



Федеральное агентство по образованию

Тульский государственный университет

КАФЕДРА ТЕХНОЛОГИИ МАШИНОСТРОЕНИЯ

Илюхин А.Ю., доцент, к.т.н.

## ПРИКЛАДНАЯ ИНФОРМАТИКА

### КОНСПЕКТ ЛЕКЦИЙ

(17 часов)

для бакалавров по направлению 150900 – «Технология, оборудование и  
автоматизация машиностроительных производств»

и студентов специальностей:

151001 «Технология машиностроения»

150401 «Проектирование технических и технологических комплексов»

151002 «Металлообрабатывающие станки и комплексы»

151003 «Инструментальные системы машиностроительных производств»  
очной формы обучения

Тула 2005

## Лекция 1 (2 часа)

*Классы наиболее типичных программных средств*

*Средства обработки текстовой информации*

*Элементы работы в среде пакета Ms Word 2000*

*Создание, редактирование и печать документов*

*Настройка панелей инструментов Ms Word 2000*

*Программа Acrobat*

### ВВЕДЕНИЕ

Для массового пользователя ЭВМ наибольший интерес представляет именно *прикладное программное обеспечение* (ППО), позволяющее непосредственно решать задачи из его *предметной* области. Учитывая разнотипность решаемых на ЭВМ задач и соответствующего им ППО, рассмотрим **классы** наиболее типичных *программных средств* (ПС), ориентированных на широкое применение.

В настоящее время компьютеры находят широкое применение в административном управлении. Одной из таких сфер применения компьютеров является *электронный офис*. Это система автоматизации работы учреждения, основанная на использовании компьютерной техники и средств связи. В нее обычно входят такие компоненты, как:

- текстовые редакторы;
- графические редакторы и графические библиотеки (для получения диаграмм, схем, графиков и др.);
- электронные таблицы;
- системы управления базами данных;
- интегрированные пакеты программ;
- электронные записные книжки;
- электронные календари с расписанием деловых встреч, заседаний и др.;
- электронные картотеки, обеспечивающие каталогизацию и поиск документов (писем, отчетов и др.) с помощью компьютера;
- автоматические телефонные справочники, которые можно листать на экране, установить курсором нужный номер и соединиться.

Эти ПС в значительной мере определяют основные направления *прикладной компьютерной информатики*. В этой связи выбраны следующие классы ПС: обработка *текстовой, табличной и графической* информации; *численных и символьных вычислений*; СУБД, СУБЗ и *экспертные системы*, а также *средства компьютерной телекоммуникации*.

Рассмотрим общие характеристики средