор» информации может создавать свой, личный список рассылки и при помощи одной команды отправлять практически неограниченное количество копий «на тот случай», что они могут понадобиться. Некоторые из таких распространителей информации понимают, что их списки никуда не годятся, но вместо того, чтобы их исправить, они помещают в начало сообщения пометку примерно такого содержания: «Если вас не интересует ..., уничтожьте это сообщение». Письмо все равно будет забивать почтовый ящик, и адресату в любом случае придется потратить время на ознакомление с ним и его уничтожение. Прямая противоположность информации «которая может пригодиться» — «своевременная» информация, или информация, на которую есть спрос. От компьютеров и сетей ждали помощи в работе именно с этим видом информации, но пока они с этим не справляются. Раньше существовало два основных метода доставки своевременной информации.

При использовании первого из них информация распределялась между приложениями и системами. Чтобы получить к ней доступ, пользователю надо было изучить, а потом постоянно выполнять множество сложных процедур доступа. Когда доступ бывал получен, каждое приложение требовало своего интерфейса. Сталкиваясь с такими трудностями, пользователи обычно просто отказывались от получения своевременной информации. Они были способны освоить доступ к одному-двум приложениям, но на остальное их уже не хватало.

Чтобы решить эту проблему, на некоторых предприятиях делались попытки накапливать всю распределенную информацию на одной главной системе. В результате пользователь получал единый способ доступа и единый интерфейс. Однако, поскольку в этом случае все запросы предприятия обрабатывались централизованно, эти системы росли и усложнялись. Прошло более десяти лет, а многие из них все еще не заполнены информацией из-за высокой стоимости ее ввода и поддержки. Были здесь и другие

проблемы. Сложность таких унифицированных систем затрудняла их модификацию и использование. Чтобы поддерживать дискретные данные процессов транзакций, разрабатывался инструментарий для управления такими системами. За последнее десятилетие данные, с которыми мы имеем дело, стали гораздо сложнее, что затрудняет процесс информационной поддержки. Изменение характера информационных потребностей и то, насколько трудно в этой области даются изменения, породили эти большие, централизованно управляемые системы, тормозящие выполнение запросов на уровне предприятия.

Web-технология предлагает новый подход к доставке информации «по требованию». Поскольку она поддерживает авторизацию и публикацию распределенной информации, а также управление ею, новая технология не приводит к таким сложностям, как старые централизованные системы. Документы составляют, поддерживают и публикуют непосредственно авторы, им не приходится просить программистов создавать новые формы для ввода данных и программы создания отчетов. Имея дело с новыми системами просмотра, пользователь может получать и просматривать информацию из распределенных источников и систем при помощи простого унифицированного интерфейса, не имея при этом ни малейшего понятия о серверах, к которым он на самом деле получают доступ. Эти простые технологические изменения произведут революцию в информационных инфраструктурах и кардинально изменят работу наших организаций.

Главная отличительная черта этой технологии — то, что управление потоком информации находится в руках не ее создателя, но потребителя. Если у пользователя есть возможность легко получать и просматривать информацию по мере необходимости, ее больше не придется посылать к нему «на случай», если она потребуется. Процесс публикации теперь может быть независим от автоматического распространения информации. Это относится к формам, отчетам, стандартам, планированию встреч, инструментарию поддержки продаж, обучающим материалам, графикам и массе других документов, обычно забивающих наши мусорные корзины. Чтобы система заработала, нужна, как сказано выше, не только новая информационная инфраструктура, но и новый подход, новая культура. Как создатели информации, мы должны научиться публиковать ее, не распространяя, как пользователи — проявлять больше ответственности при определении и отслеживании своих информационных запросов, активно и эффективно получая информацию, если она нам нужна.

## 2. Понятие корпоративной системы и сети

Прежде, чем говорить о частных (корпоративных) сетях, нужно определить, что эти слова означают. В последнее время это словосочетание стало настолько распространенным и модным, что начало терять смысл. В нашем понимании корпоративная сеть — система, обеспечивающая передачу информации между различными приложениями, используемыми в системе корпорации. Исходя из этого вполне абстрактного определения, мы рассмотрим различные подходы к созданию таких систем и постараемся наполнить понятие корпоративной сети конкретным содержанием. При этом мы считаем, что сеть должна быть максимально универсальной, то есть допускать интеграцию уже существующих и будущих приложений с минимально возможными затратами и ограничениями.

Корпоративная сеть, как правило, является территориально распределенной, т.е. объединяющей офисы, подразделения и другие структуры, находящиеся на значительном удалении друг от друга. Часто узлы корпоративной сети оказываются расположенными в различных городах, а иногда и странах. Принципы, по которым строится такая сеть, достаточно сильно отличаются от тех, что используются при создании локальной сети, даже охватывающей несколько зданий. Основное отличие состоит в том, что территориально распределенные сети используют достаточно медленные (на сегодня — десятки и сотни килобит в секунду, иногда до 2 Мбит/с.) арендованные линии связи. Если при создании локальной сети основные затраты приходятся на закупку оборудования и

прокладку кабеля, то в территориально-распределенных сетях наиболее существенным элементом стоимости оказывается арендная плата за использование каналов, которая быстро растет с увеличением качества и скорости передачи данных. Это ограничение является принципиальным, и при проектировании корпоративной сети следует предпринимать все меры для минимизации объемов передаваемых данных. В остальном же корпоративная сеть не должна вносить ограничений на то, какие именно приложения и каким образом обрабатывают переносимую по ней информацию.

Под приложениями мы здесь понимаем как системное программное обеспечение — базы данных, почтовые системы, вычислительные ресурсы, файловый сервис и прочее — так и средства, с которыми работает конечный пользователь. Основными задачами корпоративной сети оказываются взаимодействие системных приложений, расположенных в различных узлах, и доступ к ним удаленных пользователей.

Первая проблема, которую приходится решать при создании корпоративной сети — организация каналов связи. Если в пределах одного города можно рассчитывать на аренду выделенных линий, в том числе высокоскоростных, то при переходе к географически удаленным узлам стоимость аренды каналов становится просто астрономической, а качество и надежность их часто оказывается весьма невысокими. Естественным решением этой проблемы является использование уже существующих глобальных сетей. В этом случае достаточно обеспечить каналы от офисов до ближайших узлов сети. Задачу доставки информации между узлами глобальная сеть при этом возьмет на себя. Даже при создании небольшой сети в пределах одного города следует иметь в виду возможность дальнейшего расширения и использовать технологии, совместимые с существующими глобальными сетями.

Часто первой, а то и единственной такой сетью, мысль о которой приходит в голову, оказывается Internet. Использование Internet в корпоративных сетях В зависимости от решаемых задач Internet можно рассматривать на различных уровнях. Для конечного пользователя это прежде всего всемирная система предоставления информационных и почтовых услуг. Сочетание новых технологий доступа к информации, объединяемых понятием World Wide Web, с дешевой и общедоступной глобальной системой компьютерной связи Internet фактически породило новое средство массовой информации, которое часто называют просто the Net — Сеть. Тот, кто подключается к этой системе, воспринимает ее просто как механизм, дающий доступ к определенным услугам. Реализация же этого механизма оказывается абсолютно несущественной.

При использовании Internet в качестве основы для корпоративной сети передачи данных выясняется очень интересная вещь. Оказывается, Сеть сетью-то как раз и не является. Это именно Internet — междусетие. Если заглянуть внутрь Internet, мы увидим, что информация проходит через множество абсолютно независимых и по большей части некоммерческих узлов, связанных через самые разнородные каналы и сети передачи данных. Бурный рост услуг, предоставляемых в Internet, приводит к перегрузке узлов и каналов связи, что резко снижает скорость и надежность передачи информации. При этом поставщики услуг Internet не несут никакой ответственности за функционирование сети в целом, а каналы связи развиваются крайне неравномерно и в основном там, где государство считает нужным вкладывать в это средства. Соответственно, нет никаких гарантий качества работы сети, скорости передачи данных и даже просто достижимости ваших компьютеров. Для задач, в которых критичными являются надежность и гарантированное время доставки информации, Internet — далеко не лучшее решение. Кроме того, Internet привязывает пользователей к одному протоколу — IP. Это хорошо, когда мы пользуемся стандартными приложениями, работающими с этим протоколом. Использование же с Internet любых других систем оказывается делом непростым и дорогим. Если у вас возникает необходимость обеспечить доступ мобильных пользователей к вашей частной сети — Internet также не самое лучшее решение.

Казалось бы, больших проблем здесь быть не должно — поставщики услуг Internet есть почти везде, возьмите портативный компьютер с модемом, позвоните и работайте. Однако поставщик, скажем, в Новосибирске, не имеет никаких обязательств перед вами, если вы подключились к Internet в Москве. Денег за услуги он от вас не получает и доступа в сеть, естественно, не предоставит. Либо надо заключать с ним соответствующий контракт, что вряд ли разумно, если вы оказались в двухдневной командировке, либо звонить из Новосибирска в Москву.

Еще одна проблема Internet, широко обсуждаемая в последнее время, — безопасность. Если мы говорим о частной сети, вполне естественным представляется защитить передаваемую информацию от чужого взгляда. Непредсказуемость путей информации между множеством независимых узлов Internet не только повышает риск того, что какой-либо не в меру любопытный оператор сети может сложить ваши данные себе на диск (технически это не так сложно), но и делает невозможным определение места утечки информации. Средства шифрования решают проблему лишь частично, поскольку применимы в основном к почте, передаче файлов и т.п. Решения же, позволяющие с приемлемой скоростью шифровать информацию в реальном времени (например, при непосредственной работе с удаленной базой данных или файл-сервером), малодоступны и дороги. Другой аспект проблемы безопасности опять же связан с децентрализованностью Internet — нет никого, кто мог бы ограничить доступ к ресурсам вашей частной сети. Поскольку это открытая система, где все видят всех, то любой желающий может попробовать попасть в вашу офисную сеть и получить доступ к данным или программам. Есть, конечно, средства защиты (для них принято название Firewall — по-русски, точнее по-немецки «брандмауэр» — противопожарная стена). Однако считать их панацеей не стоит — вспомните про вирусы и антивирусные программы. Любую защиту можно сломать, лишь бы это окупало стоимость взлома. Необходимо также отметить, что сделать подключенную к Internet систему неработоспособной можно, и не вторгаясь в вашу сеть. Известны случаи несанкционированного доступа к управлению узлами сети, или просто использования особенностей архитектуры Internet для нарушения доступа к тому или иному серверу. Таким образом, рекомендовать Internet как основу для систем, в которых требуется надежность и закрытость, никак нельзя. Подключение к Internet в рамках корпоративной сети имеет смысл, если вам нужен доступ к тому громадному информационному пространству, которое собственно и называют Сетью.

Корпоративная сеть — это сложная система, включающая тысячи самых разнообразных компонентов: компьютеры разных типов, начиная с настольных и кончая мейнфреймами, системное и прикладное программное обеспечение, сетевые адаптеры, концентраторы, коммутаторы и маршрутизаторы, кабельную систему. Основная задача системных интеграторов и администраторов состоит в том, чтобы эта громоздкая и весьма дорогостоящая система как можно лучше справлялась с обработкой потоков информации, циркулирующих между сотрудниками предприятия и позволяла принимать им своевременные и рациональные решения, обеспечивающие выживание предприятия в жесткой конкурентной борьбе. А так как жизнь не стоит на месте, то и содержание корпоративной информации, интенсивность ее потоков и способы ее обработки постоянно меняются. Последний пример резкого изменения технологии автоматизированной обработки корпоративной информации у всех на виду — он связан с беспрецедентным ростом популярности Internet в последние 2 — 3 года. Изменения, причиной которых стал Internet, многогранны. Гипертекстовая служба WWW изменила способ представления информации человеку, собрав на своих страницах все популярные ее виды — текст, графику и звук. Транспорт Internet — недорогой и доступный практически всем предприятиям (а через телефонные сети и одиночным пользователям) — существенно облегчил задачу построения территориальной корпоративной сети, одновременно выдвинув на первый план задачу защиты корпоративных данных при передаче их через в высшей степени общедоступную публичную сеть с многомиллионным «населением».

### 3. Структура корпоративной сети

При построении территориально распределенной сети могут использоваться все описанные выше технологии. Для подключения удаленных пользователей самым простым и доступным вариантом является использование телефонной связи. Там, где это возможно, могут использоваться сети ISDN. Для объединения узлов сети в большинстве случаев используются глобальные сети передачи данных. Даже там, где возможна прокладка выделенных линий (например, в пределах одного города) использование технологий пакетной коммутации позволяет уменьшить количество необходимых каналов связи и — что немаловажно — обеспечить совместимость системы с существующими глобальными сетями. Подключение корпоративной сети к Internet оправдано, если вам нужен доступ к соответствующим услугам. Использовать Internet как среду передачи данных стоит только тогда, когда другие способы недоступны и финансовые соображения перевешивают требования надежности и безопасности. Если вы будете использовать Internet только в качестве источника информации, лучше пользоваться технологией «соединение по запросу» (dial-on-demand), т.е. таким способом подключения, когда соединение с узлом Internet устанавливается только по вашей инициативе и на нужное вам время. Это резко снижает риск несанкционированного проникновения в вашу сеть извне. Простейший способ обеспечить такое подключение — использовать дозвон до узла Internet по телефонной линии или, если возможно, через ISDN. Другой, более надежный способ обеспечить соединение по запросу — использовать выделенную линию и протокол X.25 или — что гораздо предпочтительнее — Frame Relay. В этом случае маршрутизатор с вашей стороны должен быть настроен так, чтобы разрывать виртуальное соединение при отсутствии данных в течении определенного времени и вновь устанавливать его только тогда, когда данные появляются с вашей стороны. Широко распространенные способы подключения с использованием PPP или HDLC такой возможности не дают. Если же вы хотите предоставлять свою информацию в Internet — например, установить WWW или FTP сервер, соединение по запросу оказывается неприменимым. В этом случае следует не только использовать ограничение доступа с помощью Firewall, но и максимально изолировать сервер Internet от остальных ресурсов. Хорошим решением является использование единственной точки подключения к Internet для всей территориально распределенной сети, узлы которой связаны друг с другом с помощью виртуальных каналов X.25 или Frame Relay. В этом случае доступ из Internet возможен к единственному узлу, пользователи же в остальных узлах могут попасть в Internet с помощью соединения по запросу.

Для передачи данных внутри корпоративной сети также стоит использовать виртуальные каналы сетей пакетной коммутации. Основные достоинства такого подхода — универсальность, гибкость, безопасность — были подробно рассмотрены выше. В качестве виртуальной сети при построении корпоративной информационной системы может использоваться как X.25, так и Frame Relay. Выбор между ними определяется качеством каналов связи, доступностью услуг в точках подключения и — не в последнюю очередь — финансовыми соображениями. На сегодня затраты при использовании Frame Relay для междугородной связи оказываются в несколько раз выше, чем для сетей X.25. С другой стороны, более высокая скорость передачи информации и возможность одновременно передавать данные и голос могут оказаться решающими аргументами в пользу Frame Relay. На тех участках корпоративной сети, где доступны арендованные линии, более предпочтительной является технология Frame Relay. В этом случае возможно как объединение локальных сетей и подключение к Internet, так и использование тех приложений, которые традиционно требуют X.25. Кроме того, по этой же сети возможна телефонная связь между узлами. Для Frame Relay лучше использовать цифровые каналы связи, однако даже на физических линиях или каналах тональной

частоты можно создать вполне эффективную сеть, установив соответствующее канальное оборудование. Хорошие результаты дает применение модемов Motorola 326x SDC, имеющих уникальные возможности коррекции и компрессии данных в синхронном режиме. Благодаря этому удается — ценой внесения небольших задержек — значительно поднять качество канала связи и достичь эффективной скорости до 80 кбит/сек и выше. На физических линиях небольшой протяженности могут использоваться также short-range модемы, обеспечивающие достаточно высокие скорости. Однако здесь необходимо высокое качество линии, поскольку short-range модемы никакой коррекции ошибок не поддерживают. Широко известны short-range модемы RAD, а также оборудование PairGain, позволяющее достичь скорости 2 Мбит/с на физических линиях длиной около 10 км. Для подключения удаленных пользователей к корпоративной сети могут использоваться узлы доступа сетей X.25, а также собственные коммуникационные узлы. В последнем случае требуется выделение нужного количества телефонных номеров (или каналов ISDN), что может оказаться слишком дорого. Если нужно обеспечить подключение большого количества пользователей одновременно, то более дешевым вариантом может оказаться использование узлов доступа сети X.25, даже внутри одного города.

Корпоративная сеть — это достаточно сложная структура, использующая различные типы связи, коммуникационные протоколы и способы подключения ресурсов. С точки зрения удобства построения и управляемости сети следуют ориентироваться на однотипное оборудование одного производителя. Однако практика показывает, что поставщиков, предлагающих максимально эффективные решения для всех возникающих задач, не существует. Работая сеть всегда является результатом компромисса — либо это однородная система, неоптимальная с точки зрения цены и возможностей, либо более сложное в установке и управлении сочетание продуктов различных производителей. Далее мы рассмотрим средства построения сетей нескольких ведущих производителей и дадим некоторые рекомендации по их использованию.

Все оборудование сетей передачи данных можно условно разделить на два больших класса —

1. периферийное, которое используется для подключения к сети конечных узлов, и
2. магистральное или опорное, реализующее основные функции сети (коммутацию каналов, маршрутизацию и т.д.).

Четкой границы между этими типами нет — одни и те же устройства могут использоваться в разном качестве или совмещать те и другие функции. Следует отметить, что к магистральному оборудованию обычно предъявляются повышенные требования в части надежности, производительности, количества портов и дальнейшей расширяемости.

Периферийное оборудование является необходимым компонентом всякой корпоративной сети. Функции же магистральных узлов может брать на себя глобальная сеть передачи данных, к которой подключаются ресурсы. Как правило, магистральные узлы в составе корпоративной сети появляются только в тех случаях, когда используются арендованные каналы связи или создаются собственные узлы доступа. Периферийное оборудование корпоративных сетей с точки зрения выполняемых функций также можно разделить на два класса.

Во-первых, это маршрутизаторы (routers), служащие для объединения однородных LAN (как правило, IP или IPX) через глобальные сети передачи данных. В сетях, использующих IP или IPX в качестве основного протокола — в частности, в той же Internet — маршрутизаторы используются и как магистральное оборудование, обеспечивающее стыковку различных каналов и протоколов связи. Маршрутизаторы могут быть выполнены как в виде автономных устройств, так и программными средствами на базе компьютеров и специальных коммуникационных адаптеров.

Второй широко используемый тип периферийного оборудования — шлюзы (gateways), реализующие взаимодействие приложений, работающих в разных типах сетей. В корпоративных сетях используются в основном шлюзы OSI, обеспечивающие взаимодействие локальных сетей с ресурсами X.25 и шлюзы SNA, обеспечивающие подключение к сетям IBM. Полнофункциональный шлюз всегда представляет собой программно-аппаратный комплекс, поскольку должен обеспечивать необходимые для приложений программные интерфейсы. Маршрутизаторы Cisco Systems Среди маршрутизаторов наиболее, пожалуй, известны продукты компании Cisco Systems, реализующие широкий набор средств и протоколов, используемых при взаимодействии локальных сетей. Оборудование Cisco поддерживает разнообразные способы подключения, в том числе X.25, Frame Relay и ISDN, позволяя создавать достаточно сложные системы. Кроме того, среди семейства маршрутизаторов Cisco существуют прекрасные серверы удаленного доступа к локальным сетям, а в некоторых конфигурациях частично реализованы функции шлюзов (то, что в терминах Cisco называется Protocol Translation).

Основная область применения маршрутизаторов Cisco — сложные сети, использующие в качестве основного протокола IP или, реже, IPX. В частности, оборудование Cisco широко используется в опорных узлах Internet. Если ваша корпоративная сеть предназначена прежде всего для объединения удаленных LAN и требует сложной маршрутизации IP или IPX через разнородные каналы связи и сети передачи данных, то использование оборудования Cisco будет, скорее всего, оптимальным выбором. Средства же работы с Frame Relay и X.25 реализованы в маршрутизаторах Cisco только в том объеме, который нужен для объединения локальных сетей и доступа к ним. Если вы хотите строить свою систему на базе сетей с коммутацией пакетов, то маршрутизаторы Cisco могут работать в ней только как чисто периферийное оборудование, причем многие из функций маршрутизации оказываются при этом излишними, а цена, соответственно, слишком высокой. Наиболее интересными для использования в корпоративных сетях оказываются серверы доступа Cisco 2509, Cisco 2511 и новые устройства серии Cisco 2520. Основная область их применения — доступ удаленных пользователей к локальным сетям по телефонным линиям или ISDN с динамическим назначением IP-адресов (DHCP). Оборудование Motorola ISG Среди оборудования, предназначенного для работы с X.25 и Frame Relay, наибольший интерес представляют продукты, производимые группой информационных систем корпорации Motorola (Motorola ISG). В отличие от магистральных устройств, используемых в глобальных сетях передачи данных (Northern Telecom, Sprint, Alcatel и др.), оборудование Motorola способно работать полностью автономно, без специального центра управления сетью. Набор же возможностей, важных для использования в корпоративных сетях, у оборудования Motorola гораздо шире. Особо следует отметить развитые средства аппаратной и программной модернизации, позволяющие легко приспособлять оборудование к конкретным условиям. Все продукты Motorola ISG могут работать как коммутаторы X.25/Frame Relay, многопротокольные устройства доступа (PAD, FRAD, SLIP, PPP и пр.), поддерживают Annex G (X.25 поверх Frame Relay), обеспечивают преобразование протоколов SNA (SDLC/QLLC/RFC1490). Оборудование Motorola ISG можно разделить на три группы, отличающиеся набором аппаратных средств и областью применения.

Первую группу, предназначенную для работы в качестве периферийных устройств, составляет серия Vanguard. В нее входят узлы последовательного доступа Vanguard 100 (2-3 порта) и Vanguard 200 (6 портов), а также маршрутизаторы Vanguard 300/305 (1-3 последовательных порта и порт Ethernet/Token Ring) и ISDN-маршрутизаторы Vanguard 310. Маршрутизаторы Vanguard, кроме набора коммуникационных возможностей, включают передачу протоколов IP, IPX и Appletalk через X.25, Frame Relay и PPP. Естественно, при этом поддержан необходимый для всякого современного

маршрутизатора джентельменский набор — протоколы RIP и OSPF, средства фильтрации и ограничения доступа, компрессия данных и т.д.

Следующая группа продуктов Motorola ISG включает устройства Multimedia Peripheral Router (MPRouter) 6520 и 6560, отличающиеся в основном производительностью и возможностями расширения. В базовой конфигурации 6520 и 6560 имеют, соответственно, пять и три последовательных порта и порт Ethernet, причем у 6560 все порты высокоскоростные (до 2 Мбит/сек), а у 6520 три порта имеют скорость до 80 кбит/сек. MPRouter поддерживает все доступные для продуктов Motorola ISG коммуникационные протоколы и возможности маршрутизации. Основная черта MPRouter — возможность установки разнообразных дополнительных плат, что и отражает слово Multimedia в его названии. Существуют платы последовательных портов, портов Ethernet/Token Ring, платы ISDN, Ethernet hub. Самая интересная функция MPRouter — передача голоса по Frame Relay. Для этого в него устанавливаются специальные платы, допускающие подключение обычных телефонных или факс-аппаратов, а также аналоговых (E&M) и цифровых (E1, T1) АТС. Количество одновременно обслуживаемых голосовых каналов может достигать двух и более десятков. Таким образом, MPRouter может одновременно использоваться как средство интеграции голоса и данных, маршрутизатор и узел X.25/Frame Relay.

Третья группа продуктов Motorola ISG — магистральное оборудование глобальных сетей. Это расширяемые устройства семейства 6500plus, имеющие отказоустойчивое исполнение и средства резервирования и предназначенные для создания мощных узлов коммутации и доступа. Они включают различные наборы процессорных модулей и модулей ввода-вывода, позволяющие получить высокопроизводительные узлы, имеющие от 6 до 54 портов. В корпоративных сетях такие устройства могут использоваться для построения сложных систем с большим количеством подключаемых ресурсов.

Интересно провести сравнение маршрутизаторов Cisco и Motorola. Можно сказать, что для Cisco первична маршрутизация, а коммуникационные протоколы являются только средством связи, в то время как Motorola основное внимание уделяет коммуникационным возможностям, рассматривая маршрутизацию как еще одну реализуемую с помощью этих возможностей услугу. В целом средства маршрутизации продуктов Motorola беднее, чем у Cisco, однако вполне достаточны для подключения конечных узлов к Internet или корпоративной сети.

Производительность же изделий Motorola при прочих равных условиях, пожалуй, даже выше, причем при более низкой цене. Так Vanguard 300 при сравнимом наборе возможностей оказывается примерно в полтора раза дешевле, чем его ближайший аналог Cisco 2501.

#### Решения Eicon Technology

Во многих случаях в качестве периферийного оборудования корпоративных сетей удобно использовать решения канадской компании Eicon Technology. Основой решений Eicon является универсальный коммуникационный адаптер EiconCard, поддерживающий широкий набор протоколов — X.25, Frame Relay, SDLC, HDLC, PPP, ISDN. Этот адаптер устанавливается в один из компьютеров локальной сети, который становится коммуникационным сервером. Этот компьютер может использоваться и для других задач. Это возможно благодаря тому, что EiconCard имеет достаточно мощный процессор и собственную память и способна реализовать обработку сетевых протоколов не загружая коммуникационный сервер. Программные средства Eicon, позволяют строить на базе EiconCard как шлюзы, так и маршрутизаторы, работают под управлением практически всех операционных систем на платформе Intel. Здесь мы рассмотрим самые интересные из них.

Семейство решений Eicon для Unix включает маршрутизатор IP Connect, шлюзы X.25 Connect и SNA Connect. Все эти продукты могут быть установлены на компьютере, работающем под управлением SCO Unix или Unixware. IP Connect позволяет передавать



трафик IP через X.25, Frame Relay, PPP или HDLC и совместим с оборудованием других производителей, в частности Cisco и Motorola. В комплект поставки входит Firewall, средства компрессии данных и средства управления по SNMP. Основной областью применения IP Connect является подключение серверов приложений и Internet-серверов на базе Unix к сети передачи данных. Естественно, тот же компьютер может использоваться и как маршрутизатор для всего офиса, в котором он установлен. Использование маршрутизатора Eicon вместо «чисто аппаратных» устройств имеет ряд преимуществ. Во первых, это простота установки и использования. С точки зрения операционной системы EiconCard с установленным IP Connect выглядит как еще одна сетевая плата. Это делает настройку и администрирование IP Connect достаточно простым делом для всякого, кто общался с Unix. Во-вторых, непосредственное подключение сервера к сети передачи данных позволяет уменьшить загрузку офисной LAN и обеспечить ту самую единственную точку подключения к Internet или к корпоративной сети без установки дополнительных сетевых плат и маршрутизаторов. В третьих, такое «сервер-ориентированное» решение является более гибким и расширяемым, чем традиционные маршрутизаторы. Есть и ряд других преимуществ, появляющихся при совместном использовании IP Connect с другими продуктами Eicon.

X.25 Connect является шлюзом, обеспечивающим взаимодействие приложений локальной сети с ресурсами X.25. Этот продукт позволяет осуществить подключение пользователей Unix и рабочих станций DOS/Windows и OS/2 к удаленным системам электронной почты, базам данным и другим системам. Надо, кстати, отметить, что шлюзы Eicon на сегодня, пожалуй, единственный распространенный на нашем рынке продукт, реализующий стек OSI и позволяющий подключаться к приложениям X.400 и FTAM. Кроме того, X.25 Connect позволяет подключить удаленных пользователей к Unix-машине и терминальным приложениям на станциях локальной сети, а также организовать взаимодействие удаленных Unix-компьютеров через X.25. Используя вместе с X.25 Connect стандартные возможности Unix, можно реализовать преобразование протоколов, т.е. трансляцию доступа к Unix через Telnet в вызов X.25 и наоборот. Возможно подключение удаленного пользователя X.25, использующего SLIP или PPP к локальной сети и, соответственно, к Internet. В принципе, аналогичные возможности трансляции протоколов доступны в маршрутизаторах Cisco с программным обеспечением IOS Enterprise, однако такое решение оказывается дороже, чем продукты Eicon и Unix, вместе взятые.

Еще один упомянутый выше продукт — SNA Connect. Это шлюз, предназначенный для подключения к IBM mainframe и AS/400. Как правило, он используется вместе с программным обеспечением пользователя — эмуляторами терминалов 5250 и 3270 и интерфейсами APPC — также производимыми Eicon. Аналоги рассмотренных выше решений существуют и для других операционных систем — Netware, OS/2, Windows NT и даже DOS. Особо стоит упомянуть Interconnect Server for Netware, объединяющий все перечисленные возможности со средствами удаленной настройки и администрирования и системой авторизации клиентов. Он включает два продукта — Interconnect Router, позволяющий маршрутизировать IP, IPX и Appletalk и являющийся, с нашей точки зрения, самым удачным решением для объединения удаленных сетей Novell Netware, и Interconnect Gateway, обеспечивающий, в частности, мощные средства подключения к SNA. Еще один продукт Eicon, предназначенный для работы в среде Novell Netware — WAN Services for Netware. Это набор средств, позволяющих использовать приложения Netware в сетях X.25 и ISDN. Использование его вместе с Netware Connect дает возможность удаленным пользователям подключиться к локальной сети через X.25 или ISDN, а также обеспечить выход из локальной сети в X.25. Существует вариант поставки WAN Services for Netware вместе с Multiprotocol Router 3.0 компании Novell. Этот продукт называется Packet Blaster Advantage. Доступен также Packet Blaster ISDN, работающий не с EiconCard, а с ISDN-адаптерами, также поставляемыми Eicon. При этом возможны

различные варианты подключения — BRI (2B+D), 4BRI (8B+D) и PRI (30B+D). Для работы с приложениями Windows NT предназначен продукт WAN Services for NT. Он включает IP Router, средства подключения приложений NT к сетям X.25, поддержку для Microsoft SNA Server и средства доступа удаленных пользователей через X.25 в локальную сеть с помощью Remote Access Server. Для подключения сервера Windows NT к сети ISDN может использоваться также ISDN-адаптер Eicon вместе с программным обеспечением ISDN Services for Netware.

#### 4. Пример построения корпоративной сети

Теперь, перечислив и сравнив основные технологии, которые может задействовать разработчик, давайте перейдем к базовым вопросам и методам, используемым при проектировании и разработке сети.

Требования к сети.

Специалисты, занимающиеся разработкой вычислительных сетей, и сетевые администраторы всегда стремятся обеспечить выполнение трех основных требований, предъявляемых к сети, а именно:

- масштабируемость;
- производительность;
- управляемость.

Хорошая масштабируемость необходима для того, чтобы без особых усилий можно было менять как число пользователей, работающих в сети, так и прикладное программное обеспечение. Высокая производительность сети требуется для нормальной работы большинства современных приложений. И, наконец, сеть должна быть достаточно легко управляемой, чтобы ее можно было перенастраивать для удовлетворения постоянно меняющихся потребностей организации. Эти требования отражают новый этап в развитии сетевых технологий — этап создания высокопроизводительных корпоративных сетей.

Уникальность новых программных средств и технологий усложняет разработку корпоративных сетей. Централизованные ресурсы, новые классы программ, иные принципы их применения, изменение количественных и качественных характеристик информационного потока, увеличение числа одновременно работающих пользователей и повышение мощности вычислительных платформ — все эти факторы необходимо учитывать в их совокупности при разработке сети. Сейчас на рынке имеется большое количество технологических и архитектурных решений, и выбрать из них наиболее подходящее — достаточно сложная задача.

В современных условиях для правильного проектирования сети, ее разработки и обслуживания специалисты должны учитывать следующие вопросы:

- о Изменение организационной структуры.

При реализации проекта не следует «разлучать» специалистов по программному обеспечению и сетевых специалистов. При разработке сетей и всей системы в целом нужна единая команда из специалистов разного профиля;

- о Использование новых программных средств.

Необходимо знакомиться с новым программным обеспечением еще на ранней стадии разработки сети для того, чтобы можно было своевременно внести необходимые коррективы в планирующиеся к использованию средства;

- о Исследование различных решений.

Необходимо оценивать различные архитектурные решения и их возможное влияние на работу будущей сети;

- о Проверка сетей.

Необходимо проводить тестирование всей сети или ее частей на ранних стадиях разработки. Для этого можно создать прототип сети, который позволит оценить правильность принятых решений. Так можно предупредить появление разного рода "узких мест" и определить применимость и примерную производительность разных архитектур;

о Выбор протоколов.

Чтобы правильно выбрать конфигурацию сети, нужно оценить возможности различных протоколов. Важно определить, как сетевые операции, оптимизирующие работу одной программы или пакета программ, могут повлиять на производительность других;

о Выбор физического расположения.

Выбирая место установки серверов, надо, прежде всего, определить местоположение пользователей. Возможно ли их перемещение? Будут ли их компьютеры подключены к одной подсети? Будут ли пользователи иметь доступ к глобальной сети?

о Вычисление критического времени.

Необходимо определить время допустимой реакции каждого приложения и возможные периоды максимальной нагрузки. Важно понять, как нештатные ситуации могут повлиять на работоспособность сети, и определить, нужен ли резерв для организации непрерывной работы предприятия;

о Анализ вариантов.

Важно проанализировать различные варианты использования программного обеспечения в сети. Централизованное хранение и обработка информации часто создают дополнительную нагрузку в центре сети, а распределенные вычисления могут потребовать усиления локальных сетей рабочих групп.

На сегодня нет готовой, отлаженной универсальной методики, следуя которой, можно автоматически провести весь комплекс мероприятий по разработке и созданию корпоративной сети. В первую очередь это связано с тем, что не существует двух абсолютно одинаковых организаций. В частности, каждая организация характеризуется уникальным стилем руководства, иерархией, культурой ведения дел. А если учесть, что сеть неизбежно отражает структуру организации, то можно смело сказать — двух одинаковых сетей не существует.

Архитектура сети

До того как начинать построение корпоративной сети, необходимо сначала определить ее архитектуру, функциональную и логическую организацию и учесть существующую телекоммуникационную инфраструктуру. Тщательно проработанная архитектура сети помогает оценить возможность применения новых технологий и прикладных программ, служит заделом для будущего роста, определяет выбор сетевых технологий, помогает избежать избыточных затрат, отражает связь сетевых компонентов, значительно снижает риск неправильной реализации и т.д. Архитектура сети закладывается в основу технического задания на создаваемую сеть. Следует отметить, что архитектура сети отличается от проекта сети тем, что она, например, не определяет точную принципиальную схему сети и не регламентирует размещение сетевых компонентов. Архитектура сети, например, определяет, будут ли некоторые части сети построены на базе Frame Relay, ATM, ISDN или других технологий. Сетевой проект должен содержать конкретные указания и оценки параметров, например, требуемое значение пропускной способности, реальную ширину полосы пропускания, точное расположение каналов связи и т.д.

В архитектуре сети выделяют три аспекта, три логические составляющие:

принципы построения,  
сетевые шаблоны  
и технические позиции.

Принципы построения используются при планировании сети и принятии решений. Принципы — это набор простых инструкций, которые с достаточной степенью детализации описывают все вопросы построения и эксплуатации развертываемой сети в течение длительного периода времени. Как правило, в основе формирования принципов лежат корпоративные цели и базовые методы ведения бизнеса организации.

Принципы обеспечивают первичную связь между корпоративной стратегией развития и сетевыми технологиями. Они служат для разработки технических позиций и сетевых шаблонов. При разработке технического задания на сеть принципы построения сетевой архитектуры излагаются в разделе, определяющем общие цели сети. Техническая позиция может рассматриваться в качестве целевого описания, определяющего выбор между конкурирующими альтернативными сетевыми технологиями. Техническая позиция уточняет параметры выбранной технологии и дает описание отдельно взятого устройства, метода, протокола, предоставляемого сервиса и т.д. Например, при выборе технологии локальной сети необходимо принимать во внимание скорость, стоимость, качество обслуживания и другие требования. Разработка технических позиций требует глубокого знания сетевых технологий и внимательного рассмотрения требований организации. Количество технических позиций определяется заданной степенью детализации, сложностью сети и масштабами организации. Архитектура сети может быть описана следующими техническими позициями:

Сетевые транспортные протоколы.

Какие транспортные протоколы должны использоваться для передачи информации?

Маршрутизация в сети.

Какой протокол маршрутизации должен использоваться между маршрутизаторами и коммутаторами АТМ?

Качество обслуживания.

За счет чего будет достигаться возможность выбора качества сервиса?

Адресация в сетях IP и домены адресации.

Какая адресная схема должна использоваться для сети, включая зарегистрированные адреса, подсети, маски подсети, переадресацию и т.д.?

Коммутация в локальных сетях.

Какая стратегия коммутации должна быть использована в локальных сетях?

Объединение коммутации и маршрутизации.

Где и как должны использоваться коммутация и маршрутизация; как они должны объединяться?

Организация городской сети.

Каким образом должны связываться отделения предприятия, находящиеся, скажем, в одном городе?

Организация глобальной сети.

Каким образом отделения предприятия должны связываться по глобальной сети?

Служба удаленного доступа.

Как пользователи удаленных отделений получают доступ к сети предприятия?

Сетевые шаблоны — это набор моделей сетевых структур, отражающих связь между компонентами сети. Например, для определенной архитектуры сети создается набор шаблонов, чтобы «проявить» топологию сети крупного отделения или глобальной сети, или показать распределение протоколов по уровням. Сетевые шаблоны иллюстрируют сетевую инфраструктуру, которая описывается полным набором технических позиций. Более того, в хорошо продуманной сетевой архитектуре сетевые шаблоны по степени детализации могут максимально приближаться по своему содержанию к техническим позициям. По сути дела, сетевые шаблоны — это описание функциональной схемы участка сети, имеющего конкретные границы, можно выделить следующие основные сетевые шаблоны: для глобальной сети, для городской сети, для центрального офиса, для крупного отделения организации, для отделения. Могут быть разработаны и другие шаблоны для участков сети, имеющих какие-либо особенности.

Описываемый методологический подход основан на изучении конкретной ситуации, рассмотрении принципов построения корпоративной сети в их совокупности, анализе ее функциональной и логической структуры, выработке набора сетевых шаблонов

и технических позиций. Различные реализации корпоративных сетей могут включать в свой состав те или иные компоненты. В общем случае корпоративная сеть состоит из различных отделений, объединенных сетями связи. Они могут быть глобальными (WAN) или городскими (MAN). Отделения могут быть крупными, средними и малыми. Крупное отделение может быть центром обработки и хранения информации. Выделяется центральный офис, из которого производится управление всей корпорацией. К малым отделениям можно отнести различные обслуживающие подразделения (склады, мастерские и т.д.). Малые отделения по сути являются удаленными. Стратегическое назначение удаленного отделения — разместить службы сбыта и технической поддержки поближе к потребителю. Связь с клиентами, которая в значительной мере влияет на доходы корпорации, будет более продуктивной, если все сотрудники получат возможность доступа к корпоративным данным в любой момент времени.

На первом шаге построения корпоративной сети описывается предполагаемая функциональная структура. Определяется количественный состав и статус офисов и отделений. Обосновывается необходимость развертывания собственной частной сети связи или производится выбор провайдера услуг, который способен удовлетворить предъявляемые требования. Разработка функциональной структуры производится с учетом финансовых возможностей организации, перспективных планов развития, числа активных пользователей сети, работающих приложений, необходимого качества обслуживания. В основе разработки лежит функциональная структура самого предприятия.

На втором шаге определяется логическая структура корпоративной сети. Логические структуры отличаются друг от друга только выбором технологии (ATM, Frame Relay, Ethernet ...) для построения магистрали, которая является центральным звеном сети корпорации. Рассмотрим логические структуры, построенные на базе коммутации ячеек и коммутации кадров. Выбор между этими двумя способами передачи информации осуществляется, исходя из необходимости предоставления гарантированного качества обслуживания. Могут быть использованы и другие критерии.

Магистраль передачи данных должна удовлетворять двум основным требованиям.

Возможность подключения большого количества низкоскоростных рабочих станций к небольшому количеству мощных, высокоскоростных серверов.

Приемлемая скорость отклика на запросы клиентов.

Идеальная магистраль должна обладать высокой надежностью передачи данных и развитой системой управления. Под системой управления следует понимать, например, возможность конфигурирования магистрали с учетом всех местных особенностей и поддержку надежности на таком уровне, что, даже если некоторые части сети выйдут из строя, серверы остаются доступными. Перечисленные требования определяют, вероятно, несколько технологий и окончательный выбор одной из них остается за самой организацией. Необходимо решить, что важнее всего — стоимость, скорость, масштабируемость или качество обслуживания.

Логическая структура с коммутацией ячеек применяется в сетях с мультимедийным трафиком в реальном масштабе времени (проведение видеоконференций и качественная передача голоса). При этом важно трезво оценить, насколько необходима такая дорогостоящая сеть (с другой стороны, даже дорогие сети подчас не способны удовлетворить некоторые требования). Если это так, то необходимо брать за основу логическую структуру сети с коммутацией кадров. Логическую иерархию коммутации, объединяющую два уровня модели OSI, можно представить в виде трехуровневой схемы:

Нижний уровень служит для объединения локальных сетей Ethernet,

Средний уровень представляет собой либо локальную сеть ATM, либо сеть MAN, либо магистральную сеть связи WAN.

Верхний уровень данной иерархической структуры отвечает за маршрутизацию.

Логическая структура позволяет выявить все возможные маршруты связи между отдельными участками корпоративной сети

Магистраль на базе коммутации ячеек

При использовании для построения магистрали сети технологии коммутации ячеек объединение всех коммутаторов Ethernet уровня рабочих групп осуществляют высокопроизводительные коммутаторы ATM. Работая на втором уровне эталонной модели OSI, эти коммутаторы передают 53-байтовые ячейки фиксированной длины вместо кадров Ethernet переменной длины. Такая концепция построения сети подразумевает, что коммутатор Ethernet уровня рабочей группы должен иметь выходной порт ATM с функцией сегментации и сборки (SAR), который преобразовывает кадры Ethernet переменной длины в ячейки ATM фиксированной длины перед передачей информации на магистральный коммутатор ATM.

Для глобальных сетей базовые коммутаторы ATM способны обеспечить связь отдаленных регионов. Также работая на втором уровне модели OSI, эти коммутаторы в сети WAN могут использовать каналы T1/E1 (1.544/2.0Мбит/с), канал T3 (45Мбит/с) или канал OC-3 технологии SONET (155Мбит/с). Для обеспечения городской связи можно развернуть сеть MAN с использованием технологии ATM. Та же самая магистральная сеть ATM может использоваться для связи между собой телефонных станций. В будущем в рамках телефонной модели клиент/сервер эти станции могут быть заменены голосовыми серверами в локальной сети. В этом случае возможность гарантирования качества обслуживания в сетях ATM становится очень важной при организации связи с клиентскими персональными компьютерами.

Маршрутизация

Как уже было отмечено, маршрутизация — это третий и самый высокий уровень в иерархической структуре сети. Маршрутизация, которая работает на третьем уровне эталонной модели OSI, используется для организации сеансов связи, к которым относятся:

Сеансы связи между устройствами, расположенными в различных виртуальных сетях (при этом каждая сеть является обычно отдельной IP-подсетью);

Сеансы связи, которые проходят через глобальные/городские

Одна из стратегий построения корпоративной сети состоит в установке коммутаторов на нижних уровнях общей сети. Затем локальные сети связываются с помощью маршрутизаторов. Маршрутизаторы требуются для того, чтобы разделить IP-сеть большой организации на множество отдельных IP-подсетей. Это необходимо для предотвращения «широковещательного взрыва», связанного с работой таких протоколов, как ARP. Чтобы сдерживать распространение нежелательного трафика по сети, все рабочие станции и серверы необходимо разбить на виртуальные сети. В этом случае маршрутизация управляет взаимодействием между устройствами, принадлежащими к различным виртуальным локальным сетям.

Такая сеть состоит из маршрутизаторов или серверов маршрутизации (логическое ядро), магистрали сети на базе коммутаторов ATM и большого количества коммутаторов Ethernet, расположенных на периферии. За исключением особых случаев, например, использования видеосерверов, которые подключаются непосредственно к магистрали ATM, все рабочие места и серверы должны подключаться к коммутаторам Ethernet. Такое построение сети позволит локализовать внутренний трафик внутри рабочих групп и предотвратить перекачку такого трафика через магистральные коммутаторы ATM или маршрутизаторы. Объединение коммутаторов Ethernet осуществляют коммутаторы ATM, обычно расположенные в том же самом отделении. Следует отметить, что может потребоваться несколько коммутаторов ATM, чтобы обеспечить достаточное количество портов для подключения всех коммутаторов Ethernet. Как правило, в этом случае используется связь на 155Мбит/с по многомодовому оптоволоконному кабелю.

Маршрутизаторы располагаются в стороне от магистральных коммутаторов ATM, так как эти маршрутизаторы необходимо вынести за маршруты основных сеансов связи.

Такое построение делает маршрутизацию необязательной. Это зависит от типа сеанса связи и от вида трафика в сети. Маршрутизации нужно стараться избегать при передаче видеoinформации в реальном времени, так как она может вносить нежелательные задержки. Маршрутизация не нужна для связи между устройствами, расположенными в одной виртуальной сети, даже если они находятся в различных зданиях на территории большого предприятия.

Кроме того, даже в ситуации, когда маршрутизаторы требуются для проведения определенных сеансов связи, размещение маршрутизаторов в стороне от магистральных коммутаторов АТМ позволяет минимизировать число переходов маршрутизации (под переходом маршрутизации понимается участок сети от пользователя до первого маршрутизатора или от одного маршрутизатора до другого). Это позволяет не только снизить задержку, но и уменьшить нагрузку на маршрутизаторы. Маршрутизация получила широкое распространение как технология связи локальных сетей в глобальной среде. Маршрутизаторы предоставляют разнообразные услуги, рассчитанные на многоуровневый контроль канала передачи. Сюда относятся общая схема адресации (на сетевом уровне), не зависящей от того, как формируются адреса предыдущего уровня, а также преобразование одного формата кадра контрольного уровня в другой.

Маршрутизаторы принимают решения о том, куда направлять поступающие пакеты данных, исходя из содержащейся в них информации об адресах сетевого уровня. Эта информация извлекается, анализируется и сопоставляется с содержимым таблиц маршрутизации, что позволяет определить, в какой порт следует отправить тот или иной пакет. Затем из адреса сетевого уровня вычленяется адрес канального уровня, если пакет следует передать в сегмент такой сети, как Ethernet или Token Ring.

Помимо обработки пакетов маршрутизаторы параллельно осуществляют обновление таблиц маршрутизации, которые используются для определения места назначения каждого пакета. Маршрутизаторы создают и ведут эти таблицы в динамическом режиме. В результате маршрутизаторы могут автоматически реагировать на изменение состояния сети, например, на возникновение перегрузки или повреждение каналов связи.

Определение маршрута — это достаточно сложная задача. В корпоративной сети коммутаторы АТМ должны функционировать примерно так же, как и маршрутизаторы: обмен информацией должен проходить с учетом топологии сети, доступных маршрутов и стоимости передачи. Коммутатору АТМ эта информация крайне необходима, чтобы выбрать наилучший маршрут для конкретного сеанса связи, инициируемого конечными пользователями. К тому же, определение маршрута не ограничивается одним лишь принятием решения о пути, по которому пройдет логическое соединение после формирования запроса на его создание.

Коммутатор АТМ может выбрать новые маршруты, если по каким-либо причинам каналы связи окажутся недоступными. При этом коммутаторы АТМ должны обеспечивать надежность сети на уровне маршрутизаторов. Чтобы создать расширяемую сеть с высокой экономической эффективностью, необходимо перенести функции маршрутизации на периферию сети и обеспечить коммутацию трафика в ее магистрали. АТМ является единственной сетевой технологией, которая способна сделать это.

Для выбора технологии необходимо ответить на следующие вопросы:

Обеспечивает ли технология адекватное качество обслуживания?

Может ли она гарантировать качество обслуживания?

Насколько расширяемой получится сеть?

Допускается ли выбор топологии сети?

Рентабельны ли услуги, предоставляемые сетью?

Насколько будет эффективна система управления?

Ответы на эти вопросы определяют выбор. Но, в принципе, на разных участках сети могут использоваться разные технологии. Например, если отдельные участки

требуют поддержки мультимедийного трафика в реальном времени или скорости в 45Мбит/с, то в них устанавливают АТМ. Если же участок сети требует диалоговой обработки запросов, что не допускает существенных задержек, то необходимо использовать Frame Relay, если такие услуги доступны в данной географической области (иначе придется прибегнуть к Internet).

Так, большое предприятие может соединяться с сетью через АТМ, в то время как филиалы связываются с той же самой сетью через Frame Relay.

При создании корпоративной сети и выборе сетевой технологии с соответствующим программным и аппаратным обеспечением следует учитывать соотношение цена/производительность. Трудно ожидать высоких скоростей от дешевых технологий. С другой стороны, бессмысленно использовать сложнейшие технологии для простейших задач. Следует правильно комбинировать различные технологии для достижения максимальной эффективности.

При выборе технологии следует учитывать тип кабельной системы и требуемые расстояния; совместимость с уже установленным оборудованием (значительной минимизации расходов можно достичь, если в новую систему удастся включить уже установленное оборудование).

Вообще говоря, можно выделить два пути построения высокоскоростной локальной сети: эволюционный и революционный.

Первый путь основан на расширении старой доброй технологии ретрансляции кадров. Увеличить быстродействие локальной сети в рамках такого подхода можно за счет модернизации сетевой инфраструктуры, добавления новых каналов связи и изменения способа передачи пакетов (что и сделано в коммутируемом Ethernet). Обычная сеть Ethernet совместно использует полосу пропускания, то есть трафики всех пользователей сети соперничают друг с другом, претендуя на всю пропускную способность сетевого сегмента. В коммутируемом Ethernet создаются выделенные маршруты, благодаря чему пользователям доступна реальная полоса пропускания в 10Мбит/с.

Революционный путь предполагает переход на кардинально новые технологии, например, АТМ для локальных сетей.

Богатая практика построения локальных сетей показала, что основным вопросом является качество обслуживания. Именно этим определяется, сможет ли сеть успешно работать (например с такими приложениями, как видеоконференции, которые находят все более широкое применение в мире).



## **1. 7 Лекция №7 (2 часа).**

**Тема:** «Глобальные компьютерные сети»

### **1.7.1 Вопросы лекции:**

1. Понятие и классификация глобальных компьютерных сетей
2. Выделенные каналы связи глобальных сетей
3. Глобальные сети с коммутацией каналов
4. Глобальные сети с коммутацией пакетов

### **1.7.2 Краткое содержание вопросов:**

1. Понятие и классификация глобальных компьютерных сетей

Назначение компьютерной сети.

Локальная компьютерная сеть - это совокупность компьютеров, соединенных линиями связи, обеспечивающая пользователям сети потенциальную возможность совместного использования ресурсов всех компьютеров. С другой стороны, проще говоря, компьютерная сеть - это совокупность компьютеров и различных устройств, обеспечивающих информационный обмен между компьютерами в сети без использования каких-либо промежуточных носителей информации.

Основное назначение компьютерных сетей - совместное использование ресурсов и осуществление интерактивной связи как внутри одной фирмы, так и за ее пределами. Ресурсы (resources) - это данные, приложения и периферийные устройства, такие, как внешний дисковод, принтер, мышь, модем или джойстик.

Компьютеры, входящие в сеть выполняют следующие функции:

- организацию доступа к сети
- управление передачей информации
- предоставление вычислительных ресурсов и услуг пользователям сети.

Рождение компьютерных сетей было вызвано практической потребностью - иметь возможность для совместного использования данных. Персональный компьютер - прекрасный инструмент для создания документа, подготовки таблиц, графических данных и других видов информации, но при этом нет возможности быстро поделиться своей информацией с другими.

В настоящее время локальные вычислительные (ЛВС) получили очень широкое распространение. Это вызвано несколькими причинами:

объединение компьютеров в сеть позволяет значительно экономить денежные средства за счет уменьшения затрат на содержание компьютеров (достаточно иметь определенное дисковое пространство на файл-сервере (главном компьютере сети) с установленными на нем программными продуктами, используемыми несколькими рабочими станциями);

локальные сети позволяют использовать почтовый ящик для передачи сообщений на другие компьютеры, что позволяет в наиболее короткий срок передавать документы с одного компьютера на другой;

локальные сети, при наличии специального программного обеспечения (ПО), служат для организации совместного использования файлов (к примеру, бухгалтеры на нескольких машинах могут обрабатывать проводки одной и той же бухгалтерской книги).

Кроме всего прочего, в некоторых сферах деятельности просто невозможно обойтись без ЛВС. К таким сферам относятся: банковское дело, складские операции крупных компаний, электронные архивы библиотек и др. В этих сферах каждая отдельно взятая рабочая станция в принципе не может хранить всей информации (в основном, по причине слишком большого ее объема).

Глобальная вычислительная сеть - сеть, соединяющая компьютеры, удаленные географически на большие расстояния друг от друга. Отличается от локальной сети более протяженными коммуникациями (спутниковыми, кабельными и др.). Глобальная сеть объединяет локальные сети.

Internet - глобальная компьютерная сеть, охватывающая весь мир.

Фактически Internet состоит из множества локальных и глобальных сетей, принадлежащих различным компаниям и предприятиям, связанных между собой различными линиями связи. Internet можно представить себе в виде мозаики сложенной из небольших сетей разной величины, которые активно взаимодействуют одна с другой, пересылая файлы, сообщения и т.п.

Глобальная сеть Internet, служившая когда-то исключительно исследовательским и учебным группам, чьи интересы простирались вплоть до доступа к суперкомпьютерам, становится все более популярной в деловом мире.

Компании соблазняют быстрота, дешевая глобальная связь, удобство для проведения совместных работ, доступные программы, уникальная база данных сети Internet. Они рассматривают глобальную сеть как дополнение к своим собственным локальным сетям.

#### Классификация компьютерных сетей

По способу организации сети подразделяются на реальные и искусственные.

Искусственные сети (псевдосети) позволяют связывать компьютеры вместе через последовательные или параллельные порты и не нуждаются в дополнительных устройствах. Иногда связь в такой сети называют связью по нуль-модему (не используется модем). Само соединение называют нуль-модемным. Искусственные сети используются, когда необходимо перекачать информацию с одного компьютера на другой. MS-DOS и windows снабжены специальными программами для реализации нуль-модемного соединения.

Реальные сети позволяют связывать компьютеры с помощью специальных устройств коммутации и физической среда передачи данных.

По территориальной распространенности сети могут быть локальными, глобальными, региональными и городскими.

Локальная вычислительная сеть (ЛВС) -Local Area Networks (LAN) - это группа (коммуникационная система) относительно небольшого количества компьютеров, объединенных совместно используемой средой передачи данных, расположенных на ограниченной по размерам небольшой площади в пределах одного или нескольких близко находящихся зданий (обычно в радиусе не более 1-2 км) с целью совместного использования ресурсов всех компьютеров

Глобальная вычислительная сеть (ГВС или WAN - World Area NetWork) - сеть, соединяющая компьютеры, удалённые географически на большие расстояния друг от друга. Отличается от локальной сети более протяженными коммуникациями (спутниковыми, кабельными и др.). Глобальная сеть объединяет локальные сети.

Городская сеть (MAN - Metropolitan Area NetWork) - сеть, которая обслуживает информационные потребности большого города.

Региональные - расположенные на территории города или области.

Так же, в последнее время специалисты выделяют такой вид сети, как банковская, которая представляет собой частный случай корпоративной сети крупной компании. Очевидно, что специфика банковской деятельности предъявляет жесткие требования к системам защиты информации в компьютерных сетях банка. Не менее важную роль при построении корпоративной сети играет необходимость обеспечения безотказной и бесперебойной работы, поскольку даже кратковременный сбой в ее работе может привести к гигантским убыткам.

По принадлежности различают ведомственные и государственные сети. Ведомственные принадлежат одной организации и располагаются на ее территории.

Государственные сети - сети, используемые в государственных структурах.

По скорости передачи информации компьютерные сети делятся на низко-, средне- и высокоскоростные.

низкоскоростные (до 10 Мбит/с),

среднескоростные (до 100 Мбит/с),

высокоскоростные (свыше 100 Мбит/с);

В зависимости от назначения и технических решений сети могут иметь различные конфигурации (или, как еще говорят, архитектуру, или топологию).

В кольцевой топологии информация передается по замкнутому каналу. Каждый абонент непосредственно связан с двумя ближайшими соседями, хотя в принципе способен связаться с любым абонентом сети.

В звездообразной (радиальной) в центре находится центральный управляющий компьютер, последовательно связывающийся с абонентами и связывающий их друг с другом.

В шинной конфигурации компьютеры подключены к общему для них каналу (шине), через который могут обмениваться сообщениями.

В древовидной - существует «главный» компьютер, которому подчинены компьютеры следующего уровня, и т.д.

Кроме того, возможны конфигурации без отчетливого характера связей; пределом является полносвязная конфигурация, когда каждый компьютер в сети непосредственно связан с любым другим компьютером.

С точки зрения организации взаимодействия компьютеров, сети делят на одноранговые (Peer-to-Peer Network) и с выделенным сервером (Dedicated Server Network).

Все компьютеры одноранговой сети равноправны. Любой пользователь сети может получить доступ к данным, хранящимся на любом компьютере.

Одноранговые сети могут быть организованы с помощью таких операционных систем, как LANtastic, windows'3.11, Novell Netware Lite. Указанные программы работают как с DOS, так и с Windows. Одноранговые сети могут быть организованы также на базе всех современных 32-разрядных операционных систем - Windows 9xME2k, Windows NT workstation версии, OS/2) и некоторых других.

Достоинства одноранговых сетей:

- 1) наиболее просты в установке и эксплуатации.
- 2) операционные системы DOS и Windows обладают всеми необходимыми функциями, позволяющими строить одноранговую сеть.

Недостаток одноранговых сетей в том, что затруднено решение вопросов защиты информации. Поэтому такой способ организации сети используется для сетей с небольшим количеством компьютеров и там, где вопрос защиты данных не является принципиальным.

В иерархической сети при установке сети заранее выделяются один или несколько компьютеров, управляющих обменом данных по сети и распределением ресурсов. Такой компьютер называют сервером.

Любой компьютер, имеющий доступ к услугам сервера называют клиентом сети или рабочей станцией.

Сервер в иерархических сетях - это постоянное хранилище разделяемых ресурсов. Сам сервер может быть клиентом только сервера более высокого уровня иерархии. Поэтому иерархические сети иногда называются сетями с выделенным сервером.

Серверы обычно представляют собой высокопроизводительные компьютеры, возможно, с несколькими параллельно работающими процессорами, с винчестерами большой емкости, с высокоскоростной сетевой картой (100 Мбит/с и более).

Иерархическая модель сети является наиболее предпочтительной, так как позволяет создать наиболее устойчивую структуру сети и более рационально распределить ресурсы.

Также достоинством иерархической сети является более высокий уровень защиты данных.

К недостаткам иерархической сети, по сравнению с одноранговыми сетями, относятся:

- 1) необходимость дополнительной ОС для сервера.

- 2) более высокая сложность установки и модернизации сети.
- 3) Необходимость выделения отдельного компьютера в качестве сервера.

## 2. Выделенные каналы связи глобальных сетей

Выделенные (или арендуемые - leased) каналы можно получить у телекоммуникационных компаний, которые владеют каналами дальней связи (таких, например, как «РОСТЕЛЕКОМ»), или от телефонных компаний, которые обычно сдают в аренду каналы в пределах города или региона.

Использовать выделенные линии можно двумя способами. Первый состоит в построении с их помощью территориальной сети определенной технологии, например frame relay, в которой арендуемые выделенные линии служат для соединения промежуточных, территориально распределенных коммутаторов пакетов.

Второй вариант - соединение выделенными линиями только объединяемых локальных сетей или конечных абонентов другого типа, например мэйнфреймов, без установки транзитных коммутаторов пакетов, работающих по технологии глобальной сети. Второй вариант является наиболее простым с технической точки зрения, так как основан на использовании маршрутизаторов или удаленных мостов в объединяемых локальных сетях и отсутствии протоколов глобальных технологий, таких как X.25 или frame relay. По глобальным каналам передаются те же пакеты сетевого или канального уровня, что и в локальных сетях.

Именно второй способ использования глобальных каналов получил специальное название «услуги выделенных каналов», так как в нем действительно больше ничего из технологий собственно глобальных сетей с коммутацией пакетов не используется.

Выделенные каналы очень активно применялись совсем в недалеком прошлом и применяются сегодня, особенно при построении ответственных магистральных связей между крупными локальными сетями, так как эта услуга гарантирует пропускную способность арендуемого канала. Однако при большом количестве географически удаленных точек и интенсивном смешанном трафике между ними использование этой службы приводит к высоким затратам за счет большого количества арендуемых каналов.

Сегодня существует большой выбор выделенных каналов - от аналоговых каналов тональной частоты с полосой пропускания 3,1 кГц до цифровых каналов технологии SDH с пропускной способностью 155 и 622 Мбит/с.

## 3. Глобальные сети с коммутацией каналов

Сегодня для построения глобальных связей в корпоративной сети доступны сети с коммутацией каналов двух типов - традиционные аналоговые телефонные сети и цифровые сети с интеграцией услуг ISDN. Достоинством сетей с коммутацией каналов является их распространенность, что характерно особенно для аналоговых телефонных сетей. В последнее время сети ISDN во многих странах также стали вполне доступны корпоративному пользователю, а в России это утверждение относится пока только к крупным городам.

Известным недостатком аналоговых телефонных сетей является низкое качество составного канала, которое объясняется использованием телефонных коммутаторов устаревших моделей, работающих по принципу частотного уплотнения каналов (FDM-технологии). На такие коммутаторы сильно воздействуют внешние помехи (например, грозовые разряды или работающие электродвигатели), которые трудно отличить от полезного сигнала. Правда, в аналоговых телефонных сетях все чаще используются цифровые АТС, которые между собой передают голос в цифровой форме. Аналоговым в таких сетях остается только абонентское окончание. Чем больше цифровых АТС в телефонной сети, тем выше качество канала, однако до полного вытеснения АТС, работающих по принципу FDM-коммутации, в нашей стране еще далеко. Кроме качества каналов, аналоговые телефонные сети также обладают таким недостатком, как большое время установления соединения, особенно при импульсном способе набора номера, характерного для нашей страны.

Телефонные сети, полностью построенные на цифровых коммутаторах, и сети ISDN свободны от многих недостатков традиционных аналоговых телефонных сетей. Они предоставляют пользователям высококачественные линии связи, а время установления соединения в сетях ISDN существенно сокращено.

Однако даже при качественных каналах связи, которые могут обеспечить сети с коммутацией каналов, для построения корпоративных глобальных связей эти сети могут оказаться экономически неэффективными. Так как в таких сетях пользователи платят не за объем переданного трафика, а за время соединения, то при трафике с большими пульсациями и, соответственно, большими паузами между пакетами оплата идет во многом не за передачу, а за ее отсутствие. Это прямое следствие плохой приспособленности метода коммутации каналов для соединения компьютеров.

Тем не менее при подключении массовых абонентов к корпоративной сети, например сотрудников предприятия, работающих дома, телефонная сеть оказывается единственным подходящим видом глобальной службы из соображений доступности и стоимости (при небольшом времени связи удаленного сотрудника с корпоративной сетью).

#### 4. Глобальные сети с коммутацией пакетов

В 80-е годы для надежного объединения локальных сетей и крупных компьютеров в корпоративную сеть использовалась практически одна технология глобальных сетей с коммутацией пакетов - X.25. Сегодня выбор стал гораздо шире, помимо сетей X.25 он включает такие технологии, как frame relay, SMDS и ATM. Кроме этих технологий, разработанных специально для глобальных компьютерных сетей, можно воспользоваться услугами территориальных сетей TCP/IP, которые доступны сегодня как в виде недорогой и очень распространенной сети Internet, качество транспортных услуг которой пока практически не регламентируется и оставляет желать лучшего, так и в виде коммерческих глобальных сетей TCP/IP, изолированных от Internet и предоставляемых в аренду телекоммуникационными компаниями.

В графе «Трафик» указывается тип трафика, который наиболее подходит для данного типа сетей, а в графе «Скорость доступа» - наиболее типичный диапазон скоростей, предоставляемых поставщиками услуг этих сетей.

Принципы работы сетей TCP/IP уже были подробно рассмотрены. Эти принципы остаются неизменными и при включении в состав этих сетей глобальных сетей различных технологий. Для остальных технологий, кроме SMDS, будут рассмотрены принципы доставки пакетов, пользовательский интерфейс и типы оборудования доступа к сетям данных технологий.

Технология SMDS (Switched Multi-megabit Data Service) была разработана в США для объединения локальных сетей в масштабах мегаполиса, а также предоставления высокоскоростного выхода в глобальные сети. Эта технология поддерживает скорости доступа до 45 Мбит/с и сегментирует кадры MAC - уровня в ячейки фиксированного размера 53 байт, имеющие, как и ячейки технологии ATM, поле данных в 48 байт. Технология SMDS основана на стандарте IEEE 802.6, который описывает несколько более широкий набор функций, чем SMDS. Стандарты SMDS приняты компанией Bellcore, но международного статуса не имеют. Сети SMDS были реализованы во многих крупных городах США, однако в других странах эта технология распространения не получила. Сегодня сети SMDS вытесняются сетями ATM, имеющими более широкие функциональные возможности, поэтому в данной книге технология SMDS подробно не рассматривается.

## 1. 8 Лекция №8 (2 часа).

**Тема:** «Понятие базы данных»

### 1.8.1 Вопросы лекции:

1. Базы данных. Понятие и классификация.
2. Виды моделей данных.
3. Понятие информационного объекта. Нормализация отношений. Типы связей информационных объектов.
4. Построение инфологической модели.

### 1.8.2 Краткое содержание вопросов:

1. Базы данных. Понятие и классификация.

Цель любой информационной системы - обработка данных об объектах реального мира. В широком смысле слова **база данных**- совокупность сведений о конкретных объектах реального мира в какой-либо предметной области. Создавая базу данных, пользователь стремится упорядочить информацию по различным признакам и быстро извлекать выборку с произвольным сочетанием признаков. Это возможно только если данные структурированы.

**Структурирование** - введение соглашений о способах представления данных. Структурированными (рисунок 1) называются данные, записанные, например, в текстовом файле. В неструктурированных данных сложно организовать поиск данных, а упорядочить подобную информацию почти невозможно. Чтобы автоматизировать поиск и систематизировать эти данные необходимо выработать определённые соглашения или правила о способах представления данных. Например, дату рождения студента нужно записать одинаково для каждого, она должна иметь одинаковую длину и определённое место среди остальной информации.

№ личного дела	Фам	Имя	Отч	Дата рожд
536	Ломакин	Максим	Евгеньевич	11.03.82

} Запись

{ Поле

Рисунок 1 - Пример структурированных данных

Пользователями баз данных могут быть различные прикладные программы, программные комплексы, а также специалисты предметной области, выступающие в роли потребителей или источников данных, называемые конечными пользователями.

**База данных**- поименованная совокупность структурированных данных, относящихся к определенной предметной области (например, личное дело студента).

1. По форме представления информации:

видеосистема;  
аудиосистема;  
система мультимедиа.

Эта классификация показывает, в каком виде информация хранится и выдается пользователям (в виде изображения, звука, движущегося изображения). "Изображение" используется в широком смысле: текст, графика, фото, карты, анимационные изображения.

2. По типу хранимой информации:

документальные;  
фактографические;  
лексикографические.

В **документальных БД** единицей хранения является какой либо документ и пользователю в ответ на запрос выдается либо сам документ либо ссылка на него. БД документального типа могут быть организованы по разному: без хранения и с хранением самого исходного документа на машинных носителях. Среди документальных БД различают библиографические, реферативные и полнотекстовые.

**Фактографические БД** содержат числовую и символьную информацию, которая описывает предметную область. Такие базы данных называются также структурированными. В системах фактографического типа в БД хранится информация об интересующих пользователя объектах предметной области в виде фактов; в ответ на запрос пользователя выдается требуемая информация об интересующем его объекте или сообщение о том, что информация отсутствует. Сведения о процессах или объектах могут поступать из различных сообщений и документов, поэтому здесь выявление нового атрибута (поля) происходит путем анализа каждого нового сообщения или документа, содержащего сведения об объекте.

**Лексикографические БД** содержат многоязыковые словари классификаторы, многоязычные словари, словари основ слов и т.п.).

В последние годы активно развивается объектно-ориентированный подход к созданию ИС. Объектные БД организованы как объекты и ссылки к объектам. Объект представляет собой данные и правила, по которым осуществляется операция с этими данными. Объект включает метод, который является частью определения объекта и запоминается вместе с объектом. В объектных БД данные запоминаются как объекты, классифицированные по типам классов и организованные в иерархическое семейство классов. Класс - коллекция объектов с одинаковыми свойствами. Объекты принадлежат классу. Классы организованы в иерархии.

3. По охвату программного обеспечения. Эта классификация может проводиться по разным признакам - по территориальному (город, страна), ведомственному, проблемному, временному.

4. По характеру организации данных БД могут быть разделены на:  
неструктурированные;  
частично структурированные;  
структурированные.

Этот классификационный признак относится к информации, представленной в символьном виде. К неструктурированным БД могут быть отнесены БД, организованные в виде семантических сетей. Частично структурированными можно считать БД в виде обычного текста или гипертекстовые системы. Структурированные БД требуют предварительного проектирования и описания структуры БД. Только тогда БД такого типа могут быть заполнены данными.

5. По типу использованных моделей структурированные данные делятся на:  
сетевые;  
иерархические;  
реляционные.

6. По технологии обработки данных:  
централизованные;  
распределенные.

Централизованные данные хранятся в памяти одной вычислительной системы. Если эта вычислительная система является компонентом сети ЭВМ, то возможен распределенный доступ к такой базе. Такая База данных состоит из нескольких возможно пересекающихся или даже дублирующих друг друга частей, хранимых в различных ЭВМ вычислительной сети. Работа с такой базой осуществляется с помощью системы управления распределенной базой данных (СУРБД).

7. По способу доступа к данным:  
база данных с локальным доступом (т.е. ПК);  
база данных с удаленным (сетевым) доступом.

Система централизованной базы данных с сетевым доступом предполагает различные архитектуры подобных систем:

файл-сервер;  
клиент- сервер.

**Файл-сервер.** Здесь предполагается выделение одной из машин в качестве центральной (сервер). На такой машине хранится совместно используемая централизованная база данных. Все другие машины сети выполняют функции рабочих станций, с помощью которых поддерживается доступ пользовательской системы к централизованной базе данных. Файлы базы данных в соответствии с пользовательскими запросами передаются на рабочие станции, где в основном производится обработка.

При большой интенсивности доступа к одним и тем же данным производительность информационной системы падает. Пользователи могут также создавать на рабочих станциях локальные базы данных, которые используются ими монополично (рисунок 2).

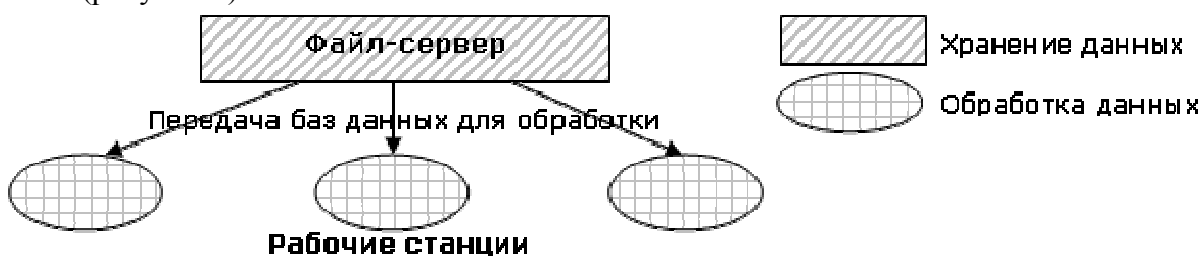


Рисунок 3.2 - Архитектура "файл-сервер"

**Клиент-сервер.** Здесь помимо хранения централизованной базы данных централизованная машина (сервер базы данных) должна обеспечивать выполнение основного объема обработки данных. Запрос на данные, выдаваемые клиентом (рабочей станцией), порождает поиск и извлечение данных на сервере. Извлеченные данные транспортируются от сервера к клиенту (рис.3).



Рисунок 3.3 - Архитектура "клиент-сервер"

Группа реляционных СУБД представлена на рынке программных продуктов очень широко (Paradox, Clarion, dBase II,III,IV, Clipper, FoxBASE, FoxPro, Clipper, Ms Access).

**СУБД** - комплекс программ и языковых средств, необходимых для создания баз данных, поддержания их и организации поиска в них необходимой информации.

Централизованный характер управления данными в базе данных предполагает необходимое существование группы лиц (одного лица), на которую возлагается функция администрирования данными, хранимыми в базе данных (администратор базы данных).

#### **Функции СУБД:**

1. Обеспечение пользователя средствами, позволяющими оперировать данными.
2. Предоставление пользователю средства описания данных.
3. Обеспечение операционного создания, уничтожения и манипулирования данными (вставка, поиск, редактирование).
4. Обеспечение секретности данных (защита от несанкционированного доступа).
5. Обеспечение целостности данных. Целостность данных гарантируется отсутствием избыточности. Так как данные хранятся в единичном экземпляре, но возможна ситуация, когда при фактических изменениях данных оно оказывается изменено не всех записях.
6. Обеспечивает проверку ограничений, предотвращающих запоминание некорректных данных в базе.



7. Обеспечение синхронизации. При работе в сети несколько пользователей одновременно получают доступ к данным.

8. Обеспечение защиты при аварийных сбоях и восстановление данных.

В настоящее время есть более 50 типов СУБД для IBM PC и совместимых с ними компьютеров.

2. Виды моделей данных.

Ядром любой базы данных является модель данных. Модель данных представляет собой множество структур данных, ограничений целостности и операций манипулирования данными. С помощью модели данных могут быть представлены объекты предметной области и взаимосвязи между ними.

Модель данных — совокупность структур данных и операций их обработки.

СУБД основывается на использовании иерархической, сетевой или реляционной модели, на комбинации этих моделей или на некотором их подмножестве.

Рассмотрим три основных типа моделей данных: иерархическую, сетевую и реляционную.

Иерархическая модель данных

Иерархическая структура представляет совокупность элементов, связанных между собой по определенным правилам. Объекты, связанные иерархическими отношениями, образуют ориентированный граф (перевернутое дерево).

К основным понятиям иерархической структуры относятся: уровень, элемент (узел), связь. Узел — это совокупность атрибутов данных, описывающих некоторый объект. На схеме иерархического дерева узлы представляются вершинами графа. Каждый узел на более низком уровне связан только с одним узлом, находящимся на более высоком уровне. Иерархическое дерево имеет только одну вершину (корень дерева), не подчиненную никакой другой вершине и находящуюся на самом верхнем (первом) уровне. Зависимые (подчиненные) узлы находятся на втором, третьем и т.д. уровнях. Количество деревьев в базе данных определяется числом корневых записей.

К каждой записи базы данных существует только один (иерархический) путь от корневой записи.

Сетевая модель данных

В сетевой структуре при тех же основных понятиях (уровень, узел, связь) каждый элемент может быть связан с любым другим элементом.

Реляционная модель данных

Понятие реляционный (англ. relation — отношение) связано с разработками известного американского специалиста в области систем баз данных Е. Кодда.

Эти модели характеризуются простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

Реляционная модель ориентирована на организацию данных в виде двумерных таблиц. Каждая реляционная таблица представляет собой двумерный массив и обладает следующими свойствами:

каждый элемент таблицы — один элемент данных;

все столбцы в таблице однородные, т.е. все элементы в столбце имеют одинаковый тип (числовой, символьный и т.д.) и длину;

каждый столбец имеет уникальное имя;

одинаковые строки в таблице отсутствуют;

порядок следования строк и столбцов может быть произвольным.

Отношения представлены в виде таблиц, строки которых соответствуют кортежам или записям, а столбцы — атрибутам отношений, доменам, полям.

Поле, каждое значение которого однозначно определяет соответствующую запись, называется простым ключом (ключевым полем). Если записи однозначно определяются значениями нескольких полей, то такая таблица базы данных имеет составной ключ.

Чтобы связать две реляционные таблицы, необходимо ключ первой таблицы ввести в состав ключа второй таблицы (возможно совпадение ключей); в противном случае нужно ввести в структуру первой таблицы внешний ключ — ключ второй таблицы.

3. Понятие информационного объекта. Нормализация отношений. Типы связей информационных объектов.

Информационный объект — это описание некоторой сущности (реального объекта, явления, процесса, события) в виде совокупности логически связанных реквизитов (информационных элементов). Такими сущностями для информационных объектов могут служить: цех, склад, материал, вуз, студент, сдача экзаменов и т.д.

Информационный объект определенного реквизитного состава и структуры образует класс (тип), которому присваивается уникальное имя (символьное обозначение), например Студент, Сессия, Стипендия.

Информационный объект имеет множество реализации — экземпляров, каждый из которых представлен совокупностью конкретных значений реквизитов и идентифицируется значением ключа (простого — один реквизит или составного — несколько реквизитов). Остальные реквизиты информационного объекта являются описательными. При этом одни и те же реквизиты в одних информационных объектах могут быть ключевыми, а в других — описательными. Информационный объект может иметь несколько ключей.

Понятие нормализации отношений

Одни и те же данные могут группироваться в таблицы (отношения) различными способами, т.е. возможна организация различных наборов отношений взаимосвязанных информационных объектов. Группировка атрибутов в отношениях должна быть рациональной, т.е. минимизирующей дублирование данных и упрощающей процедуры их обработки и обновления.

Определенный набор отношений обладает лучшими свойствами при включении, модификации, удалении данных, чем все остальные возможные наборы отношений, если он отвечает требованиям нормализации отношений.

Нормализация отношений — формальный аппарат ограничений на формирование отношений (таблиц), который позволяет устранить дублирование, обеспечивает непротиворечивость хранимых в базе данных, уменьшает трудозатраты на ведение (ввод, корректировку) базы данных.

Выделены три нормальные формы отношений и предложен механизм, позволяющий любое отношение преобразовать к третьей (самой совершенной) нормальной форме.

#### **Первая нормальная форма**

Отношение называется нормализованным или приведенным к первой нормальной форме, если все его атрибуты простые (далее неделимы). Преобразование отношения к первой нормальной форме может привести к увеличению количества реквизитов (полей) отношения и изменению ключа.

Например, отношение Студент = (**Номер**, Фамилия, Имя, Отчество, Дата, Группа) наводится в первой нормальной форме.

Вторая нормальная форма

Чтобы рассмотреть вопрос приведения отношений ко второй нормальной форме, необходимо дать пояснения к таким понятиям, как функциональная зависимость и полная функциональная зависимость.

Описательные реквизиты информационного объекта логически связаны с общим для них ключом, эта связь носит характер функциональной зависимости реквизитов.

Функциональная зависимость реквизитов — зависимость, при которой экземпляре информационного объекта определенному значению ключевого реквизита соответствует только одно значение описательного реквизита.

Такое определение функциональной зависимости позволяет при анализе всех взаимосвязей реквизитов предметной области выделить самостоятельные информационные объекты.

В случае составного ключа вводится понятие функционально полной зависимости.

Функционально полная зависимость не ключевых атрибутов заключается в том, что каждый не ключевой атрибут функционально зависит от ключа, но не находится в функциональной зависимости ни от какой части составного ключа.

Отношение будет находиться во второй нормальной форме, если оно находится в первой нормальной форме, и каждый не ключевой атрибут функционально полно зависит от составного ключа.

Третья нормальная форма

Понятие третьей нормальной формы основывается на понятии нетранзитивной зависимости.

Транзитивная зависимость наблюдается в том случае, если один из двух описательных реквизитов зависит от ключа, а другой описательный реквизит зависит от первого описательного реквизита.

Отношение будет находиться в третьей нормальной форме, если оно находится во второй нормальной форме, и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

Для устранения транзитивной зависимости описательных реквизитов необходимо провести «расщепление» исходного информационного объекта. В результате расщепления часть реквизитов удаляется из исходного информационного объекта и включается в состав других (возможно, вновь созданных) информационных объектов.

4. Построение инфологической модели.

Для построения инфологической модели данных определим сущности их связи и атрибуты.

Определим сущности:

1. Автомобили, наводящиеся в ведении предприятия;
2. Сотрудники (управляющий и обслуживающий составы);
3. Гаражное хозяйство;
4. Дополнительные данные.

Опишем атрибуты сущностей для каждой в отдельности:

Автомобили обладают следующими атрибутами:

- государственный номер;
- марка;
- вид (грузовой, легковой и т.п.)

Сотрудники:

- ФИО;
- профессия (водитель, сварщик, сборщик и т.п.);
- принадлежность (т.е. в каком гараже или отделе работает);
- закрепление (применительно к водителям, за каким автомобилем закреплен).

Гаражное хозяйство:

- вид транспорта (т.е. транспорт какой категории располагается в данном гараже);
- бригадиры;
- перечень сотрудников.

Дополнительные данные:

- информация о списанной и полученной технике;
- информация о ремонтах;
- информация о распределении транспорта (по маршрутам и т.д.);
- информация о грузоперевозках.

Построим инфологическую модель по определенным сущностям (рис.1):



Рис.1 – Инфологическая модель данных.

## **1. 9 Лекция №9 (2 часа).**

**Тема:** «Функциональные возможности СУБД»

### **1.9.1 Вопросы лекции:**

1. Непосредственное управление данными во внешней памяти
2. Управление буферами оперативной памяти
3. Управление транзакциями
4. Журнализация
5. Поддержка языков БД

### **1.9.2 Краткое содержание вопросов:**

1. Непосредственное управление данными во внешней памяти

Эта функция включает обеспечение необходимых структур внешней памяти как для хранения данных, непосредственно входящих в БД, так и для служебных целей, например, для убыстрения доступа к данным в некоторых случаях (обычно для этого используются индексы). В некоторых реализациях СУБД активно используются возможности существующих файловых систем, в других работа производится вплоть до уровня устройств внешней памяти. Но подчеркнем, что в развитых СУБД пользователи в любом случае не обязаны знать, использует ли СУБД файловую систему, и если использует, то как организованы файлы. В частности, СУБД поддерживает собственную систему именования объектов БД.

2. Управление буферами оперативной памяти

СУБД обычно работают с БД значительного размера; по крайней мере этот размер обычно существенно больше доступного объема оперативной памяти. Понятно, что если при обращении к любому элементу данных будет производиться обмен с внешней памятью, то вся система будет работать со скоростью устройства внешней памяти. Практически единственным способом реального увеличения этой скорости является буферизация данных в оперативной памяти. При этом, даже если операционная система производит общесистемную буферизацию (как в случае ОС UNIX), этого недостаточно для целей СУБД, которая располагает гораздо большей информацией о полезности буферизации той или иной части БД. Поэтому в развитых СУБД поддерживается собственный набор буферов оперативной памяти с собственной дисциплиной замены буферов.

Заметим, что существует отдельное направление СУБД, которое ориентировано на постоянное присутствие в оперативной памяти всей БД. Это направление основывается на предположении, что в будущем объем оперативной памяти компьютеров будет настолько велик, что позволит не беспокоиться о буферизации. Пока эти работы находятся в стадии исследований.

3. Управление транзакциями

Транзакция - это последовательность операций над БД, рассматриваемых СУБД как единое целое. Либо транзакция успешно выполняется, и СУБД фиксирует (COMMIT) изменения БД, произведенные этой транзакцией, во внешней памяти, либо ни одно из этих изменений никак не отражается на состоянии БД. Понятие транзакции необходимо для поддержания логической целостности БД. Если вспомнить наш пример информационной системы с файлами СОТРУДНИКИ и ОТДЕЛЫ, то единственным способом не нарушить целостность БД при выполнении операции приема на работу нового сотрудника является объединение элементарных операций над файлами СОТРУДНИКИ и ОТДЕЛЫ в одну транзакцию. Таким образом, поддержание механизма транзакций является обязательным условием даже однопользовательских СУБД (если, конечно, такая система заслуживает названия СУБД). Но понятие транзакции гораздо более важно в многопользовательских СУБД.

То свойство, что каждая транзакция начинается при целостном состоянии БД и оставляет это состояние целостным после своего завершения, делает очень удобным использование понятия транзакции как единицы активности пользователя по отношению

к БД. При соответствующем управлении параллельно выполняющимися транзакциями со стороны СУБД каждый из пользователей может в принципе ощущать себя единственным пользователем СУБД (на самом деле, это несколько идеализированное представление, поскольку в некоторых случаях пользователи многопользовательских СУБД могут ощутить присутствие своих коллег).

С управлением транзакциями в многопользовательской СУБД связаны важные понятия *сериализации транзакций* и *сериального плана выполнения смеси транзакций*. Под сериализацией параллельно выполняющихся транзакций понимается такой порядок планирования их работы, при котором суммарный эффект смеси транзакций эквивалентен эффекту их некоторого последовательного выполнения. Сериальный план выполнения смеси транзакций - это такой план, который приводит к сериализации транзакций. Понятно, что если удастся добиться действительно сериального выполнения смеси транзакций, то для каждого пользователя, по инициативе которого образована транзакция, присутствие других транзакций будет незаметно (если не считать некоторого замедления работы по сравнению с однопользовательским режимом).

Существует несколько базовых алгоритмов сериализации транзакций. В централизованных СУБД наиболее распространены алгоритмы, основанные на синхронизационных захватах объектов БД. При использовании любого алгоритма сериализации возможны ситуации конфликтов между двумя или более транзакциями по доступу к объектам БД. В этом случае для поддержания сериализации необходимо выполнить откат (ликвидировать все изменения, произведенные в БД) одной или более транзакций. Это один из случаев, когда пользователь многопользовательской СУБД может реально (и достаточно неприятно) ощутить присутствие в системе транзакций других пользователей.

#### 4. Журнализация

Одним из основных требований к СУБД является надежность хранения данных во внешней памяти. Под надежностью хранения понимается то, что СУБД должна быть в состоянии восстановить последнее согласованное состояние БД после любого аппаратного или программного сбоя. Обычно рассматриваются два возможных вида аппаратных сбоев: так называемые мягкие сбои, которые можно трактовать как внезапную остановку работы компьютера (например, аварийное выключение питания), и жесткие сбои, характеризующиеся потерей информации на носителях внешней памяти. Примерами программных сбоев могут быть: аварийное завершение работы СУБД (по причине ошибки в программе или в результате некоторого аппаратного сбоя) или аварийное завершение пользовательской программы, в результате чего некоторая транзакция остается незавершенной. Первую ситуацию можно рассматривать как особый вид мягкого аппаратного сбоя; при возникновении последней требуется ликвидировать последствия только одной транзакции.

Понятно, что в любом случае для восстановления БД нужно располагать некоторой дополнительной информацией. Другими словами, поддержание надежности хранения данных в БД требует избыточности хранения данных, причем та часть данных, которая используется для восстановления, должна храниться особо надежно. Наиболее распространенным методом поддержания такой избыточной информации является ведение журнала изменений БД.

Журнал - это особая часть БД, недоступная пользователям СУБД и поддерживаемая с особой тщательностью (иногда поддерживаются две копии журнала, располагаемые на разных физических дисках), в которую поступают записи обо всех изменениях основной части БД. В разных СУБД изменения БД журналируются на разных уровнях: иногда запись в журнале соответствует некоторой логической операции изменения БД (например, операции удаления строки из таблицы реляционной БД), иногда - минимальной внутренней операции модификации страницы внешней памяти; в некоторых системах одновременно используются оба подхода.

Во всех случаях придерживаются стратегии "упреждающей" записи в журнал (так называемого протокола Write Ahead Log - WAL). Грубо говоря, эта стратегия заключается в том, что запись об изменении любого объекта БД должна попасть во внешнюю память журнала раньше, чем измененный объект попадет во внешнюю память основной части БД. Известно, что если в СУБД корректно соблюдается протокол WAL, то с помощью журнала можно решить все проблемы восстановления БД после любого сбоя.

Самая простая ситуация восстановления - индивидуальный откат транзакции. Строго говоря, для этого не требуется общесистемный журнал изменений БД. Достаточно для каждой транзакции поддерживать локальный журнал операций модификации БД, выполненных в этой транзакции, и производить откат транзакции путем выполнения обратных операций, следуя от конца локального журнала. В некоторых СУБД так и делают, но в большинстве систем локальные журналы не поддерживают, а индивидуальный откат транзакции выполняют по общесистемному журналу, для чего все записи от одной транзакции связывают обратным списком (от конца к началу).

При мягком сбое во внешней памяти основной части БД могут находиться объекты, модифицированные транзакциями, не закончившимися к моменту сбоя, и могут отсутствовать объекты, модифицированные транзакциями, которые к моменту сбоя успешно завершились (по причине использования буферов оперативной памяти, содержимое которых при мягком сбое пропадает). При соблюдении протокола WAL во внешней памяти журнала должны гарантированно находиться записи, относящиеся к операциям модификации обоих видов объектов. Целью процесса восстановления после мягкого сбоя является состояние внешней памяти основной части БД, которое возникло бы при фиксации во внешней памяти изменений всех завершившихся транзакций и которое не содержало бы никаких следов незаконченных транзакций. Для того, чтобы этого добиться, сначала производят откат незавершенных транзакций (undo), а потом повторно воспроизводят (redo) те операции завершенных транзакций, результаты которых не отображены во внешней памяти. Этот процесс содержит много тонкостей, связанных с общей организацией управления буферами и журналом. Более подробно мы рассмотрим это в соответствующей лекции.

Для восстановления БД после жесткого сбоя используют журнал и архивную копию БД. Грубо говоря, архивная копия - это полная копия БД к моменту начала заполнения журнала (имеется много вариантов более гибкой трактовки смысла архивной копии). Конечно, для нормального восстановления БД после жесткого сбоя необходимо, чтобы журнал не пропал. Как уже отмечалось, к сохранности журнала во внешней памяти в СУБД предъявляются особо повышенные требования. Тогда восстановление БД состоит в том, что исходя из архивной копии по журналу воспроизводится работа всех транзакций, которые закончились к моменту сбоя. В принципе, можно даже воспроизвести работу незавершенных транзакций и продолжить их работу после завершения восстановления. Однако в реальных системах это обычно не делается, поскольку процесс восстановления после жесткого сбоя является достаточно длительным.

#### 5. Поддержка языков БД

Для работы с базами данных используются специальные языки, в целом называемые *языками баз данных*. В ранних СУБД поддерживалось несколько специализированных по своим функциям языков. Чаще всего выделялись два языка - *язык определения схемы БД (SDL - Schema Definition Language)* и *язык манипулирования данными (DML - Data Manipulation Language)*. SDL служил главным образом для определения логической структуры БД, т.е. той структуры БД, какой она представляется пользователям. DML содержал набор операторов манипулирования данными, т.е. операторов, позволяющих заносить данные в БД, удалять, модифицировать или выбирать существующие данные. Мы рассмотрим более подробно языки ранних СУБД в следующей лекции.

В современных СУБД обычно поддерживается единый интегрированный язык, содержащий все необходимые средства для работы с БД, начиная от ее создания, и обеспечивающий базовый пользовательский интерфейс с базами данных. Стандартным языком наиболее распространенных в настоящее время реляционных СУБД является язык SQL (Structured Query Language). В нескольких лекциях этого курса язык SQL будет рассматриваться достаточно подробно, а пока мы перечислим основные функции реляционной СУБД, поддерживаемые на "языковом" уровне (т.е. функции, поддерживаемые при реализации интерфейса SQL).

Прежде всего, язык SQL сочетает средства SDL и DML, т.е. позволяет определять схему реляционной БД и манипулировать данными. При этом именование объектов БД (для реляционной БД - именование таблиц и их столбцов) поддерживается на языковом уровне в том смысле, что компилятор языка SQL производит преобразование имен объектов в их внутренние идентификаторы на основании специально поддерживаемых служебных таблиц-каталогов. Внутренняя часть СУБД (ядро) вообще не работает с именами таблиц и их столбцов.

Язык SQL содержит специальные средства определения ограничений целостности БД. Опять же, ограничения целостности хранятся в специальных таблицах-каталогах, и обеспечение контроля целостности БД производится на языковом уровне, т.е. при компиляции операторов модификации БД компилятор SQL на основании имеющихся в БД ограничений целостности генерирует соответствующий программный код.

Специальные операторы языка SQL позволяют определять так называемые представления БД, фактически являющиеся хранимыми в БД запросами (результатом любого запроса к реляционной БД является таблица) с именованными столбцами. Для пользователя представление является такой же таблицей, как любая базовая таблица, хранимая в БД, но с помощью представлений можно ограничить или наоборот расширить видимость БД для конкретного пользователя. Поддержание представлений производится также на языковом уровне.

Наконец, авторизация доступа к объектам БД производится также на основе специального набора операторов SQL. Идея состоит в том, что для выполнения операторов SQL разного вида пользователь должен обладать различными полномочиями. Пользователь, создавший таблицу БД, обладает полным набором полномочий для работы с этой таблицей. В число этих полномочий входит полномочие на передачу всех или части полномочий другим пользователям, включая полномочие на передачу полномочий. Полномочия пользователей описываются в специальных таблицах-каталогах, контроль полномочий поддерживается на языковом уровне.

Более точное описание возможных реализаций этих функций на основе языка SQL будет приведено в лекциях, посвященных языку SQL и его реализации.



## **2. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ**

### **2.1 Лабораторная работа №1 (2 часа).**

**Тема:** «Операционная система WINDOWS. Операции с объектами WINDOWS»

**2.1.1 Цель работы:** Освоить технологию работы с объектами ОС Windows: папками, файлами и ярлыками.

#### **2.1.2 Задачи работы:**

1. Ознакомиться с операционной системой WINDOWS.
2. Ознакомиться с операциями с объектами WINDOWS.

#### **2.1.3 Перечень приборов, материалов, используемых в лабораторной работе:**

1. Персональный компьютер
2. Операционная система Windows

#### **2.1.4 Описание (ход) работы:**

Объекты файловой системы ОС Windows

Разберем теперь объекты, с которыми работает ОС семейства Windows, – диск, файл, папку (каталог) и ярлык.

Дадим определение такого объекта, как диск.

Диск – носитель информации, который может представлять собой физическое устройство или часть пространства носителя, называемого логическим диском. Он именуется одной буквой латинского алфавита с добавлением после нее знака «:» – двоеточия, именование происходит последовательным присвоением латинских букв по алфавиту.

Приведем пример именования дисков в ОС Windows:

**A:** и **B:** – накопитель на гибком магнитном диске (съемные дискеты);

**C:** и **D:** – накопитель на жестком магнитном диске (винчестер) и логический диск (таких дисков может быть и больше);

**E:** и **F:** – накопители на CD- и/или DVD-дисках (берется следующая буква за последней буквой жесткого/логического диска);

**G:** – Flash-накопитель;

**K:** **Z:** – сетевые диски (для их именования могут использоваться любые незадействованные буквы).

Дадим определение следующему объекту – файлу.

**Файл** – это именованная область внешней памяти, в которую можно записывать и из которой можно считывать данные (информацию). Файлы хранятся в памяти, не зависящей от энергопитания, на дисках.

Перечислим основные цели использования файла:

долговременное и надежное хранение информации. Долговременность достигается за счет использования энергонезависимых запоминающих устройств, а высокая надежность определяется средствами защиты доступа к файлам и общей организацией программного кода ОС, при которой сбои аппаратуры чаще всего не разрушают информацию, хранящуюся в файлах;

совместное использование информации. Файлы обеспечивают естественный и легкий способ разделения информации между приложениями и пользователями за счет наличия понятного человеку символьного имени и постоянства хранимой информации и расположения файла. Пользователь должен иметь удобные средства работы с файлами, включая каталоги-справочники, объединяющие файлы в группы, средства поиска файлов по признакам, набор команд для создания, модификации и удаления файлов. Файл может быть создан одним пользователем, а затем использоваться совсем другим пользователем, при этом создатель файла или администратор могут определить права доступа к нему других пользователей. Эти цели реализуются в ОС файловой системой.

Данные, с которыми мы работаем, различны, поэтому ОС использует разные средства для их обработки. Чтобы ОС правильно «понимала», что следует выполнить с разного вида данными, их разделяют на несколько типов.

Файловые системы поддерживают несколько функционально различных типов файлов, в число которых, как правило, входят обычные файлы, файлы-папки (каталоги), специальные файлы, именованные конвейеры, отображаемые в память файлы, и др.

*Обычные файлы*, или просто *файлы*, содержат информацию произвольного характера, которую заносит в них пользователь или которая образуется в результате работы системных и пользовательских программ. Большинство современных операционных систем не ограничивает и не контролирует содержимое и структуру обычного файла. Содержание обычного файла определяется приложением, которое с ним работает. Например, текстовый редактор создает текстовые файлы, состоящие из строк символов, представленных в каком-либо коде. Это могут быть документы, исходные тексты программ и т. п. Текстовые файлы можно прочитать на экране и распечатать на принтере. Двоичные файлы не используют коды символов, они часто имеют сложную внутреннюю структуру, например исполняемый код программы или архивный файл. Все операционные системы должны уметь распознавать хотя бы один тип файлов – их собственные исполняемые файлы.

Все типы файлов имеют символьные имена. В иерархически организованных файловых системах обычно используются следующие виды имен файлов: *простое* и *полное (составное)*.

*Простое*, или *короткое, символьное* имя идентифицирует файл в пределах одной папки. Простые имена присваивают файлам пользователи и программисты, при этом они должны учитывать ограничения ОС как на номенклатуру символов, так и на длину имени.

Пользователю гораздо удобнее работать с длинными именами, поскольку они позволяют дать файлам легко запоминающиеся названия, ясно говорящие о том, что содержится в этом файле. В файловых системах NTFS и FAT32, входящих в состав операционной системы Windows, имя файла может содержать до 255 символов (буквы и цифры латинского и русского алфавита), **кроме** символов \ / : \* ? " | < > и имен **CON**, **NUL**, **PRN**, **AUX**, **COM19**, **LPT19**.

Вторая часть имени – это **тип (расширение, формат)**, который начинается с символа точки (.) и может включать от 1 до 4 символов. Существуют зарегистрированные типы, по которым ОС связывает файл с приложением (программой), которой он будет открыт. Этот принцип называется объектно-ориентированным подходом ОС.

В иерархических файловых системах разным файлам разрешено иметь одинаковые простые символьные имена при условии, что они принадлежат разным папкам. То есть здесь работает схема *«много файлов – одно простое имя»*. Для однозначной идентификации файла в таких системах используется так называемое полное имя.

*Полное имя* представляет собой цепочку, состоящую из имени диска, простых символьных имен всех папок (каталогов), через которую проходит путь от корня до данного файла и простого имени файла. Таким образом, *полное* имя является *составным*, в котором имена папок отделены друг от друга обратным слешем «\», например: *d:\naпка1\папка2\пример1.txt*.

В иерархической файловой системе между файлом и его полным именем имеется однозначное соответствие *«один файл – одно полное имя»*.

В файловых системах, имеющих сетевую структуру, файл может входить в несколько папок (каталогов), а значит, иметь несколько полных имен; здесь справедливо соответствие *«один файл – много полных имен»*. В обоих случаях файл однозначно идентифицируется полным именем.

Теперь дадим определение папки (каталога).

**Папка (каталог)** – это особый тип файлов, который содержит системную справочную информацию о наборе файлов, сгруппированных пользователями по какому-либо признаку.

Папки устанавливают соответствие между именами файлов и их характеристиками, используемыми файловой системой для управления файлами, такими как информация о типе файла и расположении его на диске, правах доступа к файлу, его размере и дате его создания или модификации. Во всех остальных отношениях папки рассматриваются файловой системой как обычные файлы. Во многих операционных системах в папку могут входить файлы любых типов, в т. ч. другие папки, за счет чего образуется древовидная (иерархическая) структура, удобная для поиска.

С появлением графических ОС появляется еще один вид файла – ярлык.

**Ярлык** – это файл, содержащий ссылку на объект, программу или команду, а также дополнительную информацию.

Ярлык служит для быстрого доступа к нужному объекту. Его можно размещать в любом месте, выбранном пользователем. Для одного объекта можно создать несколько ярлыков, это можно использовать для запуска одного и того же приложения с разными опциями запуска. Ярлыки отображаются значками, подобными объектам, к которым они созданы, с добавлением стрелочки в нижнем левом углу. Файлы типа \*.lnk и \*.pif – это файлы ярлыков.

Файлы типа \*.lnk – это стандартные ярлыки, создаваемые операционной системой Windows. Существуют файлы типа \*.pif – это также ярлыки, но к DOS-приложениям, в свойствах таких ярлыков, кроме стандартных, присутствуют и дополнительные настройки.

1. Используя пункт Главного меню Выполнить, загрузите текстовый редактор Блокнот (исполнимый файл notepad.exe).
2. Завершите работу программы Блокнот.
3. Создайте на Рабочем столе папку под именем *Студент*.
4. В корневой папке рабочего диска создайте папку с именем *Лабораторная №2*.
5. Откройте созданную папку *Лабораторная №2* и создайте в ней вложенную папку с Вашим именем.
6. Измените имя вложенной папки на *1-й курс*.
7. Создайте в папке *1-й курс*, используя текстовый редактор WordPad, текстовый документ с именем *Фамилии.txt*, в который введите название своей группы и несколько имен и фамилий сокурсников.
8. Скопируйте файл *Фамилии.txt* в папку *Студент*.
9. Откройте окна папок *Студент* и *1-й курс* и покажите результат работы преподавателю.
10. Удалите папку *Студент*.
11. Создайте на Рабочем столе ярлык для программы Калькулятор с именем *Для выполнения вычислений* (исполнимый файл программы calc.exe). Для определения места хранения программы можно воспользоваться пунктом Найти в Главном меню.
12. Используя созданный ярлык, запустите программу Калькулятор.
13. Завершите работу программы Калькулятор.
14. Запустите стандартную программу Проводник и дальнейшие действия с объектами выполняйте в ней. Вид окна программы Проводник ОС Windows Vista – рис. 1.

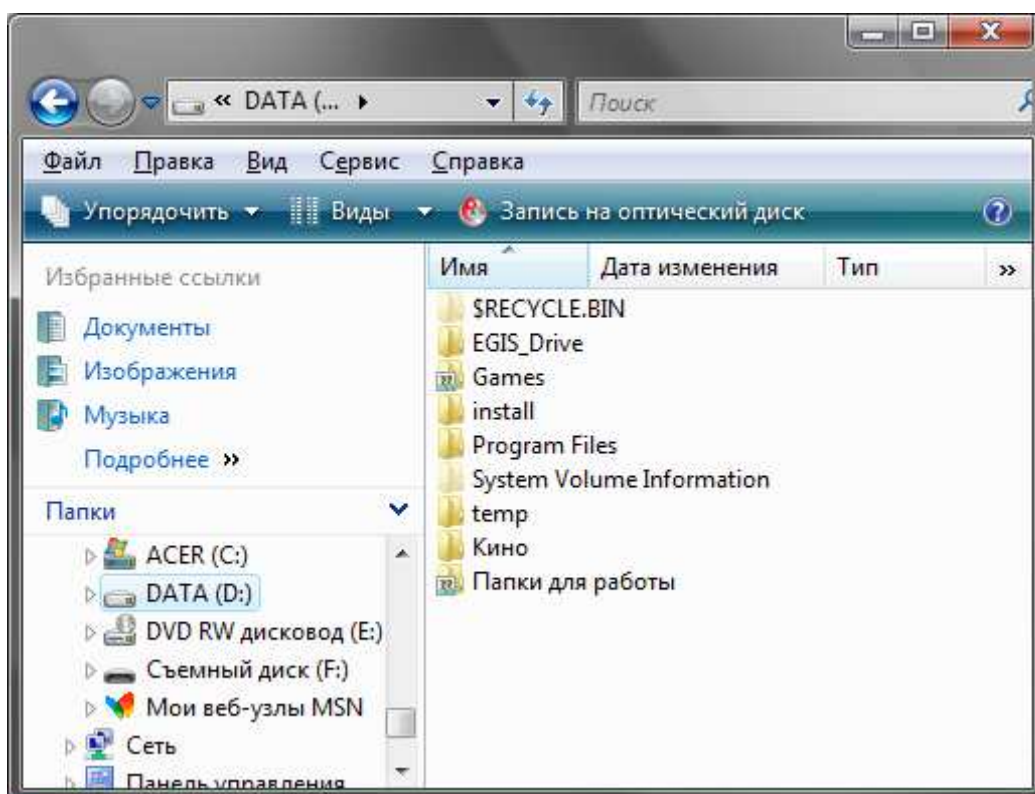


Рис. 1 – Окно программы Проводник

15. Переместите созданный Вами ярлык с Рабочего стола в папку *Лабораторная №2*.
16. Скопируйте из вложенной папки *Help* папки *Windows* диска C:, в папку *Лабораторная №2* первые пять файлов.
17. Перетаскивая значки, настроив для этого нужный вид отображения значков, создайте в папке *Лабораторная №2* из них две группы:
  - в левом верхнем углу окна – папки и ярлыки,
  - в правом нижнем углу окна – файлы.
18. Выполняя действия: Вид Упорядочить, выстроить по сетке значки в папке *Лабораторная №2*.
19. Покажите результат своей работы преподавателю.
20. Упорядочите значки в папке *Лабораторная №2* по имени.
21. Выключите панель инструментов (Обычные кнопки).
22. Включите панель инструментов (Обычные кнопки).
23. Установите режим отображения объектов Список в папке *Лабораторная №2*.
24. Установите режим отображения объектов Таблица в папке *Лабораторная №2*.
25. Выполните сортировку значков в папке *Лабораторная №2* по размерам в прямом и обратном порядке. Покажите результат работы преподавателю.
26. Удалите созданные Вами объекты и закройте все открытые окна.

## **2.2 Лабораторная работа №2 (2 часа).**

**Тема:** «Оболочка Norton Commander. Основные команды и принципы работы»

**2.2.1 Цель работы:** Ознакомиться с оболочкой Norton Commander.

**2.2.2 Задачи работы:**

1. Изучить оболочку Norton Commander.
2. Освоить основные команды и принципы работы

**2.2.3 Перечень приборов, материалов, используемых в лабораторной работе:**

1. Персональный компьютер
2. Операционная система Windows

**2.2.4 Описание (ход) работы:**

### **Программная оболочка Norton Commander и основные функции**

Программа Norton Commander, разработанная фирмой Peter Norton Computing (в 1992г. № эта фирма вошла в корпорацию Symantec), является одной из наиболее популярных программ-оболочек для работы с операционной системой DOS. Кроме Norton Commander, имеются и другие программы-оболочки, например QDos, PathMinder, XTree и др.

Norton Commander позволяет выполнять больше количество различных функций, в частности:

- ✓ Наглядно изображать содержание каталогов на дисках;
- ✓ Удобно копировать, переименовать, пересылать и удалять файлы;
- ✓ Изображать дерево каталогов на диске (на этом можно переходить в нужный каталог, создавать, переименовать и удалять каталоги);
- ✓ Просмотреть текстовые, графические и архивные файлы, базы данных и таблицы табличных процессоров и т.д.;
- ✓ Редактировать текстовые файлы;
- ✓ Выполнять любые команды DOS;
- ✓ Работать с архивными файлами почти так же удобно, как с оглавлениями каталогов;
- ✓ С помощью одного нажатия клавиши выполнять стандартные действия для типа файлов;
- ✓ ... и многое другое.

### **Содержание панелей Norton Commander**

В каждой панели Norton Commander может изображаться:

- оглавление каталога на диске. Наверху панели выводится имя этого каталога;
- дерево каталогов на диске. Наверху панели выводится «Tree» («Дерево каталогов»);
- сводная информация о диске и каталоге на другой панели. Наверху панели выводится «Info» («Информация»);
- содержание файла, выделенного на другой панели. Наверху панели обычно выводится «View» («Просмотр»);
- оглавление каталога на диске другого компьютера, соединенного с данным. Эта панель установлена так же, как панель с оглавлением каталога на диске. Наверху панели выводится «Link» («Связь») и имя каталога;
- оглавление архивного файла (архива). Панель устроена так же, как панель с оглавлением каталога диска. Наверху панели выводится тип архива, например «Zip», или имя архива;
- сведения о количестве и объеме выделенных файлов и подкаталогов в каталоге на другой панели. Наверху панели выводится «Directory Information» («Паспорт каталога»);
- результаты поиска файлов. Наверху панели выводится «Find File» («Поиск файлов»);

Norton Commander (NC) является наиболее популярной программной оболочкой для работы с DOS. Запуск Norton Commander - записать NC в командную строку и нажать Enter. Как правило, запуск NC производится автоматически при загрузке машины при помощи файла autoexec.bat.

Версии NC - 1.0 - 5.0. На современных машинах используются версии NC 4.0, NC 5.0 (англоязычная и русифицированная версии). Есть версии NC для Windows. На старых машинах и системных дискетах используются версии NC 2.0, NC 3.0. Каталоги в Norton Commander изображаются большими буквами, а файлы маленькими буквами. При помощи NC пользователи просматривают и создают файлы и каталоги, копируют, пересылают, переименовывают и удаляют файлы и каталоги, запускают программы и могут делать многое другое. Например, можно соединять (Link) между собой через порты LPT или COM два компьютера и копировать информацию с одного на другой. Многие операции в NC4, NC5 удобно делать с помощью мыши. При копировании файлов и каталогов в NC5 используется технология Drag and Drop (бери и тащи с помощью мыши). При помощи NC можно создавать и редактировать небольшие текстовые файлы (до 50 Кб). Файлы большего размера редактируют с помощью внешнего (альтернативного) редактора, который может вызываться NC при нажатии Alt+F4.

### **Основные функциональные клавиши Norton Commander**

(для версии NC 4.0 или NC 5.0)

F1 - Вызов помощи

F2 - Меню пользователя

F3 - Просмотр файла

F4 - Редактирование файла

F5 - Копирование файла (группы файлов)

F6 - Переименование или перенос файла (группы файлов)

F7 - Создание каталога

F8 - Удаление файла или каталога (группы файлов)

F9 - Управляющее меню Norton Commander

F10- Выход из Norton Commander

Ctrl+F1 - Убрать/Восстановить левую панель экрана

Ctrl+F2 - Убрать/Восстановить правую панель экрана

Ctrl+F3 - Сортировка файлов по имени (по алфавиту)

Ctrl+F4 - Сортировка файлов по расширению

Ctrl+F5 - Сортировка файлов по времени создания

Ctrl+F6 - Сортировка файлов по размеру

Ctrl+F7 - Сортировка файлов не устанавливается

Ctrl+F9 - Печать файла

Shift+F4 - Создание текстового файла. Имя файла задается.

Shift+F9 - Сохранение настроек Norton Commander

Alt+F1 (Alt+F2) - Переключение на другой диск/дискет на левой (правой) панели

Alt+F4 - Редактирование файла с помощью альтернативного редактора

Alt+F5 - Создание архива

Alt+F6 - Извлечение файлов из архива

Alt+F7 - Поиск файла на диске

Alt+F8 - История ранее выполненных команд

Alt+F9 - Переключение режима монитора с 25 на 43 линии

Alt+F10 - Вывод дерева каталогов, поиск каталога на диске. При этом в корневом каталоге создается файл treeinfo.ncd, в котором содержится информация о всем дереве каталогов.

Insert - Выделение одного файла "\*" (на левой части клавиатуры) - Выделение всех файлов в каталоге или инверсия выделения файлов

"+"-Enter - Выделение всех файлов в каталоге

"-"-Enter - Отмена выделения файлов  
Ctrl+O - Убрать/Восстановить панели экрана  
Ctrl+U - Поменять панели местами  
Ctrl+P - Убрать/Восстановить не текущую панель экрана  
Ctrl+L - Вывод информационной панели диска (дискеты)  
Ctrl+Q - Быстрый просмотр файла, информация о содержимом каталога  
Ctrl+E - Вывод ранее выполненной команды из буфера History  
Ctrl+X - Вывод следующей выполненной команды из буфера History  
Ctrl+J (или Ctrl+Enter) - Перенос выделенного курсором файла в командную строку  
Ctrl+R - Быстрое обновление информации на панели диска (дискеты)  
Ctrl+Z - Паспорт каталога, информация о содержимом каталога, и число дискет формата 1,2 и 1,44 Мб, на которых эта информация может быть размещена (NC5).  
Ctrl+\ - быстрый переход из подкаталога в корневой каталог  
Tab - Перемещение курсора с левой панели на правую и обратно  
Esc - Отмена команд в Norton Commander.  
Поиск подстроки при просмотре файла - F3, F7, ввести искомое слово и Enter.

### **Выделение файлов и каталогов**

Выделение файлов и каталогов используется как правило непосредственно перед их копированием, перемещением или удалением.

1. Для выделения (отмены выделения) малых групп или одиночных файлов и каталогов нужно использовать клавишу Insert или правую кнопку мыши.

2. Для выделения (отмены выделения) всех файлов в каталоге удобно использовать клавишу "\*", расположенную справа. Для выделения всех файлов также можно использовать комбинацию клавиш "+"-Enter, для отмены выделения "-"-Enter. Клавиши "+" и "-" расположены справа. Клавишу "\*" можно использовать для инверсии выделения, то есть, если предварительно выделена группа файлов, и нажать "\*", то с выделенных файлов будет снято выделение, а не выделенные файлы будут выделены.

3. Для выделения группы файлов с одинаковым расширением нужно нажать клавишу "+", расположенную справа, затем ввести в окно "Выбор файлов" вместо правой звездочки расширение файла, например, \*.txt, затем нажать Enter. Для отмены такого выделения используют "-"-Enter. Аналогично выделяют группу файлов с одинаковым именем, например, ps.\* в каталоге NC.

### **Копирование файлов и каталогов (для версии NC5-рус.)**

1. Откройте каталог с файлами и (или) подкаталогами.

2. Произведите выделение файлов и каталогов, предназначенных для копирования. Если копируется один файл или один каталог, то на него просто надо поместить курсор.

3. На противоположной панели откройте каталог для приема файлов и подкаталогов или вставьте дискету и откройте ее (Alt-F1 или Alt-F2), если копирование производится на дискету.

4. При копировании курсор должен находиться со стороны копируемых файлов или каталогов.

При копировании файлов с дискеты на диск курсор должен находиться со стороны выделенных файлов на дискете, а на диске следует открыть каталог для приема файлов.

5. Нажимаем клавишу F5 и Enter, если копируем только файлы. При копировании нескольких подкаталогов после нажатия F5 надо указать мышью "Включая подкаталоги" и затем Enter.

Процесс копирования может занимать какое-то время, в зависимости от объема копируемой информации и быстродействия компьютера. Прерывание или отмена копирования - Esc.

Перенос (перемещение) файлов и каталогов. Отличается от копирования тем, что вместо F5 надо нажать F6.

### **Переименование файлов и каталогов**

1. Установите курсор на файл или каталог.
2. Нажмите клавишу F6 и очистите клавишей Delete или Backspace окно "Переименование файлов".
3. Введите в окно "Переименование файлов" новое имя (расширение) файла или каталога, и Enter.

#### **Удаление файлов и каталогов**

1. Будьте внимательны, чтобы случайно не удалить важную информацию!
2. Выделите подлежащие удалению файлы и каталоги и установите курсор со стороны выделенных файлов (каталогов).
3. Нажмите клавишу F8, появится окно "Удаление файлов" и затем нажмите Enter, если удаляете только файлы. В появившемся втором окне подтверждения удаления укажите "Все" и Enter. При удалении одного файла с атрибутом "Архивный" окно подтверждения удаления не появляется.

При удалении нескольких подкаталогов после нажатия F8 надо указать мышью "Включая подкаталоги" и затем Enter. В появившемся втором окне подтверждения удаления укажите "Все" и Enter.

Отмена удаления - Esc.

#### **Поиск файлов на диске**

1. Нажимаем Alt-F7. Появляется трафарет "Поиск файла". Отмечаем, если не отмечено,

Место поиска - Весь диск C: или Каталоги (вводим имя каталога для поиска). Диск при необходимости можно сменить кнопкой "Диск", если поиск проводится не на диске C:.

2. Вводим в окно "Найти файлы" имя файла или его часть, и (или) расширение файла или его часть. Недостающие части имени или расширения заменяем знаками "\*", когда число недостающих букв неизвестно, или знаками "?" вместо недостающих букв, когда число недостающих букв известно. Учитываем, что число букв в имени не может превышать 8, а в расширении 3. Если об имени и расширении файла вообще ничего неизвестно, то поиск файла невозможен.

3. Указываем мышью кнопку "Старт". Повторный поиск - кнопка "Новый".

4. Наблюдаем найденные файлы, если поиск дал результаты. Если результатов поиска нет, то файлы не найдены.

5. Выбираем курсором из найденных файлов искомый и переходим к нему, указывая мышью кнопку "Перейти к". Для просмотра найденного файла можно нажать клавишу F3.

6. Выход из режима поиска файлов - Esc.

#### **Поиск каталогов на диске**

1. Нажимаем Alt-F10. Появляется трафарет "Дерево каталогов". При первом поиске создается файл treeinfo.ncd со списком всех каталогов на диске C:.. Этот файл надо обновить, если он старый, предварительно удалив его с диска.

2. В окно "Поиск" вводим первую букву или цифру имени каталога. Если найденный каталог не тот, что нужен, то вводим вторую букву имени и т.д., пока не найдем искомый каталог. Если буква или цифра не вводится, значит такого каталога нет.

3. Чтобы проверить, нет ли других каталогов с тем же именем, нажимаем Ctrl-Enter несколько раз, пока не найдем все каталоги.

4. Для перехода в искомый каталог надо нажать Enter, когда курсор установлен на нем в результате поиска.

5. Выход из режима поиска каталога - Esc.

#### **Другой способ поиска каталогов (1-й способ лучше):**

1. Нажимаем Alt-F7. Появляется трафарет "Поиск файла".
2. Укажите мышью кнопку "F10-Дерево". При первом поиске создается файл treeinfo.ncd со списком всех каталогов на диске C:..



3. В окно "Поиск" вводим первую букву или цифру имени каталога. Если найденный каталог не тот, что нужен, то вводим вторую букву имени и т.д., пока не найдем искомый каталог. Если буква или цифра не вводится, значит такого каталога нет.

4. Выход из режима поиска каталога - Esc.

#### **Быстрый поиск файла в известном каталоге**

1. Поиск файлов будет успешней и быстрее, если предварительно установить сортировку файлов по имени Ctrl-F3. Откройте каталог с файлами. Если в каталоге есть подкаталоги, то установите курсор на первый файл.

2. Нажмите клавишу Alt и введите первую букву (цифру) в имени файла. Курсор переместится на файл, начинающийся с введенной буквы. Если файла на эту букву нет, то буква не будет вводиться. Продолжаем вводить при необходимости следующую букву имени файла, пока не найдем искомый файл.

3. Выход из режима поиска файлов - Esc.

#### **Быстрый поиск подкаталога в известном каталоге**

1. Поиск каталогов будет успешней и быстрее, если предварительно установить сортировку файлов по имени Ctrl-F3. Откройте каталог с подкаталогами или выйдете в корневой каталог, если поиск надо провести в нем.

2. Нажмите клавишу Alt и введите первую букву (цифру) в имени каталога. Курсор переместится на каталог, начинающийся с введенной буквы. Если каталога на эту букву нет, то буква не будет вводиться. Продолжаем вводить при необходимости следующую букву имени каталога, пока не найдем искомый каталог.

3. Выход из режима поиска каталога - Esc.

#### **Действие Norton Commander при нажатии клавиши Enter.**

1. Если выделено имя каталога, то при нажатии Enter будет выведено оглавление каталога.

2. Если курсор установлен на файл с расширением exe, com, bat, то при нажатии Enter начнется исполнение этого файла (запуск программы).

3. Если выделено имя архива (файлы с расширением arj, zip, lzh, arc, pak, ice, zoo), то при нажатии Enter Norton Commander войдет в архив и выведет оглавление архива. Это свойство NC используется для архивации.

4. Если выделенный файл имеет расширение, указанное в файле nc.ext, то действие Norton Commander при нажатии Enter будет определяться файлом nc.ext.

#### **Структура файла nc.ext и его редактирование.**

На практике очень удобно, чтобы файл с соответствующим расширением был бы связан с соответствующей ему программой, например:

текстовый файл \*.txt - с текстовым редактором, файл базы данных

\*.dbf - с системой управления базами данных, программы \*.bas ,

\*.pas - со своей средой программирования, архивный файл - со своим архиватором.

Тогда при нажатии Enter Norton Commander будет автоматически загружать соответствующий выделенный файл в соответствующую ему программу. Это экономит время пользователя ПК и облегчает его работу. Для создания такого режима работы служит файл nc.ext. Он находится в каталоге NC со всеми остальными файлами Norton Commander и как любой другой текстовый файл может быть отредактирован в любом текстовом редакторе для DOS, в том числе и в редакторе, встроенном в Norton Commander. Для редактирования файла nc.ext проще всего найти его курсором и нажать F4.

Файл nc.ext может иметь, например, следующую структуру:

txt: lexicon !!

pas: turbo !!

bas: qbasic !!

dbf: c:\clipper\dbu\dbu !!

rar: rar en !!

ain: ain l !.!

В файле `nc.ext` сперва указывают расширение файла, например `txt`, а затем командный файл той программы, с которой он связывается, например, `lexicon`. В общем случае надо указывать полный путь этого файла, например, `c:\clipper\dbu\dbu`. В конце каждой команды ставится `!.` В этом случае Norton Commander при нажатии Enter будет загружать соответствующий файл в соответствующую ему программу, согласно записи в файле `nc.ext`. Длина файла `nc.ext` в NC4 не должна превышать величины около 700 байт. Иначе последние команды файла `nc.ext` не будут выполняться. Последние две команды позволяют удобно просматривать RAR и AIN архивы при помощи NC.

### **Меню пользователя и редактирование файла `nc.mnu`.**

Меню пользователя Norton Commander значительно облегчает и ускоряет запуск прикладных программ и поэтому широко применяется на практике. Меню пользователя содержится в файле `nc.mnu`. Если файл `nc.mnu` находится в каталоге Norton Commander (NC), то такое меню пользователя является Главным (Main). Длина файла `nc.mnu` в NC4 ограничена величиной около 1.6 Кб. Иначе последние пункты меню не будут выполняться. Главное меню вызывается при нажатии F2 из любого каталога, если в нем нет своего файла `nc.mnu`. В последнем случае будет вызываться Локальное меню (когда файл `nc.mnu` находится в любом каталоге, кроме каталога NC). Файл `nc.mnu` является текстовым и редактируется в любом текстовом редакторе, но чаще всего для этого применяется встроенный в Norton Commander редактор. Структура файла `nc.mnu` может иметь в простейшем случае следующий вид:

```
P: Парковка машины
park
L: Лексикон
lexicon
W: Windows-95
win
E: Фотон
c:\ft\ft
N: Norton Utilities
norton
```

Сначала в файле `nc.mnu` указывают ту клавишу (латинскую букву или цифру), которую надо будет нажимать для запуска программы. Затем на этой же строчке после двоеточия дают описание запускаемой программы на русском или английском языке. На второй строчке после двух пробелов записывается команда DOS, в общем случае с указанием полного пути командного файла. Иногда продолжение этой команды записывается на третьей, четвертой и т.д. строчке, если используется команда DOS `cd`. В нашем примере при нажатии клавиши W будет запущен Windows. Расширения файлов `exe`, `com`, `bat` можно в командах не указывать.

### **Смена атрибута файла**

Файл может иметь один из следующих четырех атрибутов:

1. Архивный (Archive) - это основной трибут файлов. При создании файл, как правило, по умолчанию получает этот атрибут.
2. Только для чтения (Read Only) - применяется для защиты файла от случайного удаления. При удалении такого файла будет выдан запрос на удаление. Архивный файл удаляется без запроса.
3. Скрытый (Hidden) - применяется для защиты от удаления особо важных файлов. Для этого файл делают невидимым с помощью NC.
4. Системный (System) - аналогичен скрытому файлу.

Для смены атрибута файла используют Меню, Файл, Атрибут (F9,F,A), выбирают мышью или пробелом один из указанных атрибутов и устанавливают его (Set). Для смены

атрибута одновременно у целой группы файлов ее выделяют, затем Меню, Файл, Атрибут (F9,F,A), устанавливают новый атрибут (Set) и удаляют старый атрибут (Clear).

Для того, чтобы сделать видимыми (или невидимыми) скрытые и системные файлы необходимо вызвать Меню F9, пункт Option, Configuration, Show Hidden Files, указать ОК. Аналогично скрытые файлы делают невидимыми.

### **Дисковые функции Norton Commander 5.0**

NC 5.0 выполняет ряд дисковых функций в пункте меню Disk, которых не было у NC 4.0. Раньше эти функции были только у комплекта системных программ Norton Utilities.

1. Копирование дискет (Copy Diskette). Позволяет создавать точные копии дискеты.
2. Форматирование дискет (Format Diskette).

Лучше использовать режим безопасного форматирования (Safe Format).

При форматировании возможно создание Системной дискеты.

3. Создание метки на диске, дискете (Label Disk).

Метка на диске (Volume Label) содержит не более 11 символов.

Это могут быть латинские буквы, цифры и некоторые спецсимволы.

4. Очистка диска (Disk Cleanup).

Позволяет очистить диск от ненужных и временных файлов. Но надо быть осторожным, чтобы случайно не стереть важную информацию!

### **2.3 Лабораторная работа №3 (2 часа).**

**Тема:** «Стандартные приложения WINDOWS: Проводник, Калькулятор, Paint, Блокнот»

**2.3.1 Цель работы:** Научиться работать со стандартными приложениями WINDOWS.

#### **2.3.2 Задачи работы:**

1. Научиться работать с приложением проводник.
2. Научиться работать с приложением калькулятор
3. Научиться работать с приложением Paint.
4. Научиться работать с приложением блокнот

#### **2.3.3 Перечень приборов, материалов, используемых в лабораторной работе:**

1. Персональный компьютер
2. Операционная система Windows

#### **2.3.4 Описание (ход) работы:**

К стандартным приложениям, входящим в операционную систему Windows, относится ограниченный набор прикладных программ для выполнения простейших задач, не требующих установки более мощных программных средств.

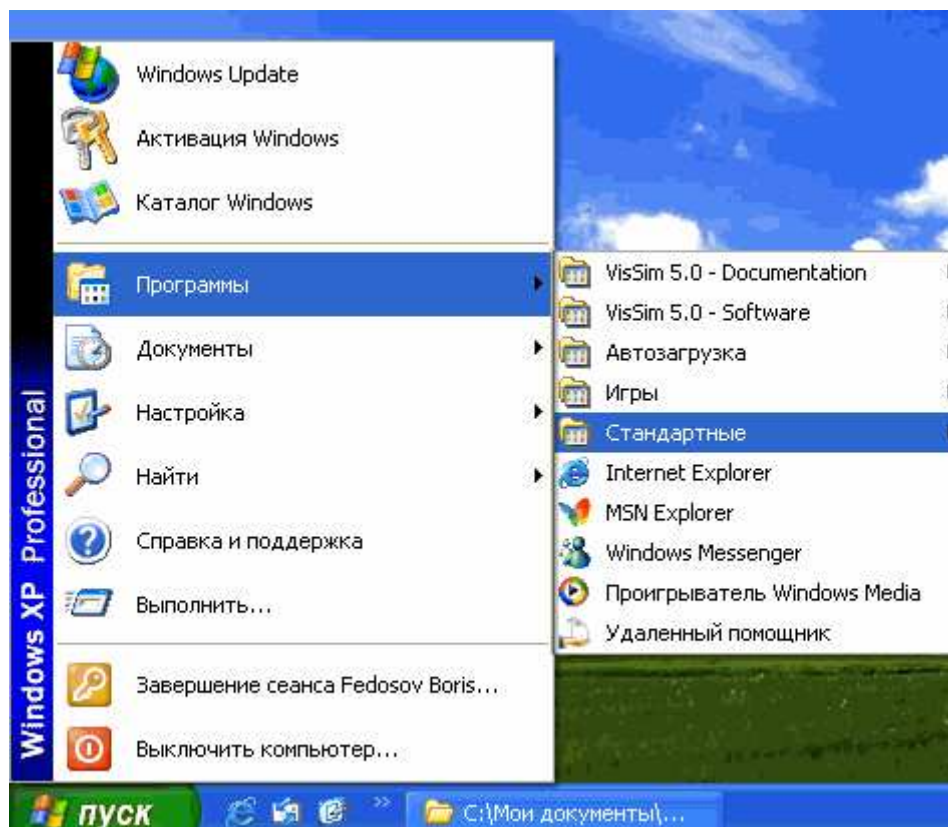
В силу относительной простоты их принято также использовать в качестве учебных. Знание приемов работы со стандартными приложениями способствует освоению специализированных программных средств.

В качестве основных стандартных приложений можно выделить следующие программы:

- Калькулятор – электронный калькулятор;
- Блокнот – текстовый редактор;
- Paint – графический редактор;
- WordPad – текстовый процессор;
- Проводник – приложение, реализующее графический интерфейс доступа пользователя к файлам ОС.

Запуск перечисленных приложений осуществляется через меню Пуск: Пуск → Программы → Стандартные

Программа Калькулятор – компьютерная версия обычного карманного или настольного электронного калькулятора. Отличительная черта – абсолютно безбумажная технология выполнения расчетов за счет возможности выведения на экран неограниченного количества калькуляторов (для параллельного выполнения расчетов, хранения промежуточных вычислений).



Блокнот – это простейший текстовый редактор, предназначенный для создания и редактирования текстовых файлов типа \*.txt и некоторых других, не требующих форматирования и не превышающих по размеру 64 КБайт.

Блокнот удобно использовать:

- для ведения коротких записей или пометок;
- редактирования командных файлов;
- создания web-страниц.

Функциональные возможности:

Строка меню

Рабочая область окна

Полосы прокрутки

- создание и редактирование простейших текстовых документов;
- просмотр текстовых файлов;
- выбор типа шрифта для всего документа, его начертания и размера (Формат →

Шрифт);

-сохранение документа (Файл → Сохранить (сохранение текущей версии файла) или Файл → Сохранить как (сохранение копии с новым именем с выбором папки и диска));

-копирование, вставка, перемещение, удаление выделенных фрагментов текста (Правка → Копировать, Правка → Вставить, Правка → Вырезать, Правка → Удалить);

-автоматическая вставка времени и даты (Правка → Время и дата);

-автоматический переход на новую строку (Формат → Перенос по словам);

-автоматический поиск и замена (Правка → Найти, Правка → Заменить);

-настройка параметров страницы (Файл → Параметры страницы);

-печать документа (Файл → Печать).

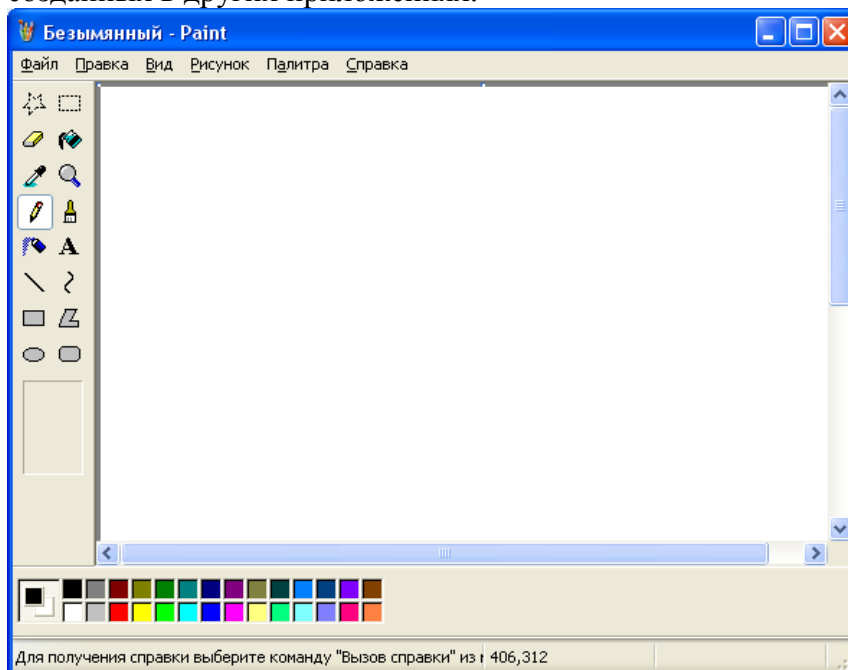
Paint

Paint – средство для рисования, однооконный редактор растровой графики (основной элемент – точка, которой на экране соответствует экранная точка – пиксел),

предназначенный для создания и редактирования графических файлов различной сложности типа \*.bmp, \*.jpg, \*.gif.

Paint можно использовать:

- для создания цветных и черно-белых рисунков, схем, чертежей, диаграмм, надписей;
- использования созданных изображений в качестве фона рабочего стола;
- просмотра и правки отсканированных изображений, а также объектов, созданных в других приложениях.



Функциональные возможности:

- просмотр графических файлов (формат \*.bmp, \*.jpg, \*.gif, \*.tiff);
- создание и редактирование изображений (цветных и черно-белых);
- автоматическое увеличение, уменьшение, поворот выделенных объектов, инвертирование их цвета (Команды меню Рисунок: Отразить/повернуть, Растянуть/Наклонить, Обратить цвета);
- сохранение документа (Файл → Сохранить (сохранение текущей версии файла) или Файл → Сохранить как (сохранение копии с новым именем с выбором папки и диска));
- копирование, вставка, перемещение, удаление выделенных фрагментов рисунка (Правка → Копировать, Правка → Вставить, Правка → Вырезать, Правка → Удалить);
- отмена последнего выполненного действия (Правка → Отменить);
- ввод текстовой информации с возможностью задания типа, размера, начертания и цвета шрифта (Панель инструментов → Надпись );
- задание размера рабочей области документа (Рисунок → Атрибуты → Высота и ширина);
- возможность создания прозрачных рисунков в формате \*.gif, которые при наложении позволяют видеть первый рисунок (Рисунок → Непрозрачный фон);
- увеличение и уменьшение выделенного фрагмента (Панель инструментов → Выделение );
- перемещение документа в другой файл/загрузка рисунка из другого файла (Правка → Копировать в файл, Правка → Вставить из файла);
- выбор параметров страницы (Файл → Параметры страницы);
- предварительный просмотр документа (Файл → Предварительный просмотр);
- получение изображения со сканера или цифровой фотокамеры (Файл → Со сканера или камеры);

-захват изображения с экрана (клавиша PrintScreen копирует в буфер обмена скриншот (screenshot) – графический образ – всего экрана; комбинация клавиш Alt+PrintScreen копирует в буфер обмена активное окно; полученный образ вставить в окно графического редактора Правка → Вставить);  
-печать документа (Файл → Печать).

## 2.4 Лабораторная работа №4 (2 часа).

**Тема:** «Общее управление MS WORD»

**2.4.1 Цель работы:** Изучение информационной технологии создания, сохранения и подготовки к печати документов MS Word.

### 2.4.2 Задачи работы:

1. Изучить технологию создания, сохранения работы с документами MS Word.

### 2.4.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. Персональный компьютер
2. Операционная система Windows

### 2.4.4 Описание (ход) работы:

1. Откройте текстовый редактор Microsoft Word.
2. Установите нужный вид экрана, например – *Разметка страницы* (*Вид/Разметка страницы*).
3. Установите параметры страницы (поля: левое – 3 см, правое – 1,5 см, верхнее – 3 см, нижнее – 1,5 см; ориентация – книжная; размер бумаги – А4;), используя команду *Файл/Параметры страницы* (вкладки *Поля* и *Размер бумаги*) (рис. 1).
4. Установите межстрочный интервал – полуторный; первая строка – отступ, на 1,25 см.; выравнивание – по центру, используя команду *Формат/Абзац* (вкладка *Отступы и интервалы*) (рис. 2).

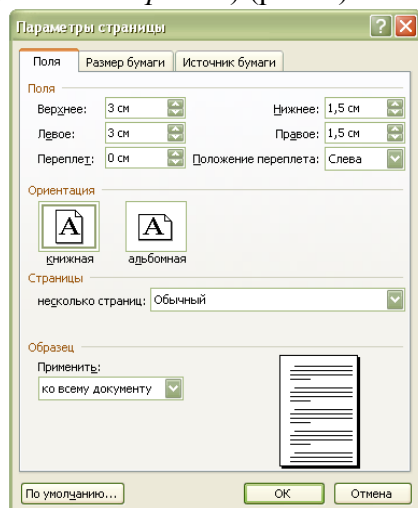


Рис. 1 - Установка параметров страницы



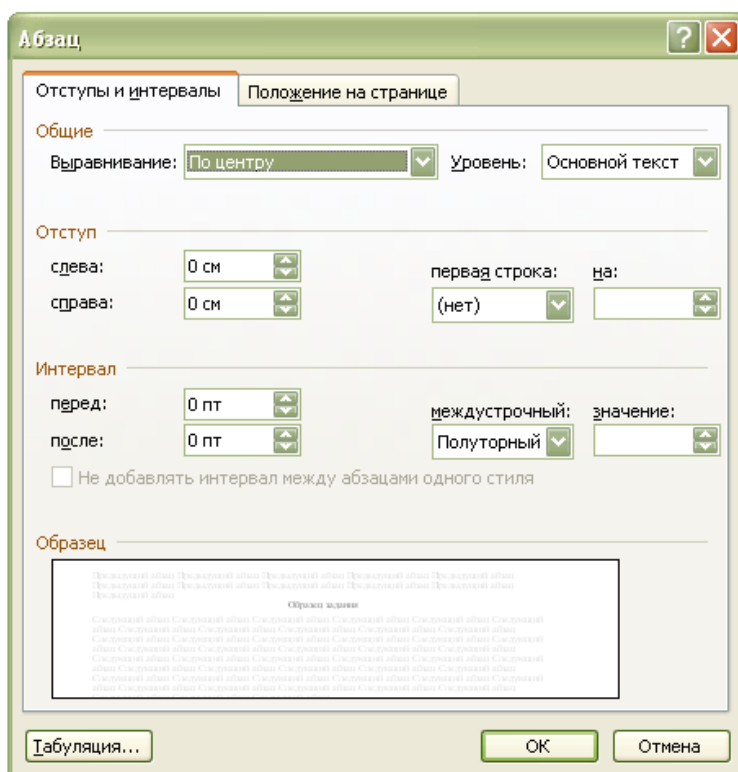


Рис. 2 - Установка параметров абзаца

5. Наберите текст, приведенный ниже (текст можно изменить и дополнить). В процессе набора текста меняйте начертание, размер шрифта (для заголовка – 14 пт., для основного текста – 12 пт.; типы выравнивания абзаца – по центру, по ширине, по правому краю), используя кнопки на панелях инструментов.

#### Образец задания

##### **ПРИГЛАШЕНИЕ**

Уважаемый

господин *Яков Михайлович Орлов!*

Приглашаем Вас на научную конференцию «Информатизация современного общества».

Конференция состоится 20 ноября 2013 г. в 12.00 в конференц-зале Технологического колледжа.

*Ученый секретарь*

*С.Д.Петрова*

6. Заклучите текст приглашения в рамку и произведите цветовую заливку.

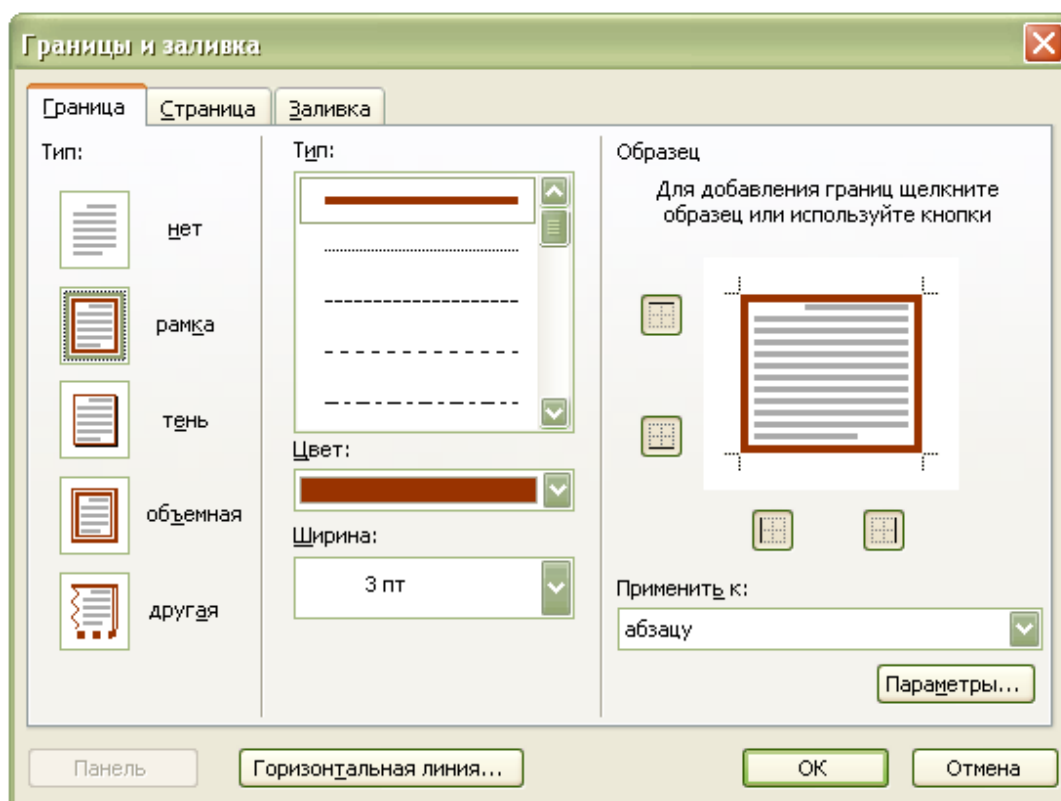


Рис. 3 - Оформление рамки вокруг приглашения

Для этого:

- выделите весь текст приглашения;
  - выполните команду *Формат/Границы и заливка*;
  - на вкладке *Граница* установите параметры границ: тип – рамка; ширина линии – 3пт.; цвет линии – по вашему усмотрению; применить – к абзацу (рис. 3);
  - на вкладке *Заливка* выберите цвет заливки (рис. 4);
  - укажите условие применения заливки – применить к абзацу;
  - нажмите кнопку *ОК*.
7. Вставьте рисунок в текст приглашения (*Вставка/Рисунок/Картинки*); задайте положение текста относительно рисунка – «Вокруг рамки» (*Формат/Рисунок/Положение/Вокруг рамки*).
8. Скопируйте дважды на лист типовое приглашение (*Правка/Копировать*, *Правка/Вставить*).
9. Отредактируйте лист с полученными двумя приглашениями и подготовьте к печати (*Файл/Предварительный просмотр*).
10. Напечатайте приглашения (при наличии принтера), выполнив команду *Файл/Печать* и установив нужные параметры печати (число копий – 1; страницы – текущая).
11. Сохраните файл в папке вашей группы, выполнив следующие действия:
- выполните команду *Файл/Сохранить как...*;
  - в диалоговом окне *Сохранить как...* укажите имя диска, например «С:» и имя папки (например, *Мои документы/Номер группы*); введите имя файла, например «Приглашение»;
  - нажмите кнопку *Сохранить*.

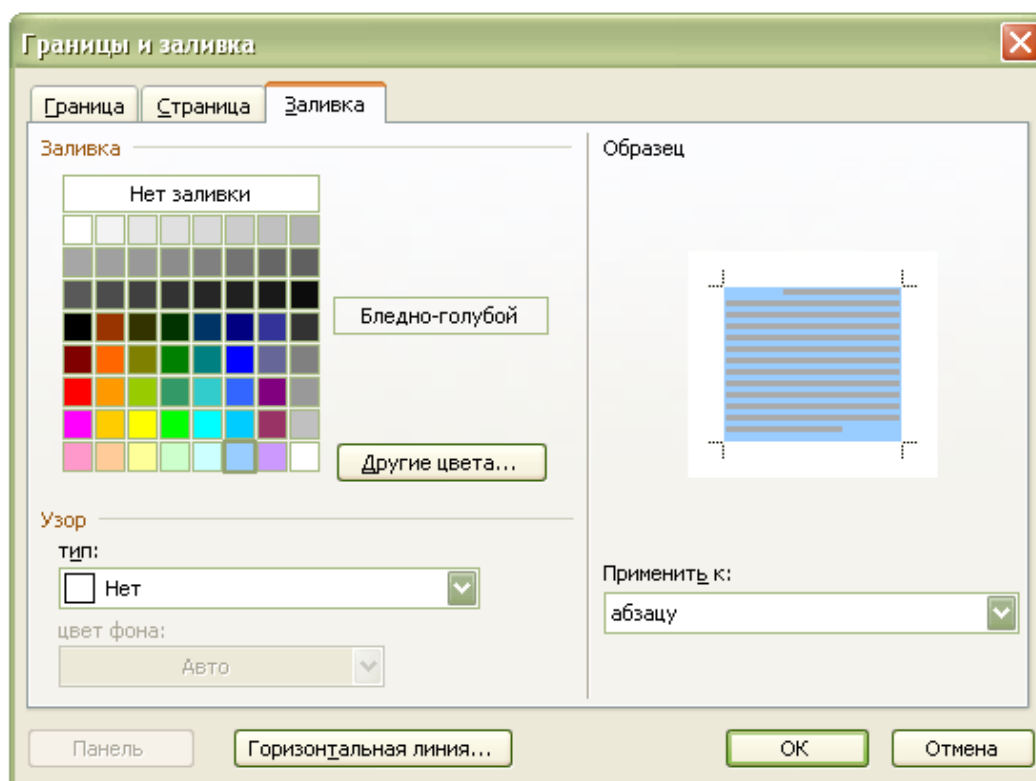


Рис. 4 – Оформление цветовой заливки приглашения

**Задание 1.2.** Оформить докладную записку по образцу.

Краткая справка. Верхнюю часть докладной записки оформляйте в виде таблицы (2 столбца и 1 строка; тип линий – нет границ). Этот прием оформления позволит выполнить разное выравнивание в ячейках таблицы: в левой ячейке – по левому краю, в правой – по центру.

**Образец задания**

Сектор аналитики и экспертизы

Директору Центра ГАНЛ  
Н.С.Петрову

**ДОКЛАДНАЯ ЗАПИСКА**

03.11.2012

Сектор не может завершить в установленные сроки экспертизу проекта маркетингового исследования фирмы «Астра-Н» в связи с отсутствием полных сведений о финансовом состоянии фирмы.

Прошу дать указания сектору технической документации предоставить полные сведения по данной фирме.

Приложение: протокол о некомплектности технической документации фирмы «Астра-Н».

Руководитель сектора  
аналитики и (подпись)  
экспертизы

М.П.Спелов

**Примечания.** После окончания работы закройте все открытые файлы, закройте окно текстового редактора Microsoft Word, после чего завершите работу компьютера (Пуск/Выключить компьютер).

**Задание 1.3.** Оформить рекламное письмо по образцу.

Краткая справка. Верхнюю часть рекламного письма оформляйте в виде таблицы (3 столбца и 2 строки; тип линий – нет границ, кроме разделительной линии

между строками). Произведите выравнивание в ячейках таблицы: первая строка – по центру, вторая строка – по левому краю.

**Образец задания**

МЕЖДУНАРОДНЫЙ ИНСТИТУТ «РАБОТА И УПРАВЛЕНИЕ»		INTERNATIONAL INSTITUTE «WORK & MANAGEMENT»
Россия, 127564, Москва, Ленинский пр., 457, офис 567 Тел./факс: (895) 273- 8585		Office 567, 457, Leninsky pr., Moscow, 127564, Russia phone/fax (895) 273- 8585

## **2.5 Лабораторная работа №5 (2 часа).**

**Тема:** «Структура электронной таблицы»

**2.5.1 Цель работы:** Ознакомиться с основными приемами работы при вводе данных в таблицы.

### **2.5.2 Задачи работы:**

1. Ознакомиться с окном программы EXCEL, с функциями основного меню, с кнопками панелей инструментов Стандартная и Форматирование.
2. Ознакомиться со справочной системой программы EXCEL.
3. Научиться выделять ячейку, группу ячеек (блок) для последующего копирования, перемещения, очистки и ввода данных. Копирование и перемещение выполнить разными способами – через основное меню, через меню по правой кнопке «мыши» и с помощью клавиатуры + «мышь».
4. Научиться выделять лист(ы) для последующего выполнения команд по вставке, удалению, копированию, перемещению и переименованию листов рабочей книги.
5. Ознакомиться с возможностями применения команд Автозамена и Автозаполнение. Создать свой список и выполнить автозаполнение в таблице.
6. Научиться применять команду Формат ячеек для изменения внешнего вида ячеек, а также для представления в возможных форматах произвольного целого и дробного числа в ячейке.
7. В соответствии с вариантом (см.файлы «Варианты заданий\_лр1.doc», «Виды таблиц\_1.doc»...«Виды таблиц\_6.doc») воспроизвести на листе EXCEL таблицу и выполнить ее оформление в соответствии с заданием по варианту.

### **2.5.3 Перечень приборов, материалов, используемых в лабораторной работе:**

1. Персональный компьютер
2. Операционная система Windows

### **2.5.4 Описание (ход) работы:**

#### **ПОРЯДОК ВЫПОЛНЕНИЯ ЗАДАНИЯ**

1. Изучить теоретическую часть работы по файлам, находящимся в каталоге этой лабораторной работы, и справочным материалам по табличному процессору EXCEL.
2. Запустить программу EXCEL.
3. Выполнить пункты 1 – 7 в разделе «Задачи работы».
4. Пользуясь содержимым файла «Указания1\_EXCEL.doc», выполнить действия, указанные в пунктах 3 – 6 раздела «Содержание работы».
5. Выйти из программы EXCEL.

Что нужно знать и уметь делать после выполнения практического задания

1. Как пользоваться справочной системой программы EXCEL.
2. Как выделять ячейку, группу ячеек (последовательно или вразброс).
3. Как вводить данные: текст, числа, формулы.
4. Как копировать, перемещать, очищать ячейку и группу ячеек, как использовать команду Специальная вставка.
5. Как выделять лист(ы) книги.
6. Как производить удаление, добавление, копирование, перемещение и переименование листов рабочей книги.
7. Как применять команды Автозамена и Автозаполнение.
8. Как производить форматирование содержимого ячеек.

Произвести набор Таблицы 1. Для этого в исходной таблице:

- добавить столбцы 4 и 6 после столбцов «До архивации» и «После архивации» по вычислению %;
- добавить столбец 8 для вычисления стандартных функций;
- в столбец 3 занести размеры файлов до архивации, разделив их предварительно на N (где N – номер варианта).

Вычислить процент для объема каждого файла и типа архиваторов.

За 100% взять максимальное значение в подстолбце для одного типа архиваторов.

2. Представить данные столбцов 3 и 5 в разных форматах и разрядности. Оформить границы таблицы различной толщиной линий и раскрасить ячейки, строки(столбцы) различным цветом заливки и шрифта для одного типа архиватора.

3. В таблице произвести суммирование показателей по произвольной строке (столбцу) с помощью Автосуммы и поместить результат в столбец 8.

4. С помощью мастера функций найти среднее, максимальное, минимальное значения по произвольной строке (столбцу), а также те же значения для трех произвольных ячеек из трех типов архиваторов и поместить результат в столбец 8. Название функций оформить в виде комментария к ячейкам, где размещен результат.

5. Из Таблицы 1 получить таблицу вида:

№ по- рядку	Имя файла с расши- рением	Объем файла в Кбайтах		Архиватор	
		до архивации	после архивации		
1	2	3	4	5	6
1.	txt			ZIP	

В таблице произвести сквозную нумерацию всех файлов в первом столбце. Затем произвести сортировку по убыванию для столбца 3 и восстановить последовательную нумерацию в столбце 1. Это будет Таблица 2.

6. В столбце 6 для каждой ячейки применить условное форматирование по отношению к столбцу 4, результатом которого будет изменение фона заливки (или появление определенного символа) столбца 6 по следующим условиям:

А) Если значение ячейки меньше максимального, но больше среднего, то цвет фона – красный.

Б) Если значение ячейки меньше среднего, но больше минимального, то цвет фона –зеленый.

В) В другом случае цвет фона – синий.

7. Далее в Таблице 2 произвести сортировку по возрастанию для столбца 4. Это будет Таблица 3. Сравнить, есть ли изменения в порядке нумерации строк столбца 1 и фона в столбце 6.

## 2.6 Лабораторная работа №6 (2 часа).

**Тема:** «Создание и заполнение таблицы постоянными данными и формулами»

**2.6.1 Цель работы:** Освоить технологию создания и заполнения таблицы постоянными данными и формулами.

### 2.6.2 Задачи работы:

1. Создать таблицу.
2. Заполнить таблицу постоянными данными и формулами

### 2.6.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. Персональный компьютер
2. Операционная система Windows

### 2.6.4 Описание (ход) работы:

#### КРАТКАЯ СПРАВКА

#### Формат ячеек таблицы

Ячейки рабочего листа имеют заданный формат, который устанавливается командой **Формат, Ячейки** или командой контекстного меню **Формат ячеек**. Эти команды имеют несколько вкладок: *Число, Выравнивание, Шрифт, Граница, Вид, Защита* (рис. 1).

- Вкладка *Число* — задает форматы представления данных в ячейке;
- *Общий* — обеспечивает отображение числовых и текстовых данных произвольного типа;
- *Числовой* — включает цифры и символы-разделители: десятичная точка, процент, знак мантииссы, знак числа, круглые скобки, денежное обозначение (р. или \$);
- *Денежный* или *Финансовый* — для отображения денежных величин;
- *Дата/время* — для отображения даты и времени в выбранном формате;
- *Процентный* — для вывода чисел, предварительно умноженных на 100, с символом процента;
- *Дробный* — для вывода дробных чисел;
- *Экспоненциальный* — для вывода чисел в экспоненциальном формате, например 1,65E+044;
- *Текстовый* — последовательность букв, цифр, специальных символов;
- *Дополнительный* — нестандартные дополнительные форматы, например номер телефона, почтовый индекс и пр.
- *Все форматы* — показывает все имеющиеся в Excel форматы.
- Вкладка *Выравнивание* определяет:
- *Выравнивание* — способ выравнивания данного в ячейке по горизонтали (по левому или правому краю, по значению, по центру выделения, по центру, по ширине, с заполнением) или по вертикали (по нижнему или верхнему краю, по центру или высоте);
- *Отображение* — определяет, можно ли переносить в ячейке текст по словам, разрешает или запрещает объединение ячеек, задает автоподбор ширины ячейки.
- Вкладка *Шрифт* — изменяет шрифт, начертание, размер, цвет, подчеркивание и эффекты текста в выделенных ячейках;
- Вкладка *Граница* — создает рамки (обрамление) вокруг выделенного блока ячеек;
- Вкладка *Вид* — позволяет задать закрашку ячейки (цвет и узор);
- Вкладка *Защита* — управляет скрытием формул и блокировкой ячеек (запрет редактирования данных ячеек). Устанавливать защиту можно в любой момент, но действовать она будет только после того, когда введена защита листа или книги с помощью команды **Сервис, Защитить лист**.

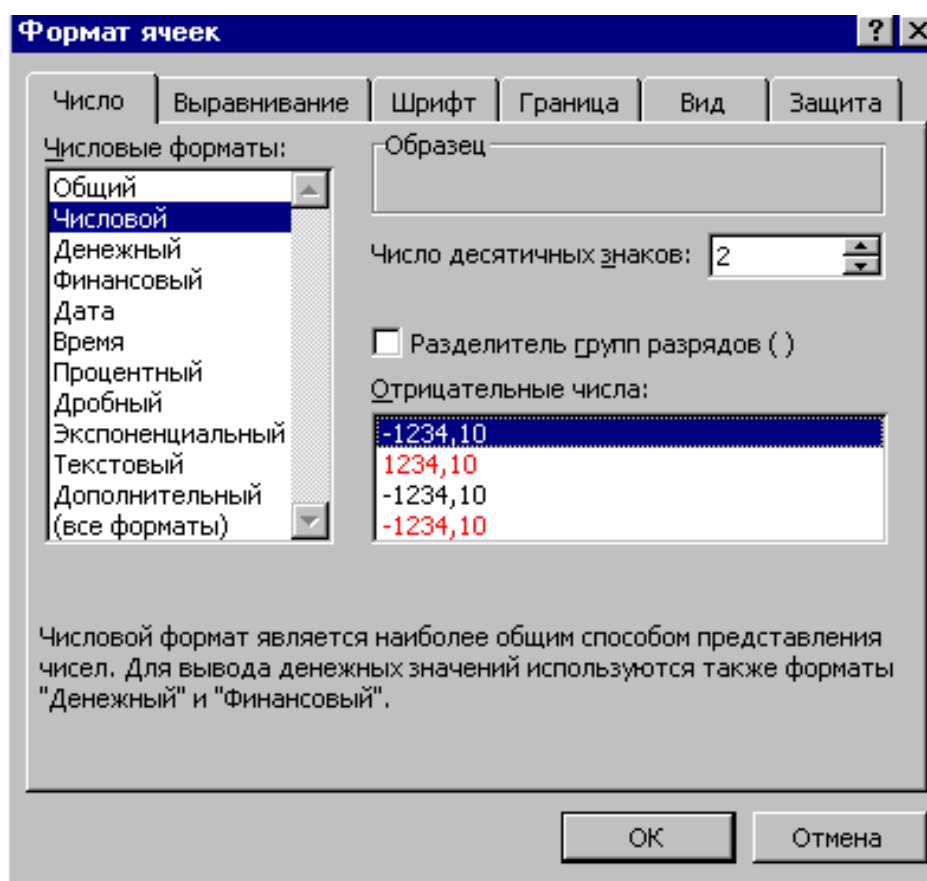


Рис. 1 - Вкладка *Число* диалогового окна <<Формат ячеек>> для выбора формата представления данных в таблице

### Технологические операции с ячейками таблиц

В операциях обработки часто используется не отдельная ячейка, а блок ячеек. *Блок* — прямоугольная область смежных или несмежных ячеек, расположенных в разных местах. Блоком ячеек можно считать и строку, и столбец. Типовые технологические операции с блоками ячеек представлены в табл. 2.

Применительно к ячейке и блоку ячеек выполняются следующие действия:

форматирование	перемещение *
копирование *	заполнение
вставка *	очистка форматов, содержимого, значений
удаление *	

### **Внимание!**

1. Первоначально выделяется блок ячеек — объект действия, а затем выбирается команда меню для исполнения действия.
2. При выделении блока несмежных ячеек необходимо предварительно нажать и удерживать клавишу <Ctrl>.
3. Помеченные (\*) действия не выполняются для блока с несмежными ячейками.

Таблица 2. Типовые технологические операции с блоками ячеек


п/п	Название технологической операции	Технология выполнения операции с помощью управляющего меню	Альтернативный вариант технологии с помощью контекстного меню или мыши
-----	-----------------------------------	------------------------------------------------------------	------------------------------------------------------------------------




	Выделение блока смежных ячеек	-	1. Установить курсор в ячейку, начиная с которой выполняется выделение. 2. Нажать левую кнопку мыши. 3. Протащить курсор, закрашивая область выделения.
	Выделение блока несмежных ячеек	-	1. Выделить блок смежных ячеек. 2. Выделить следующий блок смежных ячеек при нажатой клавише <Ctrl>.
	Форматировать блок ячеек	1. Выделить блок ячеек. 2. Выполнить команду <b>Формат, Ячейки</b>	1. Вызвать контекстное меню. 2. Команда <b>Формат ячеек</b> .

***Продолжение***

п/п	Название технологической операции	Технология выполнения операции с помощью управляющего меню	Альтернативный вариант технологии с помощью контекстного меню или мыши
	Удалить блок (изменение структуры таблицы)	1. Выделить блок ячеек. 2. Команда <b>Правка, Удалить</b> . 3. Указать объект удаления (строки, столбцы или блок ячеек со сдвигом ячеек влево или вверх).	1. Вызвать контекстное меню. 2. Команда <b>Удалить</b> .
	Вставить блок ячеек (строк, столбцов)	1. Выделить блок ячеек. 2. Выполнить команду <b>Вставка, Ячейки</b> (указать смещение — вправо или вниз, Добавление строк или столбцов).	1. Вызвать контекстное меню. 2. Выполнить команду <b>Добавить ячейки</b> .
	Копировать блок ячеек.	1. Выделить блок ячеек. 2. Команда <b>Правка, Копировать</b> . 3. Установить курсор в ячейку, куда копируется. 4. Команда <b>Правка, Вставить</b> .	1. Вызвать контекстное меню. 2. Команда <b>Копировать</b> . 3. Установить курсор в ячейку, куда копируется. 4. Команда <b>Правка, Вставить</b> .

	Вставить блок ячеек (предварительно выполнена команда копирования или вырезания)	1. Установить курсор в место вставки. 2. Команда <b>Правка, Вставить.</b>	1. Вызвать контекстное меню. 2. Выполнить команду <b>Вставить.</b>
	Вставить блок ячеек с размножением (предварительно выполнена команда копирования или вырезания)	1. Установить курсор в место вставки. 2. Выделить блок, кратный исходному блоку. 3. Команда <b>Правка, Вставить.</b>	1. Выделить блок, кратный исходному. 2. Вызвать контекстное меню. 3. Выполнить команду <b>Вставить.</b>
	Очистить блок	1. Выделить блок ячеек. 2. Выполнить команду <b>Правка, Очистить.</b> 3. Указать объект обработки: <i>все, форматы, содержимое, примечание.</i>	1. Вызвать контекстное меню. 2. Выполнить команду <b>Очистить содержимое.</b>
0	Перенести выделенный блок с помощью мыши	-	1. Выделить блок ячеек (блок строк или блок столбцов). 2. Установить курсор мыши на любую линию контура. (он примет форму  ). 3. Нажать левую кнопку мыши и, не отпуская ее, переместить блок на другое место.

Продолжение

п/п	Название технологической операции	Технология выполнения операции с помощью управляющего меню	Альтернативный вариант технологии с помощью контекстного меню или мыши
1	Скопировать выделенный блок с помощью мыши	-	<ol style="list-style-type: none"> <li>1. Выделить блок ячеек.</li> <li>2. Установить курсор мыши на любую линию контура.</li> <li>3. Нажать клавишу &lt;Ctrl&gt;.</li> <li>4. Нажать левую кнопку мыши (при этом рядом со  появится +) и, не отпуская ее, переместить блок в место вставки.</li> <li>5. Отпустить клавишу &lt;Ctrl&gt;.</li> </ol>
2	Заполнение блока значениями	<ol style="list-style-type: none"> <li>1. Ввести значения в начало или в конец блока для тиражирования.</li> <li>2. Выделить блок ячеек (вниз, вверх, вправо или влево от начальной ячейки) для заполнения.</li> <li>3. Выполнить команду <b>Правка, Заполнить</b>.</li> <li>4. Выбрать вариант заполнения: <i>вниз, вверх, вправо, влево, прогрессия</i> в зависимости от конфигурации блока</li> </ol>	-
3	Заполнение блока ячеек при копировании с помощью мыши	-	<ol style="list-style-type: none"> <li>1. Ввести значения или формулы в смежные ячейки.</li> <li>2. Выделить блок заполненных ячеек.</li> <li>3. Установить курсор в нижний правый угол блока, добившись появления черного крестика.</li> <li>4. Нажать правую кнопку мыши и протянуть курсор на требуемый размер.</li> <li>5. Выполнить команду контекстного меню</li> </ol>

			<b>Копировать ячейки.</b>
4	Заполнение блока ячеек значениями с помощью мыши	-	<p>1. Ввести значения или формулы в смежные ячейки.</p> <p>2. Выделить блок заполненных ячеек.</p> <p>3. Установить курсор в нижний правый угол блока, добившись появления черного крестика.</p> <p>4. Нажать правую кнопку мыши и протянуть курсор на требуемый размер.</p> <p>5. Выполнить команду контекстного меню <b>Заполнить значения</b></p>

*Продолжение*

п/п	Название технологической операции	Технология выполнения операции с помощью управляющего меню	Альтернативный вариант технологии с помощью контекстного меню или мыши
5	Заполнение блока ячеек списком значений (ряды) с помощью мыши	-	<ol style="list-style-type: none"> <li>1. Ввести значения в смежные ячейки.</li> <li>2. Выделить блок заполненных ячеек.</li> <li>3. Установить курсор в нижний правый угол блока, добившись появления черного крестика.</li> <li>4. Нажать правую кнопку мыши и протянуть курсор на требуемый размер.</li> <li>5. Выполнить команду контекстного меню <b>Заполнить</b>.</li> </ol>
6	Заполнение блока ячеек форматами с помощью мыши	-	<ol style="list-style-type: none"> <li>1. Настроить формат смежных ячеек.</li> <li>2. Выделить блок ячеек с заполненными форматами.</li> <li>3. Установить курсор в нижний правый угол блока, нажать правую кнопку мыши и протянуть курсор на требуемый размер.</li> <li>4. Выполнить команду контекстного меню <b>Заполнить форматы</b>.</li> </ol>
7	Заполнение блока ячеек значениями согласно прогрессии	<ol style="list-style-type: none"> <li>1. Ввести начальное значение прогрессии в ячейку.</li> <li>2. Выделить блок ячеек для заполнения.</li> <li>3. Выполнить команду <b>Правка, Заполнить, Прогрессия</b>.</li> <li>4. Указать тип и параметры прогрессии.</li> </ol>	<ol style="list-style-type: none"> <li>1. Установить курсор в нижний правый угол начальной ячейки, нажать правую кнопку мыши и протянуть курсор на требуемый размер.</li> <li>2. Выполнить команду контекстного меню <b>Прогрессия</b>.</li> <li>3. Указать тип и параметры прогрессии.</li> </ol>
8	Скрыть блок строк (столбцов)	-	<ol style="list-style-type: none"> <li>1. Установить курсор на строке с именами столбцов или на столбце с номерами строк.</li> <li>2. Выделить блок строк (столбцов).</li> <li>3. Выполнить</li> </ol>

			команду контекстного меню <b>Скрыть.</b>
--	--	--	---------------------------------------------

Продолжение

п/п	Название технологической операции	Технология выполнения операции с помощью управляющего меню	Альтернативный вариант технологии с помощью контекстного меню или мыши
9	Показать блок скрытых строк (столбцов)	<p>1. Если первый столбец (строка) является скрытым, выполните команду <b>Правка, Перейти</b>,</p> <p>2. В диалоговом окне в поле <i>Ссылка</i> введите A1 (латинская буква) и нажмите кнопку &lt;ОК&gt;.</p> <p>3. Выполните команду <b>Формат, Столбец</b> или <b>Строка, Отобразить</b>.</p>	<p>1. Выделить блок из двух столбцов (строк), между которыми находятся скрытые столбцы (строки). (Курсор устанавливается на строке с именами столбцов или на столбце с номерами строк.).</p> <p>2. Выполнить команду контекстного меню <b>Отобразить</b>.</p>
0	Построение экстраполяционных (прогнозных) рядов (т.е. расчет будущих значений в свободных ячейках на основании уже введенных значений)	-	<p>1. Ввести несколько значений в смежные ячейки и выделить этот блок.</p> <p>2. Установить курсор в нижний правый угол блока, добившись появления крестика, нажать правую кнопку мыши и протянуть курсор на требуемый размер.</p> <p>3. Выполнить команду контекстного меню <b>Линейное</b> или <b>экспоненциальное приближение</b>.</p>
1	Присвоение имени блоку ячеек	<p>1. Выделить блок ячеек.</p> <p>2. Команда <b>Вставка, Имя, Присвоить</b>.</p> <p>3. Указать имя блока ячеек, начинающееся с буквы.</p>	-
2	Присвоение блоку ячеек имени, находящегося в столбце или строке	<p>1. Выделить блок ячеек.</p> <p>2. Выполнить команду <b>Вставка, Имя, Создать</b>.</p> <p>3. Указать источник имени: верхняя или нижняя строка, левый</p>	-

		или правый столбец.	
--	--	---------------------	--

### Заполнение таблицы постоянными значениями

В ячейки рабочего листа вводятся два вида данных: постоянные значения (константы) и формулы.

*Постоянные значения* — это числа, символы, текст. В ячейку константа записывается следующим образом: курсор устанавливается в ячейку и с клавиатуры вводится значение. Редактирование введенного значения проводится после установки курсора в нужную ячейку, а далее следует либо нажать клавишу <F2>, либо щелкнуть кнопкой мыши в строке ввода в нужном месте.

*Формулы* вводятся и редактируются аналогично. Однако прежде чем их вводить, надо разобраться с правилами их формирования.

#### Формулы в таблице и технология их использования

Под *формулой* в электронной таблице понимают выражение, состоящее из операндов и операций. Формулы строятся как выражение для вычисления нового значения. Тип значения, полученного в результате вычисления по формуле, определяется типом операндов выражения. Формула всегда начинается с символа равно (=).

В качестве *операндов* используются:

- числа;
- тексты (вводятся в двойных кавычках, например «Неявка»);
- логические значения (например, ИСТИНА и ЛОЖЬ, условия типа A23=A45 и т.д.);
- значения ошибки (типа #ДЕЛ/0!, #Н/Д, #ИМЯ?, #ПУСТО!, #ЧИСЛО!, #ССЫЛКА! и #ЗНАЧ!);
- ссылки — адреса ячеек. При перечислении ссылки разделяются точкой с запятой, например: A4; C5; C10: E20;
- встроенные функции Excel 97.

**Примечание.** Альтернативный стиль ссылок R1C1 не рассматривается

Операнды в формулах соединяются с помощью символов *операций*:

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

Формулы можно копировать в другие ячейки. При этом в зависимости от типа ссылок, входящих в копируемую формулу, осуществляется их настройка: автоматическая (для относительных ссылок) или полуавтоматическая (для частично абсолютных ссылок). Различают следующие типы ссылок:

- *относительные* ссылки, например A2 или C23, которые всегда изменяются так, чтобы отобразить правило их вхождения в формулу относительно ее нового местоположения. При копировании формулы в новую книгу и лист перед ссылкой, входящей в скопированную формулу, появляется имя книги и листа, откуда производилось копирование (STAR\ЛИСТ5!F4);

- *абсолютные* ссылки, которые перед именем столбца и номером строки имеют символ \$. Назначение ссылки абсолютной производится следующим образом: в строке ввода перед ссылкой устанавливается курсор и нажимается клавиша <F4>, например \$A\$4. Можно сделать то же самое, вводя символ \$ с клавиатуры. При копировании абсолютные ссылки остаются неизменными;

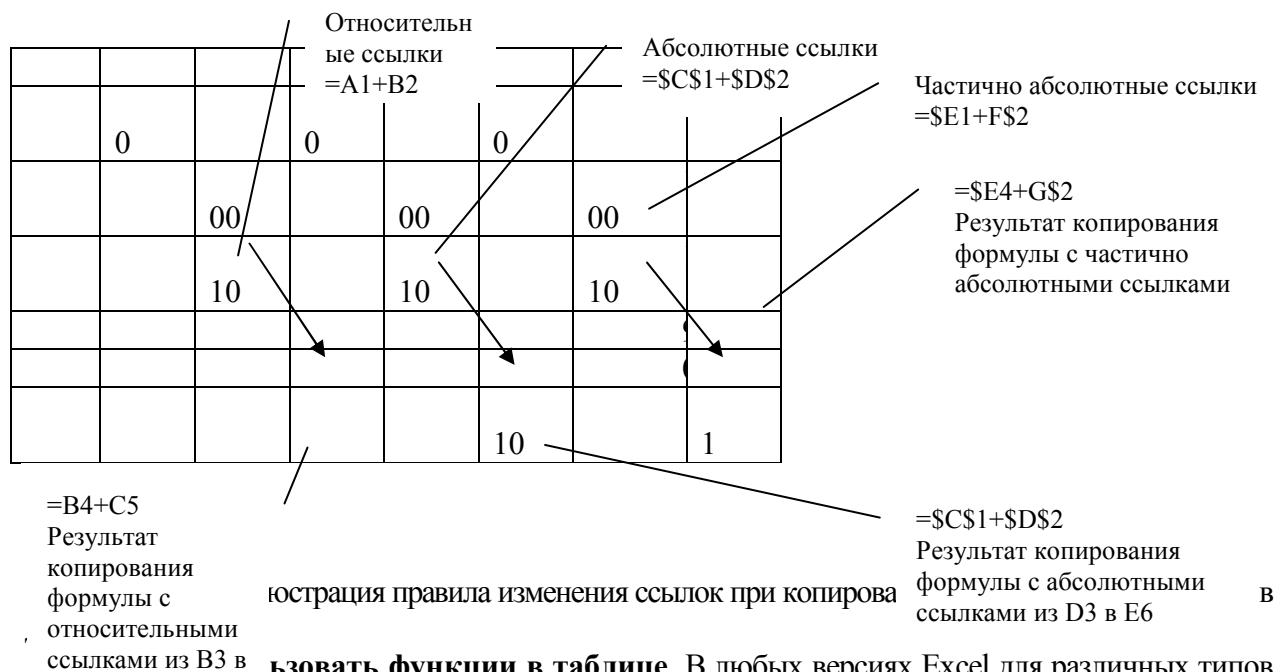
- *частично абсолютные* ссылки, которые при копировании корректируются частично. Символ \$ стоит или перед именем столбца, или перед номером строки (\$R2, F\$5). Например, при копировании формулы, содержащей \$F5, сохранится имя столбца F, а номер строки будет изменен;



■ *имена блоков*, например *ЦЕНА*. Имя связывается с данными блока, а не с его местоположением. Можно блок перенести в другое место, что не повлияет на его имя.

Формулы можно копировать в другие ячейки. При этом в зависимости от типа ссылок, входящих в копируемую формулу, осуществляется их корректировка: автоматическая (для относительных ссылок) или полуавтоматическая (для частично абсолютных ссылок).

**Пример.** На рис. 4 представлен результат копирования формул из одной ячейки в другую для трех вариантов ссылок: относительных, абсолютных, частично абсолютных.



**Использовать функции в таблице.** В любых версиях Excel для различных типов вычислений имеется большое число встроенных функций: математических, статистических, логических, текстовых, информационных и др.


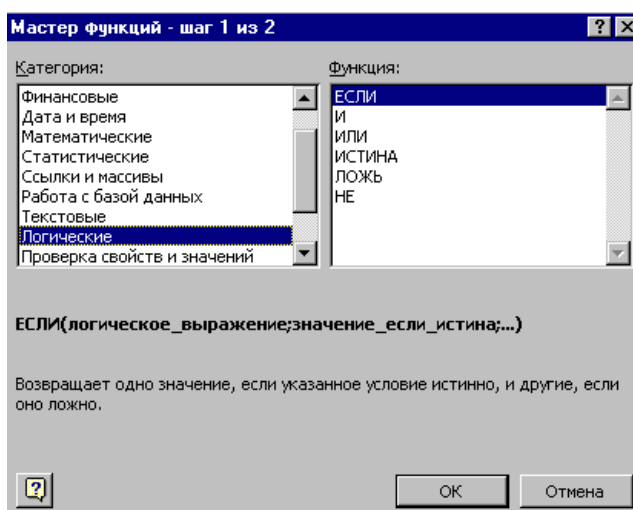
Функции вводятся обычным набором с клавиатуры или более предпочтительным способом — с помощью *Мастера функций* (рис.5), диалоговое окно которого вызывается командой **Вставка, Функция** или кнопкой .

Рис. 5. Диалоговое окно <<Мастер функций>>



для выбора категории и вида функции

Все функции разделены на категории, каждая из которых включает в себя определенный набор функций.

Для каждой категории функций справа в окне (см. рис. 5) показан их состав. Выбирается категория функция (слева), имя функции (справа), внизу дается краткий синтаксис функции. Если функция использует несколько однотипных аргументов, указан символ многоточия (...).

После нажатия кнопки <ОК> появляется следующее диалоговое окно (пример окна приведен на рис. 6) и осуществляется построение функции, т.е. указание ее аргументов. Каждый аргумент вводится в специально предназначенную для него строку, например, так, как показано на рис. 6.

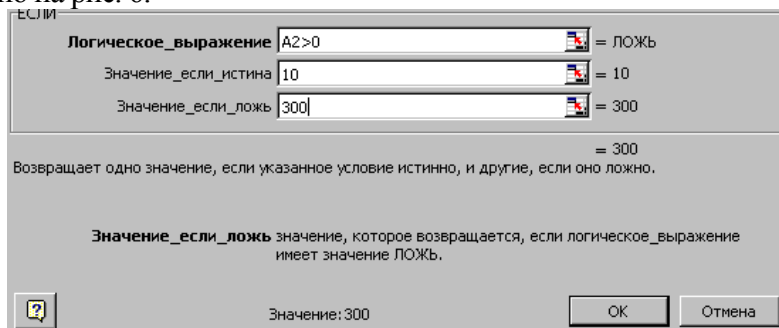


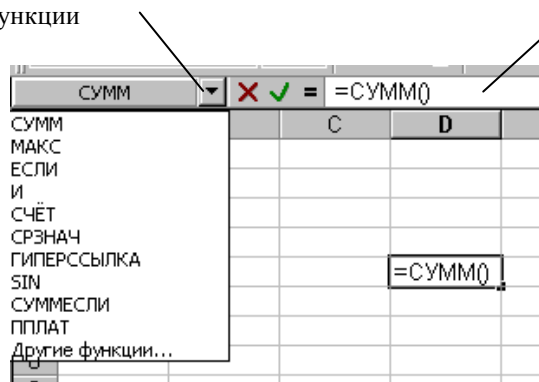


Рис. 6. Пример диалогового окна для задания аргументов логической функции *ЕСЛИ*

Кнопка  позволяет увеличить масштаб строки ввода аргумента, а кнопка  обеспечивает возврат в диалоговое окно на строку ввода.

Формулы вводятся в ячейку. Для вставки в формулу других функций в строке ввода, которая находится в верхней части окна над рабочим полем (рис. 7), предусмотрена кнопка вызова функций.

Кнопка вызова функции



Строка ввода

Рис. 7. Использование кнопки вызова функции в строке ввода

Правила построения формул с помощью *Мастера функций*:

- состав аргументов функций, порядок задания и типы значений фиксированы и не подлежат изменению;
- аргументы вводятся в специальных строках ввода, например, так, как изображено на рис. 6;
- для формирования аргумента, как результата промежуточного вычисления по функциям, нажимается кнопка вызова функций в строке ввода (рис. 7); глубина вложенности — произвольная;
- для ввода имени блока ячеек используется команда **Вставка, Имя, Вставить** с выбором имени блока;
- для построения ссылки следует установить курсор в поле ввода, а затем перевести указатель мыши на требуемый рабочий лист для выделения ячейки или блока;
- абсолютные ссылки формируются при установке курсора перед адресом ячейки в строке ввода и нажатии клавиши <F4>.

**ЗАДАНИЕ:** Сформируйте структуру таблицы и заполните ее постоянными значениями — числами, символами, текстом. В качестве примера таблицы

рассматривается экзаменационная ведомость (рис. 8). Для каждой группы создаются типовые ведомости, которые содержат списки студентов (фамилия, имя, отчество, № зачетной книжки) и полученные ими оценки на экзамене.

В данном задании требуется на базе созданной в работе 1 таблицы подготовить для каждой группы электронную экзаменационную ведомость (см. рис. 8).

### ЭКЗАМЕНАЦИОННАЯ ВЕДОМОСТЬ

Группа № \_\_\_\_\_ Дисциплина \_\_\_\_\_

п/п	Фамилия, имя, отчество	№ зачетной книжки	Оц енка	Подпись экзаменатора

“отлично” \_\_\_\_\_

“хорошо” \_\_\_\_\_

“удовлетворительно” \_\_\_\_\_

“неудовлетворительно” \_\_\_\_\_

“неявки” \_\_\_\_\_

ИТОГО \_\_\_\_\_

**Рис. 8.** Форма экзаменационной ведомости для задания 1

В любой таблице всегда можно выделить минимум две структурные части — название и ее шапку.

*Название таблицы* вводится в любую ячейку и оформляется шрифтами. Формирование *шапки таблицы* рекомендуется проводить в следующей последовательности:

- задайте способ выравнивания названия граф (при больших текстах необходимо обеспечить перенос по словам);

- в каждую ячейку одной строки введите названия граф таблицы;

- установите ширину каждого столбца таблицы.

После окончания оформления шапки таблицы введите в таблицу *постоянные данные*:

- фамилии студентов и полученные ими оценки по конкретной дисциплине;

- заголовки в нижней части таблицы для итоговых данных, которые будут подсчитаны впоследствии при выполнении задания 2.

После окончания работы по заполнению ведомости постоянными данными запомните ее как рабочую книгу.

Для лучшего понимания технологии работы в Excel выполните тренинг. Для этого проделайте все операции, указанные в табл. 2.

**Примечание.** Все расчеты в экзаменационной ведомости будут производиться в задании 2.

## **2.7 Лабораторная работа №7 (2 часа).**

**Тема:** «Обеспечение автоматизированного рабочего места»

**2.7.1 Цель работы:** Ознакомиться с обеспечением автоматизированного рабочего места.

### **2.7.2 Задачи работы:**

1. Изучить виды обеспечения автоматизированного рабочего места

### **2.7.3 Перечень приборов, материалов, используемых в лабораторной работе:**

1. Персональный компьютер
2. Операционная система Windows

### **2.7.4 Описание (ход) работы:**

Функционирование автоматизированного рабочего места требует разработки всех видов обеспечения: информационного, математического, программного, лингвистического, технологического, организационного, эргономического и правового.

Информационное обеспечение предусматривает организацию его информационной базы, регламентирует информационные связи и предопределяет состав и содержание всей системы информации. Первоочередной задачей при этом является организация внутримашинной информационной базы; выбор необходимого состава показателей, способа их организации, методов группировки и выборки необходимых данных, определение вида магнитных носителей, организацию активной и пассивной информации, организацию справочной и комментирующей информации, разработку макетов для упрощения ввода информации и выбора необходимых функций.

Математическое обеспечение представляет собой совокупность алгоритмов, обеспечивающих ввод, контроль, хранение и корректировку информации; формирование результатной информации и оформление ее в виде таблиц, графиков, диаграмм; обеспечение достоверности и защиты информации. Наиболее целесообразно организовывать математическое обеспечение по модульному принципу, выделяя типовые и стандартные, многократно повторяющиеся процедуры. Математическое обеспечение служит основой для разработки комплекса программных средств, в связи, с чем его качество должно быть высоким и оно должно согласовываться с потенциальным пользователем автоматизированного рабочего места.

Программное обеспечение подразделяется на общее и специальное. Основные элементы общего программного обеспечения обычно поставляются вместе с персональной ЭВМ. К ним относятся: операционные системы и оболочки, программные средства ведения баз данных, программные средства организации диалога, а также программы, расширяющие возможности операционной системы. Специальное программное обеспечение состоит из уникальных программ и функциональных пакетов прикладных программ. Именно от специального программного обеспечения зависят вид, содержание и конкретная специализация рабочего места. Учитывая, что специальное программное обеспечение, в конечном счете, определяет область применения автоматизированных рабочих мест и состав решаемых пользователем задач, оно должно создаваться на основе инструментальных программных средств диалоговых систем, ориентированных на решение конкретного класса задач со схожими функционально-технологическими особенностями обработки информации.

Лингвистическое обеспечение включает языки общения с пользователем - языки запросов, информационно-поисковые языки, языки-посредники в сетях. Языковые средства можно разделить по видам диалога. Средства поддержки диалога определяют те языковые конструкции, знание которых необходимо пользователю. В одном автоматизированном рабочем месте может быть реализовано несколько типов диалога.

Технологическое обеспечение представляет собой некоторую четко установленную совокупность проектных решений, определяющих последовательность операций, процедур и этапов обработки в соответствующей сфере деятельности пользователя.

Организационное обеспечение включает комплекс документов, регламентирующих деятельность специалистов при использовании персональных вычислительных машин или терминала на их рабочем месте.

Методическое обеспечение состоит из методических указаний, рекомендаций и положений по внедрению, эксплуатации и оценке эффективности функционирования автоматизированных рабочих мест. Оно включает в себя также организованную машинным способом справочную информацию об автоматизированном рабочем месте в целом и отдельных его функциях, средства обучения работе, демонстрационные и рекламные примеры.

Эргономическое обеспечение представляет собой комплекс мероприятий, выполнение которых должно создавать максимально комфортные условия для использования рабочего места специалистами, для быстреего освоения технологии работы и обеспечения качественной работы.

Правовое обеспечение включает систему нормативно-правовых документов, которые должны четко определять права и обязанности специалистов в условиях функционирования рабочего места.

## **2.8 Лабораторная работа №8 (2 часа).**

**Тема:** «Электронный офис»

**2.8.1 Цель работы:** Ознакомится с понятием «Электронный офис»

**2.8.2 Задачи работы:**

1. Изучить назначение электронного офиса.
2. Изучить функции электронного офиса.

**2.8.3 Перечень приборов, материалов, используемых в лабораторной работе:**

1. Персональный компьютер
2. Операционная система Windows

**2.8.4 Описание (ход) работы:**

Электронный офис – это инструмент для оптимизации наиболее трудоемких бизнес-процессов, который позволяет работать с внутренней и внешней информацией компании в едином информационном пространстве. Электронный офис—это единственное готовое SaaS -решение на платформе Microsoft Sharepoint, которая соответствует требованиям ГОСТ Р 53898-2010 Системы электронного документооборота. Взаимодействие систем управления документами. Требования к электронному сообщению.

Функции Электронного офиса:

Менеджмент поручений

Электронный документооборот

Управление взаимоотношениями с клиентами

Управление проектами.

Автоматизация работы с платежами

Корпоративный портал

Service Desk

Электронный документооборот в «Электронном офисе»

Централизованное хранилище электронных документов, возможность доступа к документам из любого места, где есть Интернет. Для каждого документа предусмотрены карточки, которые позволяют хранить его реквизиты.

Связь документов с контрагентами и основной деятельностью компании: заказами, проектами, контрагентами, договорами и пр. Вся связанная с документом информация отображается на карточке документа.

Контроль изменений (версий документа) и совместная работа с документами. Над одним документом могут одновременно работать несколько сотрудников компании.

Гибкий механизм разделения прав доступа к документам и папкам.

Гибкий механизм подписки на изменения документа: сотрудники могут подписываться на изменения только тех документов, которые им интересны.

Механизм маршрутизации, который поддерживает согласование, регистрацию (при необходимости), рассмотрение документа, его исполнение. По любому документу на любой стадии его жизненного цикла можно создать поручение. Поддерживаются маршруты параллельного, последовательного или последовательно-параллельного согласования и рассмотрения документов. Возможна настройка формата регистрационных номеров документов.

Механизм обработки входящих документов позволяет направлять сканированные документы сразу в хранилище документов Электронного офиса.

Полнотекстовый поиск по документам, который позволяет находить документы не только по краткому содержанию, номеру, типу, автору и т.п., но и по внутреннему содержанию документа.

Возможность обсуждения документов с помощью комментариев.

Гибкая система поиска по аналогии с современными поисковыми системами; поиск возможен следующими способами: по ключевым словам, по номеру документа, по типу

информации. Сортируются результаты поиска по релевантности документа, названию, дате изменения.

Преимущества Электронного офиса:

Функциональность: только самые важные и нужные для бизнеса функции;

Быстрота внедрения и начала работы благодаря наличию понятных инструкций и сценариев работы. За счет возможности использования Электронного офиса в режиме SaaS, внедрение занимает 2-3 дня.

Интеграция с любыми учетными системами (справочники, финансовая информация), со всеми продуктами Microsoft Office, с любыми почтовыми программами, что позволяет Электронному офису органично встраиваться в деятельность компании, не меняя привычного хода работы сотрудников.

Привычная работа с файлами в режиме проводника Windows;

Единый справочник контрагентов и синхронизация со справочником контактов в Outlook;

Работа с версиями документов, согласование, выдача поручений прямо в программах MS Office;

Готовая статистическая отчетность по задачам, документам;

Безопасность данных – гибкая система предоставления прав доступа;

Возможность оффлайн работы с данными.

## **2.9 Лабораторная работа №9 (2 часа).**

**Тема:** «Локальные вычислительные сети»

**2.9.1 Цель работы:** Изучить порядок и правила использования сетевых средств ОС Windows NT.

### **2.9.2 Задачи работы:**

1. Приобрести навыки в работе по навигации и использованию сетевых ресурсов ОС Windows NT.

2. Приобрести навыки в работе по запуску и использованию сетевых приложений Windows.

### **2.9.3 Перечень приборов, материалов, используемых в лабораторной работе:**

1. Персональный компьютер
2. Операционная система Windows

### **2.9.4 Описание (ход) работы:**

1. Задание на практическое занятие

1.1. Обслуживание ресурсов ЛВС

- запуск утилиты “Сеть” панели управления;
- изучение интерфейса и настройка параметров ЛВС с помощью утилиты “Сеть”;
- работа с файловой структурой на сетевом ресурсе с использованием программы *Сетевое окружение*;

– работа с файловой структурой на сетевом ресурсе с использованием программы *Проводник*;

– работа с файловой структурой на сетевом ресурсе с использованием программы *Windows Commander*;

- установка специальных разрешений доступа.

1.2. Решение сетевых задач

- запуск сетевого приложения из файловой структуры;
- создание ярлыка на рабочем столе для запуска сетевого приложения;
- создание ярлыка в меню кнопки *Пуск* для запуска сетевого приложения;
- создание документа с помощью сетевого приложения *MS Word*;
- совместное использование сетевого документа приложения *MS Excel*.

2. Методические указания

2.1. В классе студенты самостоятельно под руководством преподавателя выполняют п. 3 настоящего задания.

2.2. Студенты, которые успешно справились с основным заданием, завершили оформление конспекта и представили его для проверки преподавателю, допускаются к выполнению дополнительного задания (п. 3.3).

2.3. За 10 минут до окончания занятия преподаватель проверяет объем выполненной на занятии работы каждым студентом, делает отметку в протоколе занятия.


## **3. Выполнение работы**

### **3.1. Обслуживание ресурсов ЛВС**

#### **3.1.1. Запуск утилиты “Сеть” панели управления**

Используя вкладку **Поиск** справочной системы *Windows NT* в поле **Введите искомое слово или слова** ввести *Сеть*. В поле **Выберите нужный раздел и нажмите кнопку “Вывести”** выберите раздел *Изменение параметров сети и ее настройка* и ознакомьтесь с содержанием справки.

Используя меню кнопки **Пуск** выполните команду **Настройка ➤ Панель управления**. В открывшемся окне выбрать значок **Сеть** и дважды щелкнуть на нем **ЛМ.3.1.2.** Изучение интерфейса и настройка параметров ЛВС с помощью утилиты “Сеть”

Используя контекстно-зависимую справку (нажать кн.  в правом верхнем углу окна, затем подвести курсор к интересующему полю и нажать **ЛМ**) получить и законспектировать информацию по следующим вопросам.



#### Вкладка **Компьютер**

Установить курсор в поле *Имя компьютера*. Используя контекстно-зависимую справку изучить и законспектировать назначение информации вводимой в поле *Имя компьютера*. Аналогичным образом получить справку для поля *Домен*.

#### Вкладка **Службы**

Получить и ознакомиться со справкой о назначении поля *Сетевые службы*. Законспектировать операции, выполняемые сетевой службой.

#### Вкладка **Протоколы**

Получить и ознакомиться со справкой о назначении поля *Сетевые протоколы*. Законспектировать понятие протокола.

#### Вкладка **Адаптеры**

Получить справку о назначении поля *Сетевые платы*. Законспектировать понятие адаптера (сетевой платы).

### **3.1.3. Работа с файловой структурой на сетевом ресурсе с использованием программы Сетевое окружение**

Для запуска сетевого окружения с использованием программы *Сетевое окружение* необходимо выделить ее значок на рабочем столе и дважды щелкнуть на нем ЛМ. При необходимости, используя клавиши управления развернуть окно программы на весь экран. В главном меню выбрать раздел **Вид** и активизировать опции **Панель инструментов** и **Мелкие значки**.

В окне **Сетевое окружение** выбрать компьютер с номером определяемым по схеме: *Имя компьютера\_(№ рабочего места + 2)*. Просмотреть его сетевой ресурс. Выбрать сервер класса с именем *Nts-1*, в папке *Sety\_1kurs* открыть папку *Data*, просмотреть ее содержимое и скопировать в буфер все файлы с расширением *.doc*.

На своем рабочем месте открыть личную папку (при необходимости создать ее) и в ней создать новую папку под именем *ЛВС*. Внутри созданной папки создать папку *Documents*. Скопировать в нее выбранные файлы.

### **3.1.4. Работа с файловой структурой на сетевом ресурсе с использованием программы Проводник**

Для запуска сетевого окружения необходимо выполнить команду **Пуск ➤ Программы ➤ Проводник**. При необходимости, используя клавиши управления развернуть окно программы на весь экран. В главном меню выбрать раздел **Вид** и активизировать опции **Панель инструментов** и **Мелкие значки**.

В рабочем поле найти значок **Сетевое окружение** и дважды щелкнуть на нем ЛМ. В открывшемся окне **Сетевое окружение** выбрать компьютер с номером определяемым по схеме: *Имя компьютера\_(№ рабочего места + 2)*. Просмотреть его сетевой ресурс. Выбрать сервер класса с именем *Nts-1*, в папке *Sety\_1kurs* открыть папку *Data*, просмотреть ее содержимое и скопировать в буфер все файлы с расширением *.bmp*.

На своем рабочем месте в личной папке открыть папку *ЛВС* и в ней создать новую папку *Pictures*. Скопировать в нее выбранные файлы.

### **3.1.5. Работа с файловой структурой на сетевом ресурсе с использованием программы Windows Commander**

Используя ярлык на рабочем столе запустить программу *Windows Commander*.

На левой панели открыть *Сетевое окружение* и выбрать компьютер с номером определяемым по схеме: *Имя компьютера\_(№ рабочего места + 2)*. Просмотреть его сетевой ресурс. Выбрать сервер класса с именем *Nts-1*, в папке *Sety\_1kurs* открыть папку *Data*. Используя главное меню программы *Windows Commander*, либо клавишу *Ins* выделить в указанной папке все файлы с расширением *.inf*.

На правой панели отобразить содержимое папки *ЛВС* своего рабочего места и в ней создать новую папку *Inform*. Скопировать в нее выбранные ранее файлы.

### **3.1.6. Установка специальных разрешений доступа**

Используя вкладку **Поиск** справочной системы *WINDOWS NT* в поле **Введите искомое слово или слова** ввести *Специальный доступ к каталогам*. В поле **Выберите нужный раздел и нажмите кнопку “Вывести”** выберите раздел *Установка специальных разрешений доступа* и ознакомьтесь с содержанием справки. Законспектируйте назначение данного режима и порядок его выполнения.

### 3.2. Решение сетевых задач

#### 3.2.1. Запуск сетевого приложения из файловой структуры

Используя программу *Сетевое окружение* либо программу *Проводник* произвести запуск сетевого ресурса. Открыть и просмотреть содержимое доступного сетевого ресурса сервера класса – *Nts-1* (при отсутствии указанного имени уточнить у преподавателя). Открыть на сервере папки *Microsoft Office\ Office*. В открытой папке найти файл *Winword.exe* для запуска *Microsoft Word* и дважды щелкнуть на нем ЛМ. Сделать вывод о возможности запуска и использования сетевых приложений.

#### 3.2.2. Создание ярлыка на рабочем столе для запуска сетевого приложения

Произвести щелчок ПМ в свободном пространстве рабочего стола. В появившемся контекстном меню выполнить команду *Создать/Ярлык*. В окне *Создание ярлыка* используя кн. *Обзор* открыть по указанным в предыдущем пункте схеме и маршруту ярлык для запуска *Microsoft Word* с сервера. Далее последовательность действий аналогична созданию обычного ярлыка.

#### 3.2.3. Создание ярлыка в меню кн. Пуск для запуска сетевого приложения

Используя меню кн. *Пуск* выполнить команду *Настройка/Панель задач*. В открывшемся окне *Свойства: Панель задач* выбрать вкладку *Настройка меню* и нажать кн. *Добавить*. В окне *Создание ярлыка* используя кн. *Обзор* открыть папку *Nts-1\Microsoft Office\Office* и выбрать файл *Excel.exe* для запуска *Microsoft Excel*. В следующем окне *Выбор папки* выделить папку *Программы*, после чего нажать кн. *Создать папку*, которой присвоить название *Сетевой ресурс*. Далее последовательность действий аналогична созданию обычного меню кн. *Пуск*.

#### 3.2.4. Создание документа с помощью сетевого приложения Microsoft Word

Используя ярлык на рабочем столе созданный при выполнении п. 3.2.2 произвести загрузку текстового редактора *Microsoft Word* из сетевого ресурса. Создать и отредактировать документ в соответствии с образцом.

##### **Командиру батареи**

##### **Рапорт**

Настоящим докладываю, что за время моего дежурства происшествий не случилось. Оружие и документация в наличии. Нарушений распорядка дня не было.

15 мая 2016 года

Дежурный по батарее: курсант *Иванов*.

Сохранить данный файл на сервере класса в папке *Командир батареи* (папка находится в папке группы по адресу *Nts-1\Sety\_1kurs*) под именем *№\_уч.группы\_Фамилия.doc*.

#### 3.2.5. Совместное использование сетевого документа приложения Microsoft Excel

Используя созданный при выполнении п. 3.3.1.3 раздел меню кн. *Пуск* запустить сетевое приложение *Microsoft Excel*. Используя раздел “?” основного меню *Microsoft Excel* осуществить вызов справки. Используя вкладку *Предметный указатель* в поле *Введите первые буквы нужного слова* набрать *совместная правка* и нажать кн. *Вывести*. В появившемся окне *Найденные разделы* выбрать *Совместный доступ к списку*. Вывести справку и законспектировать назначение данного режима, порядок хранения и обновления списка в сети.

Загрузить книгу *Список группы.xls* открыв на сервере (имя компьютера уточнить у преподавателя) сетевую папку *Командир батареи*. Согласно установленной формы произвести заполнение таблицы в строке соответствующей номеру рабочего места. По

заполнении сохранить внесенные изменения используя панель инструментов, либо меню *Файл*.

### 3.3. Дополнительное задание

**3.3.1.** Произвести удаление ярлыков с рабочего стола и меню кнопки Пуск для созданных сетевых приложений *Microsoft Excel* и *Microsoft Word*.

**3.3.2.** Произвести назначение имени диска для общего сетевого ресурса с рабочего стола или в окне проводника. Аналогичную операцию по подключению сетевого диска выполнить с использованием программы *Windows Commander*.

## 2.10 Лабораторная работа №10 (2 часа).

**Тема:** «Построение, редактирование и форматирование диаграмм»

**2.10.1 Цель работы:** Освоить технологию построения, редактирования и форматирования диаграмм.

### 2.10.2 Задачи работы:

1. Научиться строить диаграммы
2. Научиться форматировать диаграммы
3. Научиться редактировать диаграммы

### 2.10.3 Перечень приборов, материалов, используемых в лабораторной работе:



1. Персональный компьютер
2. Операционная система Windows


### 2.10.4 Описание (ход) работы:

**Диаграммы** – это графическое представление данных. Они используются для анализа и сравнения данных, представления их в наглядном виде.

Диаграмма состоит из **элементов**: линий, столбиков, секторов, точек и т.п. Каждому элементу диаграммы соответствует число в таблице. Числа и элементы диаграммы связаны между собой таким образом, что при изменении чисел автоматически изменяется изображение элементов диаграммы и наоборот.

Различают два вида диаграмм:

**внедренные диаграммы** – сохраняются на рабочем листе вместе с данными; **диаграммные листы** – диаграмма в формате полного экрана на новом листе. Диаграмма создается с помощью **Мастера диаграмм**, вызываемого командой **Вставка, Диаграмма** или кнопкой  на панели **Стандартная** либо кнопкой  на панели **Диаграмма**.

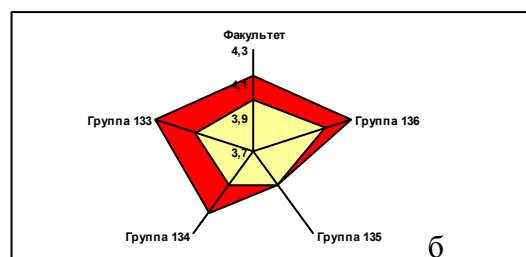
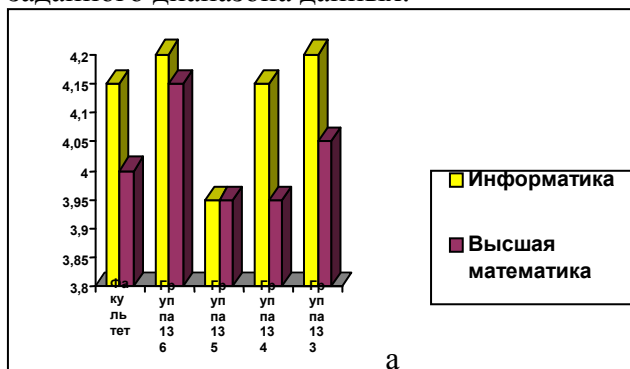
**Мастер диаграмм** позволяет строить диаграммы 14 стандартных типов плоскостного и объемного представления (с областями, линейчатая, гистограмма, график, кольцо , лепестковая, точечная, пузырьковая, поверхностная и др.) и 22 нестандартных типа. Некоторые типы диаграмм представлены на рис. 3.14.

**Мастер Диаграмм** осуществляет построение новой диаграммы в интерактивном режиме за четыре шага только для выделенного блока ячеек – диапазона (области) данных для построения диаграммы.

**Примечание.** Блок ячеек может быть выделен как до вызова **Мастера диаграмм**, так и после его вызова.

**Э т а п 1. Выбор типа и формата диаграммы.** На этом этапе необходимо выбрать тип диаграммы (рис. 3.15) и задать (в окне справа) формат, который делает ее более выразительной. После выбора надо нажать кнопку <Далее> и перейти на следующий этап.

**Э т а п 2. Выбор и указание диапазона данных для построения диаграммы.** На этом этапе задаётся диапазон данных, для которого будет построена диаграмма. Для этого в таблице с помощью переключателя (см. рис. 3.14) укажите расположение данных – по строкам или столбцам будет строиться выбранный тип диаграммы. Далее с помощью мыши выделите необходимый блок ячеек, адрес которого автоматически отразится в строке **Диапазон** (рис. 3.16). В окне образца будет отображаться тип диаграммы для заданного диапазона данных.

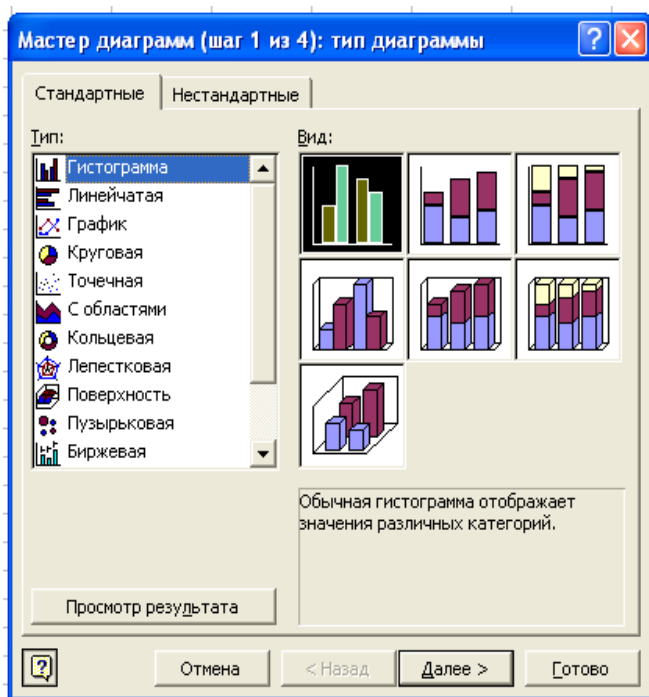


В

Г

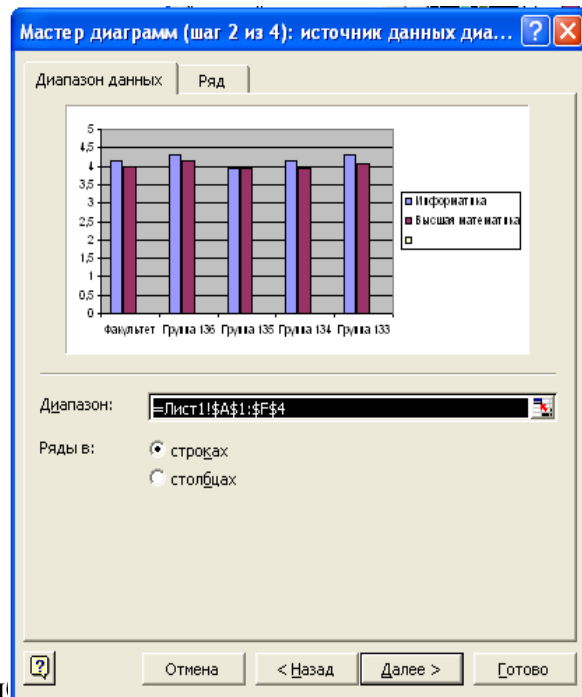
**Рис. 3.14.** Примеры различных типов диаграмм Excel 97: а – объемный вариант обычной гистограмм; б – лепестковая диаграмма; в – линейчатая диаграмма; г – смешанная диаграмма

**Примечание.** Указание диапазона данных, расположенных в несмежных рядах или столбцах, должно производиться при нажатой клавише <Ctrl>. В строке *Диапазон адреса* несмежных интервалов данных будут разделяться точкой с запятой.



грамм

для вы



1) *диаграмм* для задания диапазона данных диаграммы (этап 2)

Блок ячеек может включать как сами данные, так и их названия, которые используются для обозначения меток по оси X и в легендах (расшифровка условных обозначений на диаграммах). Например, на рис. 3.17 показан блок исходных данных, состоящий из чисел и названий (номеров групп).

Блок ячеек может содержать несмежные ячейки одного рабочего листа. В этом случае выделенные блоки должны иметь одинаковую конфигурацию, например, как показано на рис. 3.18

	Средний балл	
	Информатика	Высшая математика
Группа 101	3,5	3,2
Группа 102	4,6	3,9
Группа 103	3,9	4,3
Группа 104	4,2	3,5

7

	1	2			
	Блок1			3	4
	21	24		Блок 2	
				12	14

**Рис. 3.17.** Блок смежных ячеек с исходными несмежных ячеек для построения диаграммы «Сведения об успеваемости»

**Рис.3.18.** Блоки  
ния

Excel 97 позволяет также строить диаграмму на основании данных, распределенных по листам одной и той же или разных рабочих книг. Блоку ячеек можно заранее присвоить имя (команда **Вставка, Имя, Присвоить**).

**Этап 3. Задание параметров диаграммы.** Задание параметров диаграммы осуществляется в окнах вкладок *Мастера диаграмм*, представленного на рис. 3.19.

На вкладке *Заголовки* вводятся поочередно на соответствующую строку название диаграммы, название оси *X*, название оси *Y*, название оси *Z*.

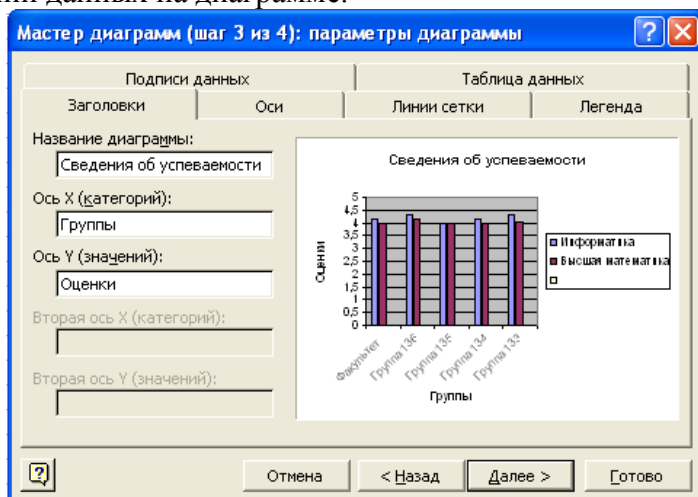
На вкладке *Оси* устанавливаются переключатели выбора вида обозначения меток осей.

На вкладке *Линии сетки* устанавливаются переключатели отображения сетки на диаграмме.

На вкладке *Оси* указывается место расположения легенды.

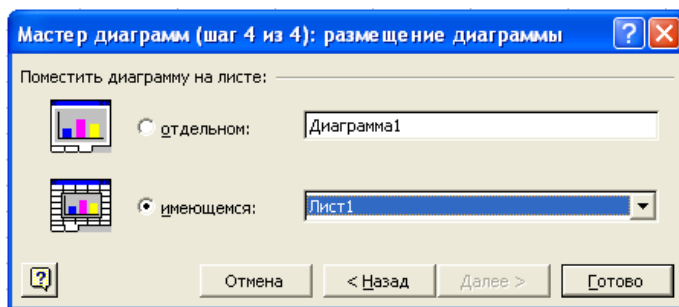
На вкладке *Таблица данных* устанавливается переключатель отображения на диаграмме таблицы исходных данных.

На вкладке *Подписи данных* устанавливается переключатель отображения значений данных на диаграмме.



**Рис. 3.19.** Диалоговое окно *Мастер диаграмм* для задания параметр диаграммы (этап 3)

**Э т а п 4. Размещение диаграммы.** Созданную диаграмму можно разместить на том же листе, где находится таблица с исходными данными, либо на отдельном листе. В диалоговом окне *Мастер диаграмм* на этапе 4 (рис.3.20) для этого надо установить соответствующий переключатель и нажать кнопку **<Готово>**.



**Рис. 3.20.** Диалоговое окно *Мастер диаграмм* для выбора места размещения диаграммы (этап 4)

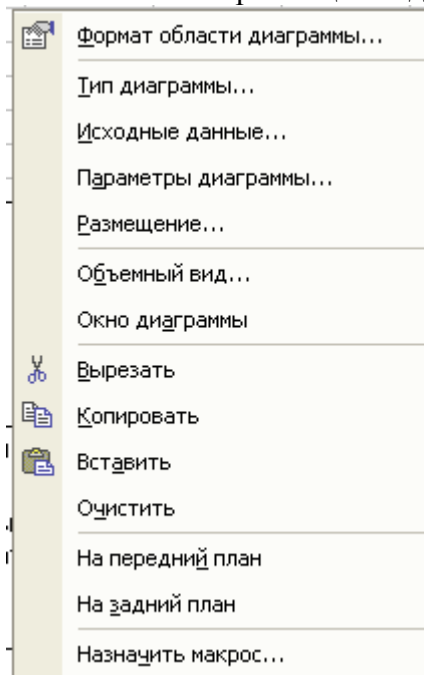
### Редактирование диаграмм

Редактирование диаграмм выполняется как с помощью контекстного меню, так и с помощью команд управляющего меню *Диаграмма*. Вызов контекстного меню осуществляется путем установки указателя мыши в пустое место диаграммы (т.е. в один из четырех ее углов) и нажатия правой клавиши мыши. Контекстное меню имеет вид, представленный на рис. 3.21.

Пункты меню 1 – 5 предоставляют пользователю возможность вернуться к любому из четырех этапов создания диаграммы и осуществить необходимые изменения, т.е.:

- изменить тип и формат диаграммы;
- изменить исходные данные;
- переопределить исходный интервал ячеек, на основании которых построена диаграмма;
- переопределить ориентацию рядов и их название;
- изменить данные, используемые для подписей оси X;
- изменить параметры диаграммы (заголовки, оси, линии сетки, легенду, подписи данных);
- изменить размещение диаграммы.

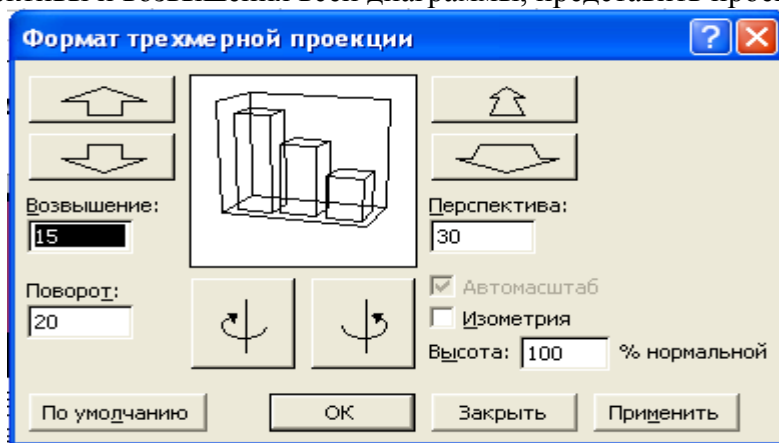
и



**Рис. 3.21.** Контекстное меню для работы с диаграммами

### Форматирование типов диаграмм

ППП Excel 97 обеспечивает удобный режим работы для «подбора» наиболее подходящего типа диаграммы в целом, для группы рядов или отдельного ряда данных. Изменение типа диаграммы можно выполнить для любой построенной диаграммы, как внедренной, так и отдельного диаграммного листа. Подобные изменения могут быть осуществлены либо посредством кнопки <<Тип диаграммы>> на панели инструментов *Диаграмма*, либо за счет перехода к первому этапу построения диаграмм и выбора команды **Тип диаграммы** в контекстном меню (см. рис. 3.21). Кроме того, контекстное меню дает также возможность корректировать вид трехмерных диаграмм. Для изменения формата трехмерной проекции необходимо выполнить команду **Объемный вид** контекстного меню (см. рис. 3.21). В результате перед пользователем предстает диалоговое окно <Формат трехмерной проекции> (рис. 3.22), которое позволяет изменить угол поворота рядов вокруг своей оси, уменьшить или увеличить масштаб, изменить угол перспективы и возвышения всей диаграммы, представить проекцию в изометрии.



**Рис. 3.22.** Диалоговое окно <<Формат трёхмерной проекции>>

### **Форматирование элементов диаграммы**

После активизации диаграммы можно выделить любой из ее элементов для корректировки, которая заключается в удалении либо в изменении свойств компонента путем форматирования.

Существует два пути форматирования компонентов диаграммы:

1. Указать курсором мыши на любой компонент диаграммы (ряд, ось, подписи, легенду ...), щелкнуть ее правой кнопкой и в появившемся контекстном меню выполнить команду **Формат ...** (... = конкретный компонент).
2. В окне <<Элементы диаграммы>> панели инструментов *Диаграмма* выбрать необходимый компонент и нажать соседний с окном значок *Формат*.

### **Форматирование области диаграммы и области построения диаграммы**

Диаграмма окружена *областью*. Форматирование обеспечивает изменение *вида* области (цвета фона, выбор узора, использование рамок вокруг области форматирования) и *шрифта* (типа, стиля и размера для размещаемых в области форматирования символов текста).

Диаграмма любого типа расположена в области *построения*. Выделенную область можно удалить (ряды данных располагаются в области диаграмм, а область построения является их окружением). При форматировании области построения изменению так же, как и для области диаграммы, подлежит лишь ее вид.

### **Форматирование рядов данных**

Диаграмма может содержать несколько рядов данных – групп элементов данных, соответствующих одному блоку ячеек рабочего листа, не обязательно смежных. Каждый ряд на диаграмме выделяется цветом и/или узором.



При выделении отдельного ряда в поле *имен* появляется стандартное обозначение *P(номер)* – номер ряда данных, в строке формул выводится формула ряда, например:

=РЯД(Лист1!\$C\$1;Лист1!\$A\$2:\$A\$6;Лист1!\$C\$2:\$C\$6;2),

где Лист1!\$C\$1 – абсолютная ссылка на ячейку, содержащую название ряда, которое используется при построении легенды;

Лист1!\$A\$2:\$A\$6 – блок ячеек, содержащий метки (категории) значений оси *X*;

Лист1!\$C\$2:\$C\$6 – блок ячеек, содержащий элементы значений ряда;  
2 – порядковый номер ряда.

Выделенный ряд можно удалить.

Форматирование рядов диаграмм осуществляется при помощи диалогового окна «Формат ряда данных», пример которого представлен на рис. 3.23.

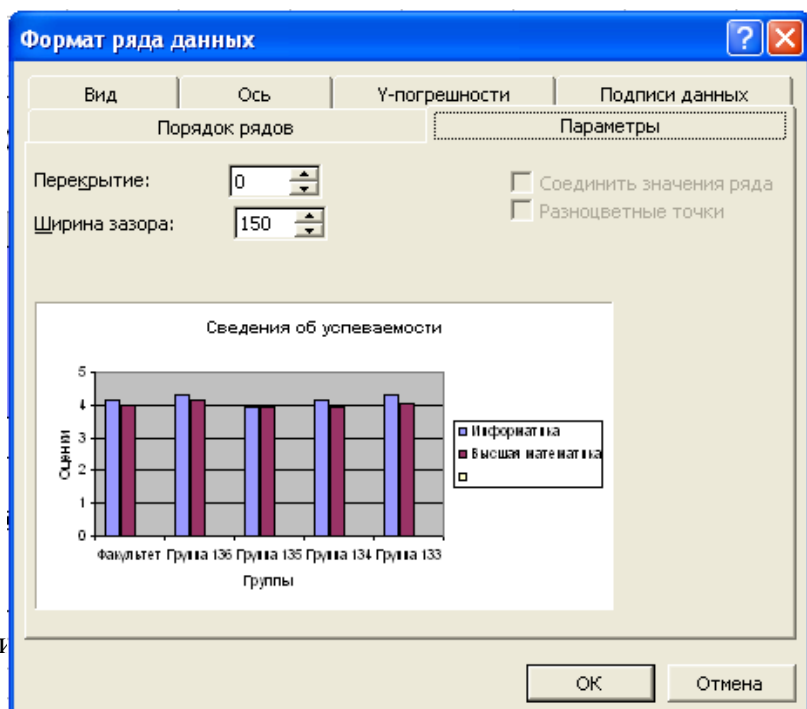


Рис. 3. 23. Ди

Существует некоторое отличие в составе вкладок диалогового окна «Формат ряда данных», обусловленное типом диаграммы, который воспроизводят данные ряды, а именно плоскими и объемными диаграммами. Так вкладки *Вид*, *Подписи данных*, *Порядок рядов*, *Параметры* являются общими для двухмерных и объемных (трехмерных) диаграмм. При этом для двухмерных диаграмм добавляются вкладки *Ось* и *Y-погрешности*, а для трехмерных – добавляется вкладка *Фигура*.

На вкладке *Вид* задаются установки внешнего вида элементов ряда.

На вкладке *Подписи данными* выполняется настройка типа подписей (отсутствие, значения, категория и т.п.) для элементов ряда.

Вкладка *Порядок рядов* позволяет задать требуемый порядок следования рядов, осуществить их перестановку на диаграмме, не изменяя физического местоположения в таблице; в списке рядов выделяется имя перемещаемого ряда; с помощью кнопок <Вверх>, <Вниз> изменяется его порядок следования в диаграмме.

Вкладка *Параметры* позволяет задать важнейшие характеристики диаграмм (рис. 3.25). Состав настраиваемых параметров зависит от типа диаграммы:

Параметр *Перекрытие* (см. рис. 3.24) определяет процент наложения изображений маркеров элементов данных рядов друг на друга, указывается для групп плоских линейчатых диаграмм и гистограмм:

0 – маркеры различных рядов находятся рядом;

-100 – маркеры рядов отстоят друг от друга на максимальном расстоянии;  
100 – маркеры рядов сливаются вместе.



Параметр *Ширина зазора* (рис.3.25) устанавливает величину интервала между кластерами (группой элементов данных различных рядов, отнесенной к одной и той же категории оси X), изменяется в пределах от 0 до 500. Доступна для групп линейчатых диаграмм, графиков, гистограмм, объемных линейчатых диаграмм и объемных гистограмм.



Параметр *Соединять значения ряда* обеспечивает соединение линией вершин маркеров данных одного ряда. Применима только к наложенным линейчатым диаграммам и наложенным гистограммам.

Параметр *Разноцветные точки* элементов данных задается, только для диаграмм, содержащих один ряд данных. Доступна для групп линейчатых диаграмм, гистограмм, графиков, круговых и кольцевых диаграмм, радаров, XY-точечных диаграмм, объемных линейчатых диаграмм, объемных гистограмм и объемных круговых диаграмм.

Параметр *Линии проекции* обеспечивает установку перпендикуляров, опущенных от каждого маркера на ось X. Доступна для групп диаграмм с областями и графиков.

На вкладке *Ось* задается связь ряда данных с основной или вспомогательной осью Y диаграммы. Элементы ряда имеют привязку к двум осям:

X – порядковый номер элемента или определенная категория;

Y – значение элемента в соответствии с его единицей измерения.

Различные ряды диаграммы могут представлять данные, несопоставимые по масштабу. Поэтому при построении диаграмм можно привязать определенный ряд к основной или вспомогательной оси Y, причем к одной оси сможет быть сразу отнесено несколько рядов. Для каждой оси Y выполняется независимая от другой оси Y настройка.

На вкладке *Y-погрешности* определяется тип планок погрешностей для элементов ряда согласно его статистическим характеристикам.

На вкладке *Фигура* в качестве визуального представления ряда данных можно выбрать одну из шести трехмерных фигур: параллелепипед, конус, цилиндр, пирамиду, усеченный конус или усеченную пирамиду.

**Форматирование осей диаграммы**

Оси диаграммы предназначены для обрамления области построения диаграммы, нанесения разметки (шкал), которым соответствуют основные значения элементов данных и категории.

Для большинства плоских диаграмм используются первичные (основные) и вспомогательные оси, что обусловлено типом диаграммы, величиной (порядком величин) значений элементов данных рядов. В частности, круговые и кольцевые диаграммы не имеют осей, радиальные диаграммы имеют отдельные оси для каждого ряда данных. Ориентация ряда относительно оси  $Y$  выполняется при форматировании рядов, по умолчанию все ряды отнесены к основной оси. Ось  $X$  является общей для всех рядов, независимо от их ориентации относительно оси  $Y$ .

Объемная диаграмма стоит на *основании* и окружена стенками, которые можно самостоятельно форматировать.

При форматировании осей активизируется:

вкладка *Вид* – выбирается внешнее оформление оси (линии), указывается наличие и расположение основных и вспомогательных засечек и их меток;

вкладка *Шрифт* – выбирается вид и размер шрифта меток засечек;

вкладка *Шкала* – задается масштаб значений элементов данных ряда на оси и разметка оси (расстояние между засечками и линиями сетки), значение точки пересечения с осью  $X$  (для плоских) или с основанием (для объемных) диаграмм; выбирается при необходимости логарифмическая шкала и обратный порядок значений (для оси категорий – изменение порядка следования меток слева направо, для оси значений – изменение значения от меньшего к большему);

вкладка *Число* – задается формат меток засечек или самих числовых значений на оси;

вкладка *Выравнивание* – изменяется ориентация текста меток засечек.

### **Форматирование сетки**

Сетка обеспечивает лучшее изображение числовых данных, зрительно облегчает сопоставление данных. При выделении сетки в поле имен появляется стандартное название компонента – *Сетка*. При форматировании сетки указывается тип линии и шкала разметки (пересечение линий сетки с осями).

### **Форматирование легенды**

Легенда – специальное окно, содержащее для каждого ряда данных *ключ* и *поле-название* ряда. Ключ легенды повторяет цвет и узор, заданный для элементов данных ряда. Легенда может создаваться в автоматическом режиме, если при построении диаграммы интервал ячеек включал названия рядов. Если имена рядов явно не заданы, то легенда использует стандартное имя **Ряд (номер)**.

При форматировании целого окна легенды задается вид рамки, цвет и узор закрашки области легенды, определяется шрифт, используемый для текста легенды, размещение легенды (*внизу, вверху, слева, справа, в углу*).

### **Тренды**

Тренд – это функция заданного вида, с помощью которой можно аппроксимировать построенный по данным таблицы график. Тренд служит для выявления тенденций развития процесса, представленного в виде диаграммы, и обеспечивает прогноз на заданный период. В Excel предусмотрено несколько стандартных типов тренда (рис. 3.26): линейный, логарифмический, степенной, экспоненциальный, полиномиальный, скользящее среднее. Необходимые условия построения тренда:

- период времени, за который изучается прогнозируемый процесс, должен быть достаточным для выявления закономерности;
- тренд в анализируемый период должен развиваться эволюционно;
- процесс, представленный диаграммой, должен обладать определенной инерционностью.

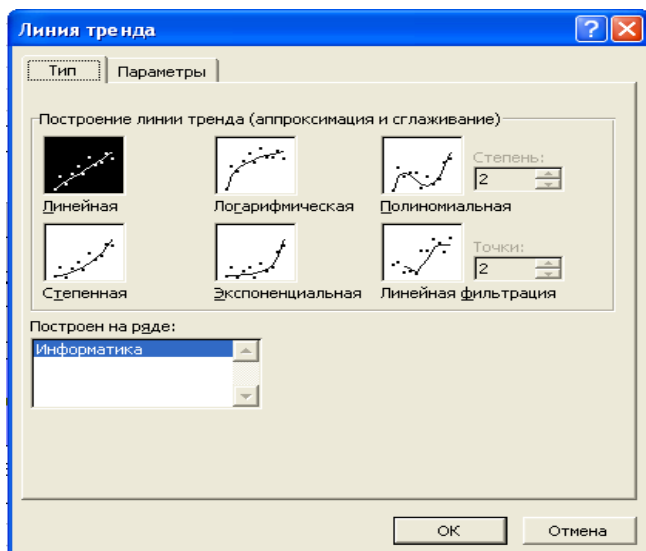
Тренд можно строить для диаграмм типа:

- *линейчатый график;*
- *гистограмма;*
- *диаграмма с областями;*
- *XY-точечная диаграмма.*

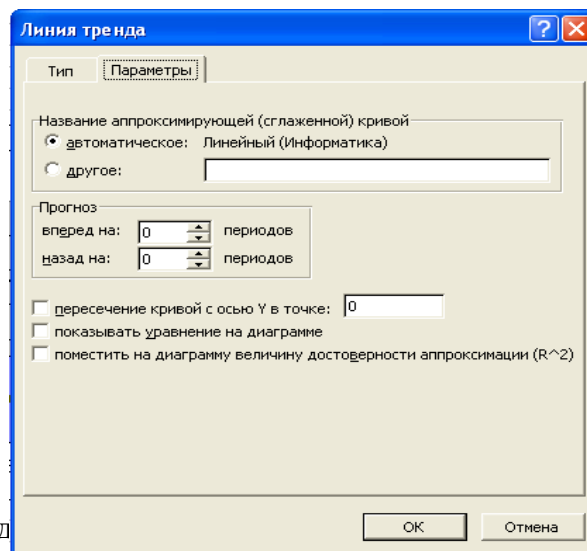
Построение тренда осуществляется по следующей технологии:

- построить диаграмму для одного ряда данных;
- выделить эту диаграмму, щелкнув по ней левой кнопкой мыши. На диаграмме должны появиться маркеры;
- вызвать контекстное меню и выполнить команду **Добавить линию тренда**;
- в диалоговом окне <<Линия тренда>> на вкладке Тип (см. рис.3.26а) выбрать тип тренда, а на вкладке Параметры (см. рис.3.26б) установить параметры: *Количество периодов прогноза, Показывать уравнение на диаграмме, Поместить на диаграмму величину достоверности аппроксимации*;
- нажать кнопку <ОК>.

Для последующего редактирования линии тренда требуется ее выделение (двойным щелчком левой кнопки мыши), при этом появляются специальные отметки на линии тренда. С помощью правой кнопки мыши можно вызвать контекстное меню, обеспечивающее форматирование (команда **Форматировать линию тренда**), либо удаление линии тренда (команда **Очистить**).



и (б) Д



## **2.11 Лабораторная работа №11 (2 часа).**

**Тема:** «Корпоративные информационные системы»

**2.11.1 Цель работы:** изучение принципов построения хранилищ данных корпоративных информационных систем.

### **2.11.2 Задачи работы:**

1. Разработать структуру корпоративной базы данных предприятия с использованием СУБД MS SQL Server;
2. Написать хранимые процедуры для заполнения базы данных данными, имитирующими деятельность предприятия за несколько лет;
3. Разработать структуру хранилища данных для созданной модели корпоративной базы данных.

### **2.11.3 Перечень приборов, материалов, используемых в лабораторной работе:**

1. Персональный компьютер
2. Операционная система Windows

### **2.11.4 Описание (ход) работы:**

Принятие любого управленческого решения в процессе управления крупным предприятием, невозможно не обладая необходимой для этого аналитической информацией, получаемой в процессе сбора, отсеивания и предварительной обработки данных с целью предоставления результирующей информации пользователям для статистического анализа и создания аналитических отчетов. Поэтому, корпоративные информационные системы, как правило, содержат приложения, предназначенные для комплексного многомерного анализа данных, их динамики, тенденций и т.п., называемые системами поддержки принятия решений. Эти приложения основываются на концепции хранилищ данных (Data warehouses).

Ральф Кимбалл (Ralph Kimball), один из авторов концепции хранилищ данных, описывал хранилище данных как "место, где люди могут получить доступ к своим данным". Он же сформулировал и основные требования к хранилищам данных:

- поддержка высокой скорости получения данных из хранилища;
- поддержка внутренней непротиворечивости данных;
- возможность получения и сравнения так называемых срезов данных (slice and dice);
- наличие удобных утилит просмотра данных в хранилище;
- полнота и достоверность хранимых данных;
- поддержка качественного процесса пополнения данных.

Удовлетворять всем перечисленным требованиям в рамках одного и того же продукта зачастую не удастся. Поэтому для реализации хранилищ данных обычно используется несколько продуктов, одни из которых представляют собой собственно средства хранения данных, другие – средства их извлечения и просмотра, третьи – средства их пополнения и т.д.

Типичное хранилище данных, как правило, отличается от обычной реляционной базы данных. Во-первых, обычные базы данных предназначены для того, чтобы помочь пользователям выполнять повседневную работу, тогда как хранилища данных предназначены для принятия решений. Например, продажа товара и выписка счета производятся с использованием базы данных, предназначенной для обработки транзакций, а анализ динамики продаж за несколько лет, позволяющий спланировать работу с поставщиками, – с помощью хранилища данных.

Во-вторых, обычные базы данных подвержены постоянным изменениям в процессе работы пользователей, а хранилище данных относительно стабильно: данные в нем обычно обновляются согласно расписанию (например, еженедельно, ежедневно или ежечасно – в зависимости от потребностей). В идеале процесс пополнения представляет собой просто добавление новых данных за определенный период времени без изменения прежней информации, уже находящейся в хранилище.

И в-третьих, обычные базы данных чаще всего являются источником данных, попадающих в хранилище. Кроме того, хранилище может пополняться за счет внешних источников, например статистических отчетов.

Системы поддержки принятия решений обычно обладают средствами предоставления пользователю агрегатных данных для различных выборок из исходного набора в удобном для восприятия и анализа виде. Как правило, такие агрегатные функции образуют многомерный (и, следовательно, нереляционный) набор данных (нередко называемый гиперкубом или метакубом), оси которого содержат параметры, а ячейки — зависящие от них агрегатные данные - причем храниться такие данные могут и в реляционных таблицах, но в данном случае мы говорим о логической организации данных, а не о физической реализации их хранения). Вдоль каждой оси данные могут быть организованы в виде иерархии, представляющей различные уровни их детализации. Благодаря такой модели данных пользователи могут формулировать сложные запросы, генерировать отчеты, получать подмножества данных.

Технология комплексного многомерного анализа данных получила название OLAP (On-Line Analytical Processing). OLAP — это ключевой компонент организации хранилищ данных. Концепция OLAP была описана в 1993 году Эдгаром Коддом, известным исследователем баз данных и автором реляционной модели данных. В 1995 году на основе требований, изложенных Коддом, был сформулирован так называемый тест FASMI (Fast Analysis of Shared Multidimensional Information — быстрый анализ разделяемой многомерной информации), включающий следующие требования к приложениям для многомерного анализа:

- предоставление пользователю результатов анализа за приемлемое время (обычно не более 5 с), пусть даже ценой менее детального анализа;

- возможность осуществления любого логического и статистического анализа, характерного для данного приложения, и его сохранения в доступном для конечного пользователя виде;

- многопользовательский доступ к данным с поддержкой соответствующих механизмов блокировок и средств авторизованного доступа;

- многомерное концептуальное представление данных, включая полную поддержку для иерархий и множественных иерархий (это — ключевое требование OLAP);

- возможность обращаться к любой нужной информации независимо от ее объема и места хранения.

Следует отметить, что OLAP-функциональность может быть реализована различными способами, начиная с простейших средств анализа данных в офисных приложениях и заканчивая распределенными аналитическими системами, основанными на серверных продуктах. Но прежде чем говорить о различных реализациях этой функциональности, давайте рассмотрим, что же представляют собой кубы OLAP с логической точки зрения.

#### **Порядок выполнения**

Создание базы данных.

Атрибуты колонок таблиц SQL-Server 2000:

1. Column Name – имя колонки
2. Data Type – тип данных
3. Length – длина n для символьных видов
4. Allow Nulls – разрешение значения NULL
5. Description – описание
6. Default Value – значение по умолчанию (н-ер: NewId() для *uniqueidentifier*, GetDate() для *datetime*)
7. Precision – точность (общее количество знаков), p – для *decimal* и *numeric*
8. Scale – масштаб (количество знаков после запятой), s – для *decimal* и *numeric*
9. Identity – признак счетчика

10. Identity Seed – начальное значение счетчика
11. Identity Increment – шаг приращения счетчика
12. Is RowGuid – признак глобального идентификатора
13. Formula – для вычисляемых столбцов (н-ер: Цена \* Количество)
14. Collation – сопоставление для сравнения и сортировки строк

Типы данных SQL-Server 2000:

Вид	Тип	Интервал значений	Размер
Двоичные	<i>binary</i> (n) <i>varbinary</i> (n) <i>image</i>	до 8 000 байт до 8 000 байт до 2 Гбайт	n n
Символьные	<i>char</i> (n) <i>varchar</i> (n) <i>nchar</i> (n) <i>nvarchar</i> (n)	до 8 000 байт до 8 000 байт до 4 000 байт (Unicode) до 4 000 байт (Unicode)	n n n n
Текст	<i>text</i> <i>ntext</i>	до 2 Гбайт до 1 Гбайт (Unicode)	
Дата и время	<i>datetime</i> <i>smalldatetime</i>	01.01.1753-31.12.9999г. до 3,33 мс 01.01.1900-06.06.2079г. до 1 мин.	8 байт 4 байта
Точное представление чисел	<i>decimal</i> (p, s) <i>numeric</i> (p, s)	p <= 38, s <= p p <= 38, s <= p	При p = 2 – 2 байта, при p = 38 – 17 байт
Числа с плавающей точкой	<i>float</i> (n) <i>real</i>	$\pm 1,8 \cdot 10^{308}$ $\pm 1,8 \cdot 10^{308}$	n = 1 – 53 n = 24 n – число бит мантиссы
Целочисленные типы	<i>int</i> <i>smallint</i> <i>tinyint</i> <i>bigint</i>	$\pm 2 \cdot 10^{10}$ $\pm 32\,767$ 0 – 255 $\pm 9 \cdot 10^{19}$	4 байта 2 байта 1 байт 8 байт
Денежные типы	<i>money</i> <i>smallmoney</i>	$\pm 9 \cdot 10^{15}$ , 4 знака после запятой $\pm 214\,748.3648$	8 байт 4 байта
Специальные	<i>bit</i> <i>timestamp</i> <i>uniqueidentifier</i> <i>sysname</i> <i>sql_variant</i> <i>table</i> <i>cursor</i>	0 или 1 (логический) отслежив.последоват.изм. записей NewID()	1 бит 8 байт 16 байт

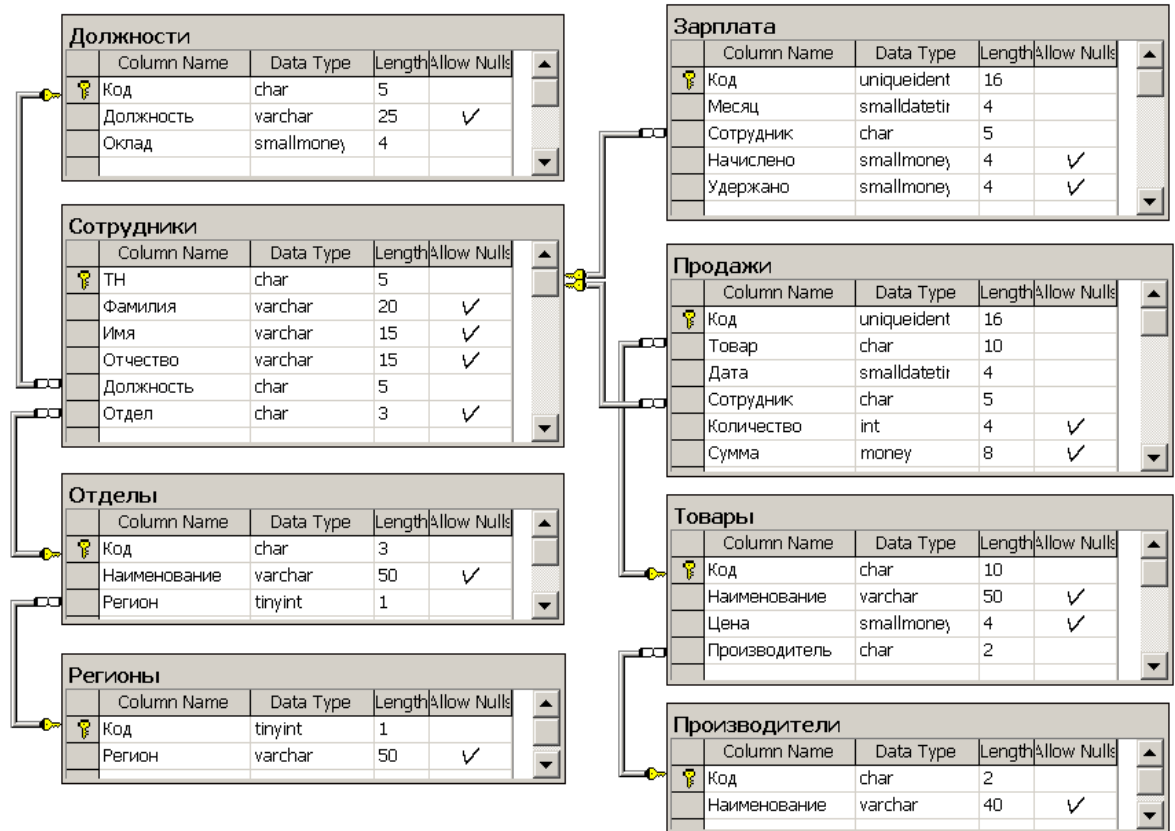
Установка формата даты: SET DATEFORMAT dmy. Первого дня недели: SET DATEFIRST 1 (1 – понедельник, 7 – воскресенье).

Преобразование типов данных:

CAST(expr AS type)

CONVERT(type, expr, [style]) style – при преобразовании в дату: 4 – dd.mm.yy, 104 – dd.mm.yyyy

Структура примера исходной базы данных корпоративной информационной системы «КИС»:



### 1.2.2 Создание хранимых процедур для заполнения базы данных

Ниже приведены примеры процедур и функций, заполняющие таблицу заработной платы сотрудников предприятия и случайным образом генерирующие данные о продажах товаров этими сотрудниками в течении заданного периода времени.

```

CREATE FUNCTION ПервыйДеньМесяцаПоДате (@Дата SmallDateTime)
RETURNS SmallDateTime
AS
BEGIN
    Select @Дата = @Дата - Day(@Дата) + 1
    return @Дата
END
CREATE FUNCTION ПроверкаТабельногоНомера (@Код char(4))
RETURNS bit
AS
BEGIN
    Declare @Yes bit
    if exists(select ТабельныйНомер from Сотрудники where ТабельныйНомер =
@Код)
        Set @Yes = 1
    else
        Set @Yes = 0
    return @Yes
END

```



```

CREATE PROCEDURE dbo.НачислениеЗП (@Месяц SmallDateTime)
AS
insert into dbo.Зарплата
    select  NewId() as Код,
            @Месяц as Месяц,
            dbo.Сотрудники.ТН,
            dbo.Должности.Оклад as Начислено,
            dbo.Должности.Оклад*0.13 as Удержано
    from
        dbo.Сотрудники inner join dbo.Должности
            on dbo.Сотрудники.Должность = dbo.Должности.Код
GO

```

```

CREATE PROCEDURE dbo.MassCalculate (@НачДата SmallDateTime)
AS
declare @Дата SmallDateTime
declare @i TinyInt
set @i = 1
set @Дата = Convert(Smalldatetime, '01 ' + DateName(Month, @НачДата)
    + '' + Cast(DatePart(Year, @НачДата) as Char(4)), 107)
while @i < 120
begin
    select @Дата = DateAdd(Month, 1, @Дата)
    exec dbo.НачислениеЗП @Дата
    select @i = @i + 1
end
GO

```

```

CREATE PROCEDURE dbo.Продажи10лет (@ДатаНач SmallDateTime) AS
Declare @iМесяц Int, @iСотр Int, @iТов Int
Declare @СотрВсего Int, @Сотр Int
Declare @ТНСотр Char(5)
Declare @ТовВсего Int, @Тов Int, @КолТов Int
Declare @КодТовара Char(5)
Declare @ЦенаТовара SmallMoney
Declare @iПродаж Int, @КолПродаж Int
Declare КурсорСотр Cursor local scroll for Select ТН from Сотрудники
Declare КурсорТовары Cursor local scroll for Select Код, Цена from Товары
Open КурсорСотр
Open КурсорТовары
Select @СотрВсего = Count(Сотрудники.ТН) From Сотрудники
Select @ТовВсего = Count(Товары.Код) From Товары
Delete from Продажи
Select @ДатаНач = ПервыйДеньМесяцаПоДате(@ДатаНач)
Set @iМесяц = 1
While @iМесяц <= 120
begin
    Set @iСотр = 1
    While @iСотр <= @СотрВсего
    begin
        Select @Сотр = Round(Rand() * @СотрВсего, 0)
        Fetch Absolute @Сотр from КурсорСотр Into @ТНСотр
    end
end

```

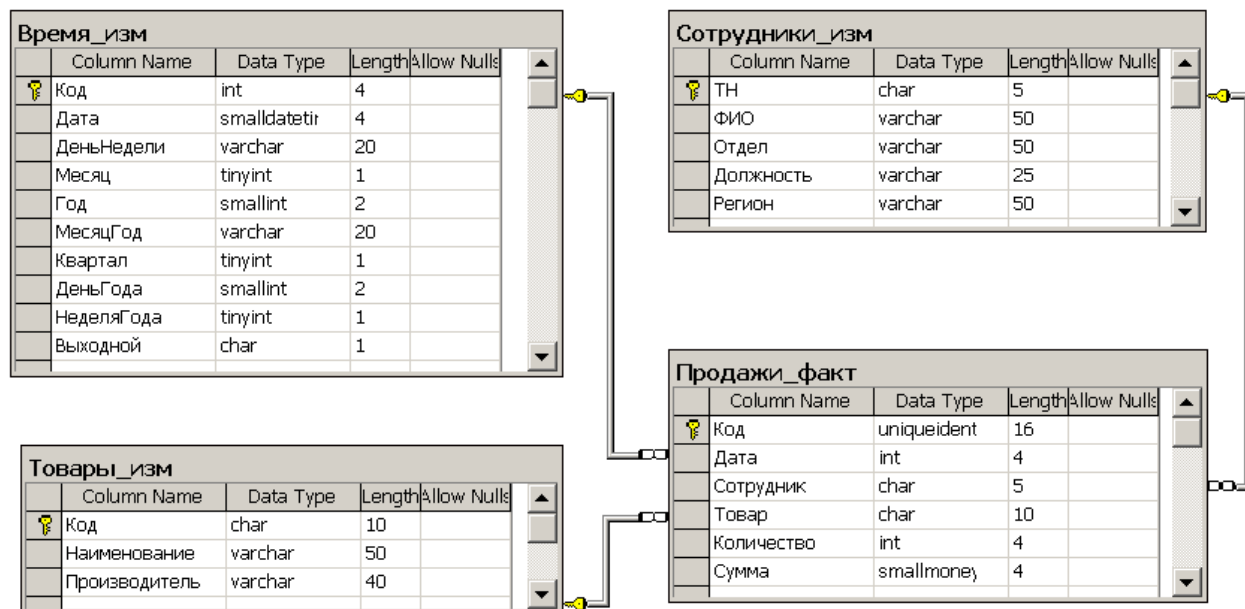
```

Set @iПродаж = 1
Select @КолПродаж = Floor(Rand() * 50)
While @iПродаж <= @КолПродаж
begin
    Select @КолТов = Floor(Rand() * 5)
    Select @Тов = Round(Rand() * @ТовВсего, 0)
    Fetch Absolute @Тов from КурсорТовары Into @КодТовара,@ЦенаТовара
    Insert Into Продажи (Код,Дата,Сотрудник,Товар, Количество,Сумма) Values
    (NewId(),@ДатаНач,@ТНСотр,@КодТовара,@КолТов,@ЦенаТовара*@Кол
Тов)
    Select @iПродаж = @iПродаж + 1
end
Set @iСотр = @iСотр + 1
end
Select @ДатаНач = DateAdd(Month, -1, @ДатаНач)
Select @iМесяц = @iМесяц + 1
end
Close КурсорСотр
Close КурсорТовары
Deallocate КурсорСотр
Deallocate КурсорТовары
GO

```

### 1.2.3 Создание хранилища данных

Структура хранилища данных «КИС\_ХД» для приведенной выше базы данных:



## **2.12 Лабораторная работа №12 (2 часа).**

**Тема:** «Глобальные компьютерные сети»

**2.12.1 Цель работы:** научиться работать в локальной и глобальной компьютерной сети.

### **2.12.2 Задачи работы:**

- 1.Выполнение операций копирования и печати через локальную сеть.
2. Поиск данных в Internet и копирование их на свой винчестер.
3. Почта в Internet – создание почтового ящика, отправка и получение корреспонденции.

4. Создание сайта в Internet.

### **2.12.3 Перечень приборов, материалов, используемых в лабораторной работе:**

1. Персональный компьютер
2. Операционная система Windows

### **2.12.4 Описание (ход) работы:**

Общие положения

В настоящее время большинство компьютеров используются не изолированно от других компьютеров, а подключены к локальным, распределённым или глобальным компьютерным сетям.

Локальная сеть – это группа из нескольких компьютеров, соединённых между собой кабельной линией, через которую компьютеры могут обмениваться информацией. Использование локальных сетей обеспечивает:

- коллективную обработку данных пользователями подключённых в сеть компьютеров и обмен данными между ними;
- совместное использование программ, а также принтеров, модемов, сканеров и другого оборудования.

Локальная сеть используется, в основном, пределах одного здания, поэтому все фирмы, имеющие более одного компьютера, объединяют их в сеть.

Распределённая сеть соединяет группу компьютеров посредством телефонных или кабельных линий (например, межбанковская городская, государственная сеть), т.е. объединяет группу локальных сетей.

Глобальная сеть соединяет группы компьютеров различных стран посредством телефонных каналов или через радиоканалы – посредством спутниковой связи - Internet.

Оборудование для локальных сетей. Для объединения компьютеров в локальную сеть в каждый компьютер вставляется сетевой контроллер, который позволяет ему получать информацию из локальной сети или передавать данные в сеть. Соединяются компьютеры между собой кабелями, по которым происходит передача данных и сигналов управления между компьютерами, принтерами, сканерами и другими подключенными устройствами.

В простейших сетях (для одного помещения) кабели соединяют компьютеры непосредственно, в других (более мощных) соединении кабелей производится через концентраторы (или хабы) – для сжатия данных и усиления сигналов, коммутаторы - для распределения сигналов и соединения между собой только конкретных, а не всех компьютеров.

Серверы.

Сервер – это специальный компьютер, обеспечивающий функционирование локальной сети.

Он является самым мощным компьютером в системе с большим объёмом оперативной и дисковой памяти, имеет повышенную надёжность, высокую производительность ввода-вывода, дублирование устройств и хранимых данных, средства контроля над состоянием сервера, обеспечения бесперебойной работы системы при отказе некоторых устройств и др. На его дисках расположены совместно используемые программы, базы данных и т.д.

Остальные компьютеры называются рабочими станциями, на них часто, из соображений экономии или безопасности, не устанавливают жестких дисков. Для сети, состоящей из 20-25 и более компьютеров, наличие сервера обязательно – иначе производительность такой компьютерной системы будет неудовлетворительной. Серверы обычно не используются в качестве рабочих станций.

Для простых сетей функции сервера выполняет один из компьютеров сети.

Программное обеспечение. Операционные системы Windows 98, Windows 2000, Windows ХТ имеют встроенные возможности по организации небольших локальных сетей, поэтому для компьютеров, использующих эти системы, дополнительное программное обеспечение не требуется.

В локальных сетях с выделенным компьютером-сервером используют специальные операционные системы – Novell NetWare, Windows NT Server и др. На рабочих станциях такой локальной сети может использоваться любая операционная система (DOS, Windows и т.д.) и должен быть запущен программный драйвер, который обеспечивает доступ каждого компьютера к сети.

В сетях с сервером применяется также вспомогательное программное обеспечение:

электронная почта – обеспечивает доставку писем и других файлов от одних пользователей сети к другим, средства удалённого доступа – позволяют подключать компьютер, не входящий в локальную сеть через модем, т.е. по телефонным проводам (модем – это модулятор\демодулятор, преобразующий аналоговый сигнал в цифровой и обратно), в том числе и к сети Internet, средства групповой работы – позволяют нескольким пользователям совместно работать над одним документом и др.

Internet– это общемировая совокупность компьютерных сетей, связывающая между собой миллионы компьютеров через спутниковый радиоканал, появилась в конце 60-х годов, т.е. это всемирная глобальная компьютерная сеть.

WWW (Word Wide Web – «всемирная паутина»)– подсистема Internet, являющаяся всемирной распределённой базой текстовых документов, хранящихся на компьютерах-серверах по всему миру. Поставщиком документов может стать любой человек, имеющий компьютер, подключенный к Internet.

Для просмотра других серверов WWW пользователь должен иметь программу просмотра WWW, называемую Web –броузером. Подсоединившись к Internet, пользователь должен ввести имя нужного ему Web – сервера и его содержимое появится на экране. При подключении к Web – серверу на экран выводится Web – страница, похожая на экран справочника Windows- программы, пролистав гиперссылки которого можно за несколько минут просмотреть любую информацию Web – серверов в десятках стран – научные труды, газеты, журналы, информацию по бизнесу, промышленности, спорту, культуре и т.п.

Вход в локальную сеть.

В конкретной локальной сети для получения доступа к другому компьютеру нужно на рабочем столе Windows—98 (Windows- NT и т.п.) найти значок “Сетевое окружение”, щелкнуть по нему, откроется окно со списком компьютеров, входящих в данную компьютерную сеть. Двойной щелчок по значку любой станции откроет диски А, С, D и т.д., выбранного компьютера, сетевые принтер, сканер и т.п. При этом можно переписать данные со своего винчестера на дискету компьютера – абонента, распечатать свой документ на сетевом принтере, ввести изображение, полученное на сканере абонента в свой компьютер и др.

Прекращение связи с абонентом выполняется закрытием его окна на своём экране.

При одновременном обращении к одному абоненту нескольких пользователей компьютер-сервер устанавливает очерёдность запросов. Внеочередным правом входа в любой абонент сети пользуется сам сервер, он также может менять очерёдность запросов, отменять какие-то запросы или переводить их на другую рабочую станцию и т.д.

Вызов некоторых абонентов и, особенно, сервера может потребовать ввода пароля.

В целом работа с компьютером – абонентом происходит медленнее за счёт задержек времени, возникающих при прохождении сигналов по кабелям.

Кабель типа витая пара пропускает информацию со скоростью до 3000 Кб/с, коаксиальный кабель – до 9000 Кб/с (Кб – килобит).

Работа в сети Internet.

Каждая страница в WWW имеет собственный уникальный адрес, называемый URL – Uniform Resource Locator и состоящий из названия протокола, названия сервера и пути доступа к странице, например: <http://www.aires.com.ru/book/proto..html>

Здесь: [http](http://www.aires.com.ru/book/proto..html) (HyperText Transfer Protocol) – обозначение протокола. Протокол – это набор правил, регламентирующих способы передачи данных между компьютерами в сети, определяет правила обращения к Web-странице. Существуют также другие протоколы, например, [ftp](#) - File Transfer Protocol.

[www.aires.com.ru](http://www.aires.com.ru) – наименование Web-сервера (доменное имя) компьютера, содержащего требуемую Web-страницу, где [com](#) указывает на вид домена - коммерческий ([mil](#) – военный, от [military](#) и т.п.), [ru](#) – гипердомен, отведённый в Internet России ([en](#) – Англия, [do](#) – Германия и т.д.) [book/proto..html](#) – путь доступа к странице.

Этот адрес указывает, что Web-страница [proto..html](#) находится в папке [Book](#), находящейся в коммерческом домене российского гипердомена компьютера - Web-сервера с именем [www.aires.com.ru](http://www.aires.com.ru), к которому нужно обращаться по правилам протокола [http](#).

Web-страница – это документ Web, содержащий информацию вида – текст, рисунки, фотографии, фрагменты аудио- и видеозаписей. Хранятся на компьютерах – Web-серверах. Web-страницы обычно содержат гиперссылки - выделенные цветом и подчёркиванием фрагменты текста или графические изображения в цветных рамках, которые позволяют при нажатии на них переходить к другим страницам Web без ввода URL этих страниц.

Web-страницы записываются на языке программирования HTML – HyperText Markup Language и сохраняются в виде файлов с расширением [.html](#) или [.htm](#). web-страница – это документ в формате HTML, содержащий текст и специальные инструкции – дескрипторы (тэги) HTML, которые сообщают Web-браузеру информацию о структуре и особенностях форматирования Web-страницы. Дескрипторы записываются в угловых скобках, например, тэг [<TITLE>](#) служит для обозначения наименования страницы.

Web-сервер – компьютер-сервер, хранящий на своих дисках Web-страницы.

Web-браузер – программа, предоставляющая средства поиска и просмотра информации в Web. Web-браузер считывает Web-страницы и другие файлы с диска Web-сервера и отображает их на мониторе компьютера пользователя. Наиболее популярными программами-браузерами являются Microsoft Internet Explorer и Netscape Navigator.

Web-сайт – совокупность Web-страниц, начинается с главной страницы, которую пользователь, обращающийся к Web-сайту, видит в первую очередь. Файл главной страницы обычно называется [index..html](#) или [index..htm](#).

Главная страница должна содержать краткий обзор содержимого сайта, его оглавление и набор гиперссылок для быстрого перехода к темам сайта.

Основными практическими задачами работы в Internet являются:

1. Поиск необходимой информации (статей, книг, новостей, почты, объявлений и др.).
2. Копирование найденной информации на свой компьютер или его распечатка непосредственно из Internet.
3. Создание почтового ящика, отправка и получение корреспонденции.
4. Создание сайта в Internet.

Для выхода в сеть Internet нужно дважды щелкнуть мышью по значку Internet на Рабочем столе, появится окно Microsoft Internet Explorer (рис.1) с окном Удалённое соединение.

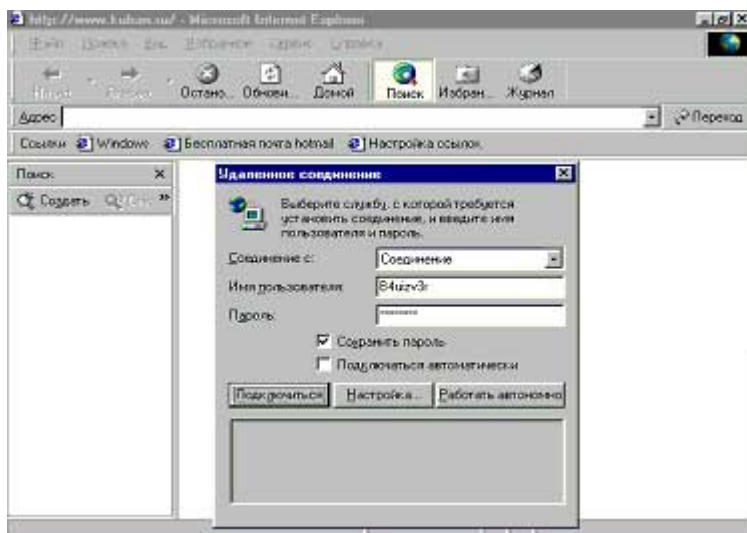


Рис. 1. Окно Microsoft Internet Explorer в котором нужно ввести пароль и имя пользователя. Пароль берётся из Internet-карты.

После нажатия клавиши Подключиться компьютер соединяется сетью Internet.

Поиск и копирование данных.

Выполняется следующим образом:

1. Нажать кнопку Поиск на панели Стандартная. Рабочая зона разделится на две части: левая – для ввода ключевых слов для поиска, правая – для вывода найденной информации. Поиск выполняется с помощью поисковиков Aport, Rambler и др. По умолчанию предлагается Aport.

2. Ввести ключевые слова в левую часть, нажать клавишу Начать поиск. В результате в левой части появится количество найденных документов и список первых 15-ти из них. Перейти к следующим 15 документам можно внизу списка. Содержание выделенного документа выводится в правой части.

3. Выделить текст выбранного документа командой Правка\Выделить всё, затем выполнить команду Правка\Копировать выйти из Internet.

4. Открыть на своём компьютере новый документ, выполнить команду Правка\Вставить. Можно вместо выполнения п. 4, не выходя из Internet, выполнить команду Файл\Сохранить как... и указать в окне сохранения имя файла и папку, в которой документ должен быть сохранён., например, Диск 3,5(A).

Электронный почтовый ящик.

Создаётся следующим образом:

1. После открывания окна Microsoft Internet Explorer в поле Адрес набрать WWW Newmail.ru – это сайт (страница, в переводе – местоположение, местонахождение) компании ``Online Resource Center`` которая бесплатно предоставляет в Internet почтовые услуги.

2. Адрес почтового ящика состоит из Логина и Доменного имени, разделённых символом ``собака`` @ (В Китае его зовут ``улитка``, в Венгрии - ``червяк``, в Норвегии – ``котёнок``). Например, в адресе komkov@newmail.ru логином является komkov, а доменным именем – newmail.ru. Логин – это название почтового ящика, а Доменное имя – название сервера, т.е. компьютера (Web-сервера), на котором хранятся Web-страницы. Web – это сокращение от WWW – World Wide Web – всемирная паутина или просто Internet

В левой части окна нажать Регистрация, в правой части появится текст Договора по пользованию электронной почтой. В конце текста Договора нажать Я согласен, появится регистрационная страница, на которой нужно выбрать регистрационное имя, в строке Логин ввести название почтового ящика, например, priba, в строке Доменное имя выбрать из списка New mail.ru, nm.ru или любое другое, в строке Укажите пароль ввести свой пароль, повторить его в строке Повторите пароль.

3. Если Вы забыли пароль, запишите в окнах Вопрос и Уникальный ответ, те данные, которые помогут вспомнить пароль. Например, для пароля Мурка вводим вопрос ``Любимые домашние животные?`` и ответ ``Кошка``.

4. Нажать надпись Продолжить регистрацию, компьютер выполняет сверку с базой данных, если такого логина нет, то регистрация продолжается, если есть, то база данных просит сменить его. Появляется надпись ``Регистрация успешно завершена``.

5. После регистрации заполняется поле необязательных данных (имя, отчество, дата рождения, организация и др.), вводится имя другого, ранее сделанного почтового ящика, на который будут переданы логин и пароль только что зарегистрированного почтового ящика.

Почтовый ящик создан, им можно пользоваться:

6. В окне Online Resource Center щелкнуть по тексту Внести изменения, ввести логин и пароль, в пункте Почта выбрать Написать письмо, заполнить графы От кого, Кому (указать почтовый ящик адресата), ввести текст письма, выбрать кодировку (для России принят код кои-8), нажать Отправить. Если адрес есть и верен, т.е. графа Кому заполнена правильно, появится текст ``Сообщение успешно отправлено``, если нет – ``Не могу послать сообщение``.

7. К письму можно присоединить с помощью браузера до пяти файлов размером около 1,5 М каждый, нажав на кнопку Обзор ниже текста письма. После выбора файлов нужно щелкнуть мышью по значку Скрепка, файлы присоединятся к письму, затем нажать Отправить.

8. Для чтения письма получатель должен в своём ящике нажать на строку Новое письмо или Чтение письма. Количество полученных корреспонденций сообщается в том же окне ящика.

9. Для выхода из электронной почты нужно нажать Завершить сеанс, иначе любой пользователь может с этого компьютера войти в Ваш ящик, не вводя логина и пароля, т.к. они уже введены, а сеанс не завершён.

Создание сайта в Internet связано с разработкой Web-страниц, что является сложной задачей, требующей знания языка программирования HTML, поэтому здесь не рассматривается

### **2.13 Лабораторная работа №13 (2 часа).**

**Тема:** «Обеспечение безопасности информации в информационных технологиях»

**2.13.1 Цель работы:** Применение основ информационной безопасности для имитации действий нарушителя по раскрытию (нарушению конфиденциальности) при использовании одного и того же одноразового блокнота (гаммы) на основе побитового сложения по модулю 2 (взлом двухразового блокнота).

#### **2.13.2 Задачи работы:**

1. Изучить теоретический материал.
2. Проанализировать проделанную работу и предложить свой метод противодействия реализованной атаке.

#### **2.13.3 Перечень приборов, материалов, используемых в лабораторной работе:**

1. Персональный компьютер
2. Операционная система Windows

#### **2.13.4 Описание (ход) работы:**

Краткие теоретические сведения Безопасность для различных сфер жизнедеятельности личности, общества и государства опирается на понятие ограничения доступа к информации. Шифрование используется как метод усиления таких свойств качественной информации, как конфиденциальность и целостность. Можно легко получить не раскрываемый шифр, но тут имеется одна тонкость. Сначала нужно найти процесс, который может генерировать произвольную бесконечную строку битов, которая называется гаммой. Во-вторых, необходимо преобразовать открытый текст в битовую строку и вычислить побитовое сложение по модулю 2 (операция XOR) открытого текста и ключевой строки. После чего можно послать результирующий зашифрованный текст получателю по незащищенному каналу. Шифрованный текст не может быть раскрыт, поскольку каждая возможная ключевая строка является одинаково вероятным кандидатом; нарушитель не имеет информации. Дополнительно к отправке зашифрованного текста отправитель передает ключевую строку по защищенному каналу получателю так, что получатель может расшифровать текст, обрабатывая XOR зашифрованного текста и ключевой строки. Хитрость состоит в том, что нам нужно иметь защищенный канал для отправки ключевой строки. Более практичный подход для отправителя: сгенерировать эту ключевую строку заранее. Например, отправитель может создать 1000 компакт-дисков, полных произвольных битов, и переправить их получателю на БТР. Хотя еженедельная посылка 1000 дисков и является широкополосной операцией, у нее есть один изъян: ключевой поток должен быть такой же длины, как и данные; если данные на один бит длиннее, чем ожидалось, то появятся проблемы. Строки однократного использования используются иногда на практике (например, однократные пароли). Обходной путь, часто применяющийся на практике, должен выбрать короткий ключ (скажем, 64 бита) и использовать псевдослучайный генератор чисел, чтобы генерировать ключевую строку из короткого ключа. Теперь нужно послать только 64 бита по защищенному каналу, но придется полагаться на некоторое искусство. А именно: нужен хороший псевдослучайный генератор, для которого по выходу нельзя догадаться о ключе, и для которого один и тот же выход всегда генерируется из данного короткого ключа. Но если нарушитель может управлять входом, то он может восстановить выход произвольного генератора и, возможно, угадать будущие случайные числа! Из этого следует, что трудно разработать хороший псевдослучайный генератор. Поэтому нужны альтернативные способы шифрования. Можно ли вычислить, какие два документа зашифровал Борис, используя один и тот же одноразовый блокнот (гамму) – побитовое сложение по модулю 2 (и что собой представляет этот одноразовый блокнот)?

Краткие теоретические сведения

Безопасность для различных сфер жизнедеятельности личности, общества и государства опирается на понятие ограничения доступа к информации. Шифрование используется как метод усиления таких свойств качественной информации, как



конфиденциальность и целостность. Можно легко получить не раскрываемый шифр, но тут имеется одна тонкость. Сначала нужно найти процесс, который может генерировать произвольную бесконечную строку битов, которая называется гаммой. Во-вторых, необходимо преобразовать открытый текст в битовую строку и вычислить побитовое сложение по модулю 2 (операция XOR) открытого текста и ключевой строки. После чего

можно послать результирующий зашифрованный текст получателю по незащищенному каналу. Шифрованный текст не может быть раскрыт, поскольку каждая возможная ключевая строка является одинаково вероятным кандидатом; нарушитель не имеет информации. Дополнительно к посылке шифрованного текста отправитель передает ключевую строку по защищенному каналу получателю так, что получатель может расшифровать текст, обрабатывая XOR шифрованного текста и ключевой строки.

Хитрость состоит в том, что нам нужно иметь защищенный канал для посылки ключевой строки. Более практичный подход для отправителя: сгенерировать эту ключевую строку заранее. Например, отправитель может создать 1000 компакт-дисков, полных произвольных битов, и переправить их получателю на БТР. Хотя еженедельная посылка 1000 дисков и является широкополосной операцией, у нее есть один изъян: ключевой поток должен быть такой же длины, как и данные; если данные на один бит длиннее, чем ожидалось, то появятся проблемы. Строки однократного использования используются иногда на практике (например, однократные пароли).

Обходной путь, часто применяющийся на практике, должен выбрать короткий ключ (скажем, 64 бита) и использовать псевдослучайный генератор чисел, чтобы генерировать ключевую строку из короткого ключа. Теперь нужно послать только 64 бита по защищенному каналу, но придется полагаться на некоторое искусство. А именно: нужен хороший псевдослучайный генератор, для которого по выходу нельзя догадаться о ключе, и для которого один и тот же выход всегда генерируется из данного короткого ключа. Но если нарушитель может управлять входом, то он может восстановить выход произвольного генератора и, возможно, угадать будущие случайные числа!

Из этого следует, что трудно разработать хороший псевдослучайный генератор. Поэтому нужны альтернативные способы шифрования. Можно ли вычислить, какие два документа зашифровал Борис, используя один и тот же одноразовый блокнот (гамму) – побитовое сложение по модулю 2 (и что собой представляет этот одноразовый блокнот)?

Известна следующая информация:

1. Шифрование использует коды ASCII со 128 возможными символами в любой позиции (хотя некоторые – значительно более вероятны, чем другие).
2. Используемый одноразовый блокнот был произведён псевдослучайным образом, со значениями в пределах от 0 до 127.
3. Использовался один и тот же блокнот, чтобы зашифровать оба исходных текста.
4. Если длины исходных текстов не совпадают, то более короткий из них дополняется пробелами.
5. Эти тексты - части относительно известных текстов на английском языке.

Шифр 1:

```
42 102 120 61 61 67 57 84 117 66 41 33 100 116 15 55 80 16 120 0 54
78 105 113 96 25 43 69 39 82 125 40 40 24 120 94 92 37 114 53 64 63
107 19 82 62 99 81 81 69 103 22 120 123 71 1 113 57 5 50 67 40 2 85
67 11 40 56 22 89 127 95 59 121 27 121 95 121 114 3 1 5 45 103 112
127 62 34 39 13 44 30 80 19 2 60 72 80 56 18 93 31 69 66 45 122 71
33 58 113 12 120 50 63 39 5 110 28 14 48 109 10 68 95 92 88 0 30 107
4 54 92 104 122 5 95 15 118 42 93 75 83 9 35 106 8 13 53 101 93 32
60 53 36 72 101 121 121 121 99 98 89 30 71 87 87 14 107 28 36 42 108
98 95 99 68 2 60
```

Шифр 2:

```
34 40 111 117 37 64 32 88 55 74 112 117 103 121 23 54 91 5 116 84 42
```

79 127 35 114 80 48 67 39 71 53 62 97 12 113 48 47 34 122 57 80 63  
 122 77 61 93 119 68 71 83 107 87 116 115 2 19 101 112 86 127 78 109 2  
 89 81 17 85 5 21 94 127 84 59 109 13 42 25 116 126 7 7 18 106 118  
 113 62 37 63 43 102 69 73 79 14 7 105 70 17 18 25 93 56 7 27 7 84  
 8 117 50 123 9 44 42 50 98 76 111 6 4 48 117 7 86 88 92 75 29 16  
 121 65 52 80 107 50 19 8 41 46 10 84 74 95 93 57 106 27 72 125 101  
 73 97 56 58 51 89 101 108 125 112 99 114 72 18 9 84 30 7 107 89 34  
 39 103 33 86 36 3 74 104

Программа должна без потерь расшифровывать приведенные файлы (включая список печаток). Дополнительным заданием может служить создание программы, которая генерирует зашифрованные тексты по заданным открытым текстам.

Контрольные вопросы

1. Кратко сформулируйте виды безопасности для соответствующих сфер жизнедеятельности личности, общества и государства.
2. В чем состоят источники угроз интересам личности?
3. В чем состоят источники угроз интересам общества?
4. В чем состоят источники угроз интересам государства?
5. Перечислите виды информации и основные методы ее защиты.
6. В чем состоят национальные интересы Российской Федерации в информационной сфере и их что собой представляет их обеспечение.
7. Раскройте понятие информационно-безопасного шифрования.
8. В чем состоит сложность использования симметричных криптографических систем?
9. В чем заключаются слабости решения с помощью псевдослучайных генераторов чисел?
10. Приведите несколько примеров применения одноразовых блокнотов.
11. Расскажите о методах противодействия данной атаке.

## 2.14 Лабораторная работа №14 (2 часа).

**Тема:** «Связанные таблицы. Расчет промежуточных итогов в электронных таблицах»

**2.14.1 Цель работы:** Связывание листов электронной книги. Расчет промежуточных итогов. Структурирование таблицы.

### 2.14.2 Задачи работы:

1. Рассчитать зарплату за декабрь и построить диаграмму. Создать итоговую таблицу ведомости квартального начисления заработной платы, провести расчет промежуточных итогов по подразделениям.

### 2.14.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. Персональный компьютер
2. Операционная система Windows

### 2.14.4 Описание (ход) работы:

1. Запустите редактор электронных таблиц Microsoft Excel и откройте созданный в Практической работе 1 файл «Зарплата».

2. Скопируйте содержимое листа «Зарплата ноябрь» на новый лист электронной книги (Правка/Переместить/Скопировать лист). Не забудьте для копирования поставить галочку в окошке Создавать копию.

3. Присвойте скопированному листу название «Зарплата декабрь». Исправьте название месяца в ведомости на декабрь.

	А	В	С	Д	Е	Ф	Г	Н	І
1	<b>Ведомость начисления заработной платы</b>								
2	за декабрь 2006 г.								
3	Табельный номер	Ф.И.О.	Оклад (руб)	Премия (руб)	Доплата	Всего начислено (руб)	Удержания (руб)	К выдаче (руб)	
4				27%	8%		13%		
5	200	Иванов	8266,00	2231,82	661,28	11159,10	1450,68	9708,42	
6	201	Петров	4568,00	1233,36	365,44	6166,80	801,68	5365,12	
7	202	Сидоров	9500,00	2565,00	760,00	12825,00	1667,25	11157,75	
8	203	Петров	6804,00	1837,08	544,32	9185,40	1194,10	7991,30	
9	204	Паньчук	6759,00	1824,93	540,72	9124,65	1186,20	7938,45	
10	205	Васин	4673,00	1261,71	373,84	6308,55	820,11	5488,44	
11	206	Сорокин	5677,00	1532,79	454,16	7663,95	996,31	6667,64	
12	207	Федорова	6836,00	1845,72	546,88	9228,60	1199,72	8028,88	
13	208	Светов	3534,00	954,18	282,72	4770,90	620,22	4150,68	
14	209	Меньшов	5789,00	1563,03	463,12	7815,15	1015,97	6799,18	
15	210	Козлов	4673,00	1261,71	373,84	6308,55	820,11	5488,44	
16	211	Титова	6785,00	1831,95	542,80	9159,75	1190,77	7968,98	
17	212	Шашкин	9100,00	2457,00	728,00	12285,00	1597,05	10687,95	
18	213	Жарова	6240,00	1684,80	499,20	8424,00	1095,12	7328,88	
19		Всего	89204,00	24085,08	7136,32	8424,00	15655,30	104770,10	
20									
21	Максимальный доход		11157,75						
22	Минимальный доход		4150,68						
23	Средний доход		7483,58						

Рис. 2.1 – Ведомость зарплаты за декабрь

4. Измените значение Премии на 46 %, Доплаты – на 8 %. Убедитесь, что программа произвела пересчет формул.

5. По данным таблицы «Зарплата декабрь» постройте гистограмму доходов сотрудников. В качестве подписей оси X выберите фамилии сотрудников. Проведите форматирование диаграммы. Конечный вид гистограммы приведен на рис. 2.2.

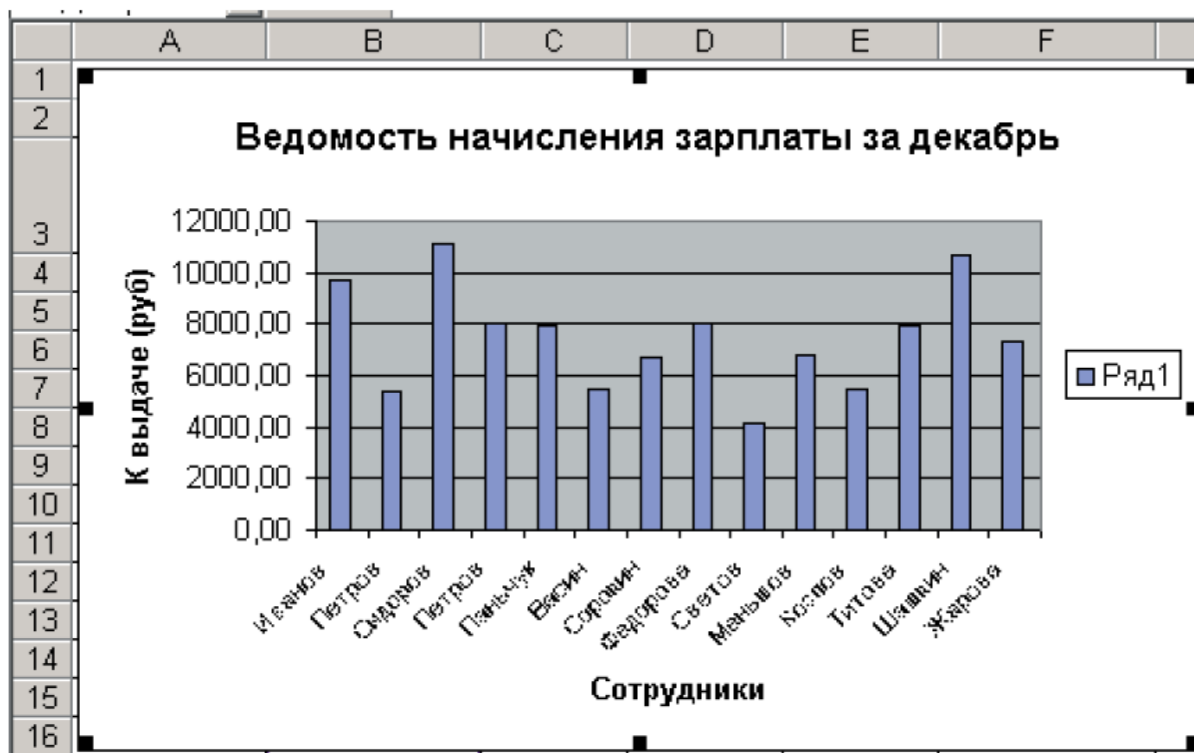


Рис. 2.2 – Гистограмма зарплаты за декабрь

6. Перед расчетом итоговых данных за квартал проведите сортировку по фамилиям в алфавитном порядке (по возрастанию) в ведомостях начисления зарплаты за октябрь–декабрь.

7. Скопируйте содержимое листа «Зарплата октябрь» на новый лист (Правка/Переместить/ Скопировать лист). Не забудьте для копирования поставить галочку в окошке Создавать копию.

8. Присвойте скопированному листу название «Итоги за квартал». Измените название таблицы на «Ведомость начисления заработной платы за 4 квартал».

9. Отредактируйте лист «Итоги за квартал» согласно образцу на рис. 2.3. Для этого удалите в основной таблице (см. рис. 2.1) колонки Оклада и Премии, а также строку 4 с численными значениями %Премии и % Удержания и строку 19 «Всего». Удалите также строки с расчетом максимального, минимального и среднего доходов под основной таблицей. Вставьте пустую третью строку.

10. Вставьте новый столбец «Подразделение» (Вставка/Столбец) между столбцами «Фамилия» и «Всего начислено». Заполните столбец «Подразделение» данными по образцу (см. рис. 2.3).

	A	B	C	D	E	F
1	<b>ВЕДОМОСТЬ НАЧИСЛЕНИЯ ЗАРАБОТНОЙ ПЛАТЫ</b>					
2	за 4 квартал 2006 г.					
3	Табельный номер	Ф.И.О.	Подразделение	Всего начислено (руб)	Удержания (руб)	К выдаче (руб)
4	200	Иванов	Отдел менеджмента	?	?	?
5	201	Петров	Отдел менеджмента	?	?	?
6	202	Сидоров	Отдел реализации	?	?	?
7	203	Петров	Бухгалтерия	?	?	?
8	204	Паньчук	Отдел реализации	?	?	?
9	205	Васин	Отдел реализации	?	?	?
10	206	Сорокин	Отдел менеджмента	?	?	?
11	207	Федорова	Отдел менеджмента	?	?	?
12	208	Светов	Бухгалтерия	?	?	?
13	209	Меньшов	Отдел реализации	?	?	?
14	210	Козлов	Отдел менеджмента	?	?	?
15	211	Титова	Бухгалтерия	?	?	?
16	212	Шашкин	Отдел реализации	?	?	?
17	213	Жарова	Отдел реализации	?	?	?

Рис. 2.3 – Таблица для расчета итоговой квартальной заработной платы

11. Произведите расчет квартальных начислений, удержания и суммы к выдаче как сумму начислений за каждый месяц (данные по месяцам располагаются на разных листах электронной книги, поэтому к адресу ячейки добавится адрес листа). Краткая справка. Чтобы вставить в формулу адрес или диапазон ячеек с другого листа, следует во время ввода формулы щелкнуть по закладке этого листа и выделить на нем нужные ячейки. Вставляемый адрес будет содержать название этого листа. В ячейке D5 для расчета квартальных начислений «Всего начислено» формула имеет вид = 'Зарплата декабрь'!F5 + 'Зарплата ноябрь'!F5 + 'Зарплата октябрь'!E5. Аналогично произведите квартальный расчет «Удержания» и «К выдаче».

Примечание. При выборе начислений за каждый месяц делайте ссылку на соответствующую ячейку из таблицы соответствующего листа электронной книги «Зарплата». При этом произойдет связывание информации соответствующих ячеек листов электронной книги.

12. Для расчета промежуточных итогов проведите сортировку по подразделениям, а внутри подразделений – по фамилиям. Таблица примет вид, как на рис. 2.4.

13. Подведите промежуточные итоги по подразделениям, используя формулу суммирования. Для этого выделите всю таблицу и выполните команду Данные/Итоги (рис. 2.5). Задайте параметры подсчета промежуточных итогов:

при каждом изменении в – Подразделение;

операция – Сумма;

добавить итоги по: Всего начислено, Удержания, К выдаче.

	H15	=				
	A	B	C	D	E	F
1	<b>ВЕДОМОСТЬ НАЧИСЛЕНИЯ ЗАРАБОТНОЙ ПЛАТЫ</b>					
2	за 4 квартал 2006 г.					
3	Табельный номер	Ф.И.О.	Подразделение	Всего начислено (руб)	Удержания (руб)	К выдаче (руб)
4	203	Петров	Бухгалтерия	27556,20	3582,31	23973,89
5	208	Светов	Бухгалтерия	14312,70	1860,65	12452,05
6	211	Титова	Бухгалтерия	27479,25	3572,30	23906,95
7	200	Иванов	Отдел менеджмента	33477,30	4352,05	29125,25
8	210	Козлов	Отдел менеджмента	18925,65	2460,33	16465,32
9	201	Петров	Отдел менеджмента	18500,40	2405,05	16095,35
10	206	Сорокин	Отдел менеджмента	22991,85	2988,94	20002,91
11	207	Федорова	Отдел менеджмента	27685,80	3599,15	24086,65
12	205	Васин	Отдел реализации	18925,65	2460,33	16465,32
13	213	Жарова	Отдел реализации	25272,00	3285,36	21986,64
14	209	Меньшов	Отдел реализации	23445,45	3047,91	20397,54
15	204	Паньчук	Отдел реализации	27373,95	3558,61	23815,34
16	202	Сидоров	Отдел реализации	38475,00	5001,75	33473,25
17	212	Шашкин	Отдел реализации	36855,00	4791,15	32063,85
18						

Рис. 2.4 – Вид таблицы после сортировки

Рис. 2.5 – Окно задания параметров расчета промежуточных итогов

Отметьте галочкой операции «Заменить текущие итоги» и «Итоги под данными». Примерный вид итоговой таблицы представлен на рис. 2.6.

14. Изучите полученную структуру и формулы подведения промежуточных итогов, устанавливая курсор на разные ячейки таблицы. Научитесь сворачивать и разворачивать структуру до разных уровней (кнопками «+» и «-»).

Краткая справка. Под структурированием понимается многоуровневая группировка строк и столбцов таблицы и создание элементов управления, с помощью которых легко можно скрывать и раскрывать эти группы.



F21			=ПРОМЕЖУТОЧНЫЕ.ИТОГИ(9;F4:F19)					
1	2	3	A	B	C	D	E	F
1			ВЕДОМОСТЬ НАЧИСЛЕНИЯ ЗАРАБОТНОЙ ПЛАТЫ					
2			за 4 квартал 2006 г.					
3			Табельный номер	Ф.И.О.	Подразделение	Всего начислено (руб)	Удержания (руб)	К выдаче (руб)
4	•		203	Петров	Бухгалтерия	27556,20	3582,31	23973,89
5	•		208	Светов	Бухгалтерия	14312,70	1860,65	12452,05
6	•		211	Титова	Бухгалтерия	27479,25	3572,30	23906,95
7					Бухгалтерия Всего	69348,15	9015,26	60332,89
8	•		200	Иванов	Отдел менеджмента	33477,30	4352,05	29125,25
9	•		210	Козлов	Отдел менеджмента	18925,65	2460,33	16465,32
10	•		201	Петров	Отдел менеджмента	18500,40	2405,05	16095,35
11	•		206	Сорокин	Отдел менеджмента	22991,85	2988,94	20002,91
12	•		207	Федорова	Отдел менеджмента	27685,80	3599,15	24086,65
13					Отдел менеджмента Всего	121581,00	15805,53	105775,47
14	•		205	Васин	Отдел реализации	18925,65	2460,33	16465,32
15	•		213	Жарова	Отдел реализации	25272,00	3285,36	21986,64
16	•		209	Меньшов	Отдел реализации	23445,45	3047,91	20397,54
17	•		204	Паньчук	Отдел реализации	27373,95	3558,61	23815,34
18	•		202	Сидоров	Отдел реализации	38475,00	5001,75	33473,25
19	•		212	Шашкин	Отдел реализации	36855,00	4791,15	32063,85
20					Отдел реализации Всего	170347,05	22145,12	148201,93
21					Общий итог	361276,20	46965,91	314310,29
22								

Рис. 2.6 – Итоговый вид таблицы расчета квартальных итогов по зарплате  
 15. Сохраните файл «Зарплата» с произведенными изменениями (Файл/Сохранить).

## 2.15 Лабораторная работа №15 (2 часа).

**Тема:** «Создание базы данных»

**2.15.1 Цель работы:** освоить процесс разработки типичной базы данных (используется пример с заданием «Поваренная книга») с использованием визуального интерфейса Microsoft Access.

### 2.15.2 Задачи работы:

1. В приложении Microsoft Access разработать и заполнить базу данных, предназначенную для представления данных, описанных в задании.
2. Разработка структуры базы данных в Microsoft Access.
  - 2.1. Нормализация исходных данных до 3-й нормальной формы.
  - 2.2. Создание нормализованных таблиц в Access.
  - 2.3. Определение связей между таблицами.
  - 2.4. Назначения для связей поддержки целостности в виде каскадного удаления связанных записей.
3. Заполнение таблиц данными для контроля правильности работы базы данных и клиентского приложения.

### 2.15.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. Персональный компьютер
2. Операционная система Windows

### 2.15.4 Описание (ход) работы:

На рис. 1 приведена схема базы данных «Поваренная книга». Детальная структура каждой из таблиц описана ниже.

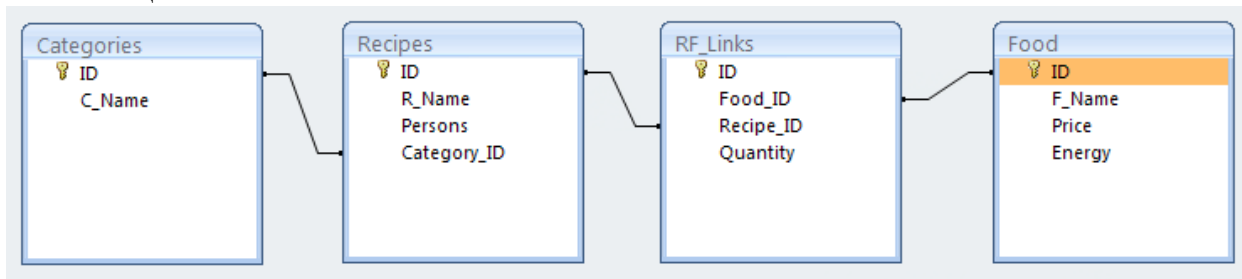


Рис. 1 Структура базы данных CookBook (Поваренная книга)

Таблица 1. Структура таблицы Categories (Категории продуктов)

Имя поля	Тип данных	Описание
ID	Счетчик	Идентификатор, первичный ключ
C_Name	Текстовый	Наименование категории

Таблица 2. Структура таблицы Recipes (Рецепты)

Имя поля	Тип данных	Описание
ID	Счетчик	Идентификатор, первичный ключ
R_Name	Текстовый	Наименование блюда
Persons	Числовой	Количество персон
Category_ID	Числовой	Идентификатор категории

Таблица 3. Структура таблицы RF\_Links (Связи «рецепты-продукты»)



Имя поля	Тип данных	Описание
ID	Счетчик	Идентификатор, первичный ключ
Food ID	Числовой	Идентификатор продукта
Recipe ID	Числовой	Идентификатор рецепта
Quantity	Числовой	Количество продукта в составе рецепта

Таблица 4. Структура таблицы Food (Продукты)

Имя поля	Тип данных	Описание
ID	Счетчик	Идентификатор, первичный ключ
F Name	Текстовый	Наименование продукта
Price	Денежный	Цена
Energy	Числовой	Калорийность

Вопросы для самопроверки

1. В чем необходимость установки связей между таблицами на уровне СУБД?
2. В чем преимущество нормализованной базы данных в случае вашего задания?
3. Обоснуйте ваш выбор типов данных для полей таблиц.

Задание

Поваренная книга

Структура

данных:

Таблица 1

#### Продукты

Наименование	Цена	Калорийность
--------------	------	--------------

Таблица 2

#### Рецепты

Название блюда	Список продуктов и их количества	Размер готовой порции (число персон)	Категория блюда: первое, второе, напиток и т.д.
----------------	----------------------------------	--------------------------------------	-------------------------------------------------

Функции:

1. Ведение таблиц.
2. Вывод списка продуктов и сметы для данного блюда на N персон.
3. Поиск цены по наименованию продукта.

## **2.16 Лабораторная работа №16 (2 часа).**

**Тема:** «Формирование запросов, отчетов и формы для однотабличной базы данных»

**2.16.1 Цель работы:** Изучить технологию формирования запросов, отчетов и форм для однотабличной базы данных.

### **2.16.2 Задачи работы:**

1. Изучить процесс формирования запросов на выборку
2. Освоить процесс формирования запросов
3. Создать отчет

### **2.16.3 Перечень приборов, материалов, используемых в лабораторной работе:**

1. Персональный компьютер
2. Операционная система Windows

### **2.16.4 Описание (ход) работы:**

Формирование запросов на выборку.

Запросы являются мощным средством обработки данных, хранимых в таблицах Access 97. С помощью запросов можно просматривать, анализировать и изменять данные из нескольких таблиц. Они также используются в качестве источника данных для форм и отчетов. Запросы позволяют вычислять итоговые значения и выводить их в компактном формате, подобном формату электронной таблицы, а также выполнять, вычисления над группами записей.

Запросы можно создавать самостоятельно и с помощью мастеров. Мастера запросов автоматически выполняют основные действия в зависимости от ответов пользователя на поставленные вопросы. Самостоятельно разработать запросы можно в режиме конструктора.

В Access 97 можно создавать следующие типы запросов:

запрос на выборку;  
запрос с параметрами;  
перекрестный запрос;  
запрос на изменение (запрос на удаление, обновление и добавление записей на создание таблицы);

запросы SQL (запросы на объединение, запросы к серверу, управляющие запросы, подчиненные запросы)

Запрос на выборку используется наиболее часто. При его выполнении данные, удовлетворяющие условиям отбора, выбираются из одной или из нескольких таблиц и выводятся в определенном порядке. Например, можно вывести на экран данные о фамилиях доцентов, стаж которых более 15 лет (на основе таблицы Преподаватели).

Примечание. Простые запросы на выборку практически не отличаются от фильтров. Более того, фильтры можно сохранять как запросы.

Можно также использовать запрос на выборку, чтобы сгруппировать записи для вычисления сумм, средних значений, пересчета и других действий. Например, используя запрос на выборку, можно получить данные о среднем стаже доцентов и профессоров (на основе таблицы Преподаватели).

Запрос с параметрами - это запрос, при выполнении которого в его диалоговом окне пользователю выдается приглашение ввести данные, на основе которых будет выполняться запрос. Например, часто требуются данные о том, какие дисциплины ведут преподаватели. Чтобы не создавать отдельные запросы по каждому преподавателю, можно создать один запрос с параметрами, где в качестве параметра будет использоваться фамилия преподавателя. При каждом вызове этого запроса вам будет предложено ввести фамилию преподавателя, а затем на экран будут выведены все поля, которые вы указали в запросе, например фамилия, имя, отчество преподавателя и читаемая им дисциплина.

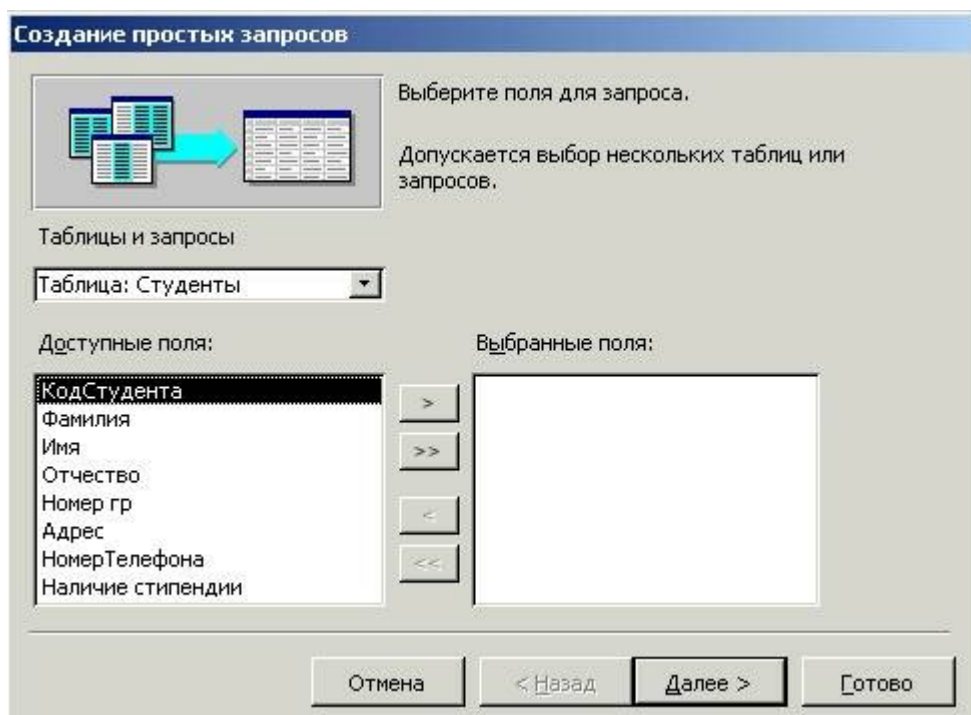


Рис. 1 – Окно выбора вариантов построения запросов

Для создания нового запроса надо в окне базы данных выбрать вкладку Запросы и щелкнуть по кнопке <Создать>. Откроется окно "Новый запрос", вид которого представлен на рис. 1.

В окне вы должны выбрать один из пяти пунктов: Конструктор, Простой запрос, Перекрестный запрос, Повторяющиеся записи. Записи без подчиненных. Конструктор позволит вам самостоятельно создать любой тип запроса, но этот режим рекомендуется пользователям, уже имеющим некоторый опыт создания запросов. Простой запрос позволит создать с помощью Мастера запрос на выборку из определенных полей таблиц или других запросов. (Это наилучший способ создания запроса для начинающих пользователей.) Что такое перекрестный запрос, рассматривалось выше. При выборе пункта Повторяющиеся записи будет создан запрос на поиск повторяющихся записей в простой таблице или в запросе, а при выборе пункта Записи без подчиненных - запрос на поиск записей, которым не соответствует ни одна запись в подчиненной таблице. Такой запрос используется для многотабличных баз данных.

У вас может возникнуть вопрос: как создавать запросы с параметрами и запросы на изменение, если при создании запроса они явно не указаны? Следует отметить, что основой для всех этих запросов является запрос на выборку, т.е. сначала необходимо определить набор данных, с которым хотите работать. Затем для созданного запроса на выборку надо перейти в режим конструктора. Задание параметров производится в строке Условия отбора для соответствующих полей. Подробнее это будет рассмотрено ниже при выполнении задания. Для доступа к запросам на изменение надо открыть пункт меню Запрос - в открывшемся списке вы увидите все виды запросов на изменение.

При выполнении запроса на выборку Access 97 извлекает записи из таблиц и формирует результирующий набор данных. Он выглядит, как таблица, хотя и не является ею. Результирующий набор данных является динамическим (или виртуальным) набором записей и не хранится в базе данных.

После закрытия запроса результирующий набор данных этого запроса прекращает свое существование. Хотя сам по себе динамический набор данных больше не существует, помните, что данные, которые в нем содержались, остаются в базовых таблицах.

При сохранении запроса остается только структура запроса - перечень таблиц, список полей, порядок сортировки, ограничения на записи, тип запроса и т.д. При

сохранении в базе данных запрос, по сравнению с результирующим набором данных, имеет ряд преимуществ:

на физическом носителе информации (обычно это жесткий диск) требуется меньший объем пространства;

запрос может использовать обновленные версии любых записей, измененных со времени последнего запуска запроса.

При каждом выполнении запрос обращается к базовым таблицам и снова создает результирующий набор данных. Поскольку сам по себе результирующий набор данных не сохраняется, запрос автоматически отображает любые изменения, происшедшие в базовых таблицах с момента последнего запуска этого запроса (даже в реальном времени в многопользовательской среде).

Для сохранения запроса следует выполнить следующие действия. Выполните команду Файл, Сохранить или щелкните по кнопке <Сохранить> на панели инструментов. Если вы впервые сохраняете запрос, введите новое имя запроса в диалоговом окне "Сохранение".

Формирование отчетов.

Отчет - это гибкое и эффективное средство для организации просмотра и распечатки итоговой информации. В отчете можно получить результаты сложных расчетов, статистических сравнений, а также поместить в него рисунки и диаграммы.

Пользователь имеет возможность разработать отчет самостоятельно или создать отчет с помощью мастера. Мастер по разработке отчетов выполняет всю рутинную работу и позволяет быстро разработать отчет. После вызова Мастера выводятся диалоговые окна с приглашением ввести необходимые данные, и отчет создается на основании ответов пользователя. Мастер необходим даже для опытных пользователей, так как позволяет быстро разработать макет, служащий основой создаваемого отчета. После этого можно переключиться в режим конструктора и внести изменения в стандартный макет.

При работе с мастером в зависимости от того, какой отчет вы хотите создать (т.е. как вы отвечаете на вопросы мастера) Access предлагает вам различные варианты макетов отчета. Например, если вы создаете простой отчет без группировки данных, то вам предлагается три варианта макета: в столбец, табличный и выровненный. При этом в небольшом окне представляется вид этих макетов. Если вы задаете уровни группировки (т.е. по каким признакам надо сгруппировать данные, например по должности), то вам предлагаются шесть видов макетов. Перечислять их не будем, так как они хорошо проиллюстрированы в окне создания макетов.

Основное различие между отчетами и формами заключается в их назначении. Если формы задуманы преимущественно для ввода данных, то отчеты - для просмотра данных (на экране либо на бумаге). В формах используются вычисляемые поля (обычно с помощью вычислений на основе полей в текущей записи). В отчетах вычисляемые поля (итоги) формируются на основе общей группы записей, страницы записей или всех записей отчета. Все, что можно сделать с формой (за исключением ввода данных), можно сделать и с отчетом. Действительно, форму можно сохранить в виде отчета, а затем изменить элементы управления формы в окне конструктора отчета.

Для создания отчета надо открыть вкладку Отчеты и щелкнуть по кнопке <Создать>. Откроется окно "Новый отчет", в котором приведены шесть пунктов меню, т.е. шесть способов создания отчета: Конструктор, Мастер отчетов, Автоотчет в столбец, Автоотчет ленточный, Мастер диаграмм и Почтовые наклейки. Конструктор позволит вам самостоятельно создать отчет, но это не просто даже для опытного пользователя. Мастер отчетов автоматически создаст отчет на основе выбранных вами полей таблиц (запросов) и макетов отчетов. Этот способ создания отчетов является наиболее удобным как для начинающих, так и для опытных пользователей. Автоотчет в столбец и Автоотчет ленточный - самые простейшие способы создания отчетов: достаточно указать только имя таблицы (запроса), на основе которого будет создан отчет, а остальное сделает мастер




отчетов. Мастер диаграмм поможет создать отчет в виде диаграммы, а Почтовые наклейки создадут отчет, отформатированный для печати почтовых наклеек.

#### ЗАДАНИЕ 1

Формирование запросов на выборку.

1. На основе таблицы Преподаватели создайте простой запрос на выборку, в котором должны отображаться фамилии, имена, отчества преподавателей и их должность.
2. Данные запроса отсортируйте по должностям.
3. Сохраните запрос.
4. Создайте запрос на выборку с параметром, в котором должны отображаться фамилии, имена, отчества преподавателей и преподаваемые ими дисциплины, а в качестве параметра задайте фамилию преподавателя и выполните этот запрос для преподавателя Гришина.

#### ТЕХНОЛОГИЯ РАБОТЫ

1. Для создания простого запроса:  
в окне базы данных откройте вкладку Запросы;  
в открывшемся окне щелкните по кнопке <Создать>;  
из появившихся пунктов окна "Новый запрос" выберите Простой запрос и щелкните по кнопке <ОК>;  
в появившемся окне в строке Таблицы/запросы выберите таблицу Преподаватели (если других таблиц или запросов не было создано, она будет одна в открывающемся списке);  
в окне "Доступные поля" переведите выделение на параметр Фамилия, щелкните по кнопке. Слово Фамилия перейдет в окно "Выбранные поля";  
аналогично в окно "Выбранные поля" переведите поля "Имя", "Отчество", "Должность" (порядок важен - в таком порядке данные и будут выводиться);  
щелкните по кнопке. <Далее>;  
в строке параметра Задайте имя запроса введите новое имя Должности преподавателей;  
щелкните по кнопке <Готово>. На экране появится таблица с результатами запроса.  
Для сортировки данных:  
щелкните в любой строке поля "Должность";  
отсортируйте данные по убыванию. Для этого щелкните по кнопке  на панели инструментов или выполните команду Записи, Сортировка, Сортировка по убыванию.
3. Для сохранения запроса: щелкните по кнопке  или выполните команду Файл, Сохранить;  
закройте окно запроса.
4. Для создания запроса на выборку с параметром:  
создайте запрос на выборку для следующих полей таблицы Преподаватели: "Фамилия", "Имя", "Отчество", "Преподаваемая дисциплина". Запрос создавайте аналогично тому, как это делалось в п.1;  
задайте имя запросу Преподаваемые дисциплины;  
щелкните по кнопке <Готово>. На экране появится таблица с результатами запроса;  
перейдите в режиме конструктора, щелкнув по кнопке или выполнив команду Вид, Конструктор;  
в строке параметра Условия отбора для поля "Фамилия" введите фразу (скобки тоже вводить): [Введите фамилию преподавателя]  
выполните запрос, щелкнув по кнопке  на панели инструментов или выполнив команду Запрос, Запуск;  
Примечание. Вышеописанным способом запрос выполняется только в режиме конструктора. Для того чтобы выполнить запрос из другого режима, надо открыть вкладку Запросы, выделить требуемый запрос и щелкнуть по кнопке <Открыть>.

в появившемся окне введите фамилию Гришин и щелкните по кнопке <ОК>. На экране появится таблица с данными о преподавателе Гришине - его имя, отчество и преподаваемая им дисциплина;

сохраните запрос;

закройте окно запроса.

## ЗАДАНИЕ 2

На основе таблицы Преподаватели создайте отчет с группированием данных по должностям.

### ТЕХНОЛОГИЯ РАБОТЫ

Для создания отчета:

Откройте вкладку Отчеты и щелкните по кнопке <Создать>;

в открывшемся окне выберите пункт Мастер отчетов;

щелкните по значку раскрывающегося списка в нижней части окна;

выберите из появившегося списка таблицу Преподаватели;

щелкните по кнопке <ОК>, В появившемся окне выберите поля, которые будут присутствовать в форме. В данном примере присутствовать будут все поля из таблицы, поэтому щелкните по кнопке щелкните по кнопке <Далее>;

в появившемся окне присутствует перечень полей. Переведите выделение на поле "Должность";

щелкните по кнопке. Таким образом вы задаете группировку данных по должности;

щелкните по кнопке <Далее>;

параметры появившегося окна оставим без изменений, поэтому щелкните по кнопке <Далее>;

в появившемся окне выберите стиль оформления отчета;

щелкните по кнопке. <Далее>;

в появившемся окне введите название отчета Преподаватели;

щелкните по кнопке <Готово>. На экране появится сформированный отчет;

просмотрите, а затем закройте отчет.

## **2.17 Лабораторная работа №17 (2 часа).**

**Тема:** «Обслуживание информационной системы средствами СУБД Access»

**2.17.1 Цель работы:** Ознакомиться с формированием информационных баз данных и основными командами СУБД ACCESS.

### **2.17.2 Задачи работы:**

1. Произвести набор Таблицы 1 в Word (имя файла tab1.doc).
2. Импортировать набранную таблицу в Excel (имя файла labis6-2.xls).
3. Создать с помощью конструктора БД (имя файла labis6-1.mdb) и произвести фильтрацию по условиям, а также запрос.

### **2.17.3 Перечень приборов, материалов, используемых в лабораторной работе:**

1. Персональный компьютер
2. Операционная система Windows

### **2.17.4 Описание (ход) работы:**

1. Изучить теоретическую часть работы по файлам, находящимся в каталоге этой лабораторной работы, и справочным материалам по программе ACCESS.

2. Создать Таблицу1 в MS WORD, импортировать набранную таблицу в EXCEL и присвоить имя файлу labis6-2.xls.

3. Запустить программу Microsoft ACCESS и открыть окно СУБД.

3.1. В появившемся окне диалога выберите режим «Новая база данных» и укажите имя файла labis6-1 в каталоге c:\temp\is. Каталог is используется для всей группы, поэтому файлы должны иметь имена, указывающие на номер варианта.

3.2. Создание таблицы в Microsoft ACCESS.

3.2.1. В окне базы данных выберите вкладку «Таблицы», нажмите кнопку «создать», выберите режим «конструктор». В макет таблицы (построчно) введите имена столбцов Таблицы 1 (формат данных для первого и последнего столбцов – текстовый, а для остальных – числовой, длинное целое, фиксированный).

3.2.2. Перейдите в «Режим таблицы» (п. Вид) с сохранением файла под именем «Нач\_таб» и подготовьте таблицу следующего содержания:

Если при вводе данных окажется, что дробная часть не отображается в таблице, то нужно переопределить поля 3-го и 4-го столбцов. Для этого выберите на вкладке таблицу «Нач\_таб», нажмите пиктограмму «Конструктор», выберите строку с именем «I, mA» и переместите курсор в нижнюю часть вкладки «Общие» на строку «Размер поля». Выберите значение «Одинарное с плавающей точкой». После изменений сохраните заполненную таблицу под тем же именем «Нач\_таб».

Таблица 1

Тип лампы	U заж, В	I, мА	Диаметр баллона, мм	Длина лампы, мм	Тип цоколя
СН-1	150	20	55	90	Р-27
СН-2	65	30	55	90	Р-27
МН-3	48	1	15	35	1Ш-12
МН-4	80	1,5	15	35	1Ш-12
МН-5	50	0,2	9	33	Р-10
МН-6	60	0,8	6,8	28	Нет

4. Перейдите в окно СУБД Access.

4.1. Импортируйте таблицу labis6-2.xls в базу данных labis6-1. Для этого выполните команду «Файл/Внешние данные/Импорт». В диалоговом окне «Импорт» на вкладке «Тип файла» выберите режим Microsoft EXCEL. (\*.XLS). Затем установите флажок «Первая строка – заголовки столбцов», откажитесь от самостоятельного описания полей, ключ не устанавливайте. Переименуйте таблицу в «Доп\_таб».

4.2. Объедините таблицы «Нач\_таб» и «Доп\_таб» с использованием команд п. Вставка.

4.2.1. Скопируйте записи таблицы «Доп\_таб» в буфер.

4.2.2. Перейдите в таблицу «Нач\_таб» и вставьте из буфера данные.

При вставке данных обратите внимание на то, что в таблице «Нач\_таб» для вставки записей из другой таблицы нужно выделять всю последнюю строку, кроме первого столбца «Код». Измените вновь вставленный столбец «I, мА» таблицы "Доп\_таб" следующим образом: каждое значение умножьте на номер варианта.

4.2.3. Отсортируйте по возрастанию значения в столбце - "I,мА" (см. Основное меню). Обратите внимание на изменения в столбце «Код». Объединенную таблицу переименуйте в «Неоновые лампы».

4.3. Проведите выборку данных методом фильтрации для результирующей таблицы «Неоновые лампы». Для этого выполните команду «Записи/Фильтр/Изменить фильтр». В окне фильтра задайте условия фильтрации: для столбца «Тип» напишите выражение – «МН\*», а для столбца «Uзаж» – «<= 60». Выполните команду «Применить фильтр». Сохраните таблицу (при этом созданный фильтр автоматически сохранится).

4.4. Создайте запрос по базе данных.



Для этого перейдите на вкладку «Запросы» окна базы данных labis6-1, выполните команду «Создать». В диалоговом окне «Новый запрос» укажите режим «Конструктор», а в окне «Добавление таблицы» выделите таблицу «Неоновые лампы» и щелкните на кнопке «Добавить». Закройте окно «Добавление таблицы». В строке «Поле» для первого столбца выберите – «Тип», для второго – «Uзж, В», для третьего – «I, мА», а в строке «Условие отбора» наберите «МН\*», «<90» и «<=1» соответственно. Сохраните запрос под именем "Запрос1". Выведите на экран результаты выборки, для чего выполните запрос.

5. Не выходя из СУБД ACCESS, предъявите преподавателю таблицы «Доп\_таб.», «Неоновые лампы» и «Запрос 1».

6. Удалите файлы labis6-1 и labis6-2 из каталога c:\temp\is.

Что нужно знать и уметь делать после выполнения практического задания

1. Как создать БД в MS Access.
2. Как создать таблицу в БД.
3. Как создать/изменить фильтр.
4. Как создать/изменить запрос.
5. Как выполнить сортировку записей в таблице.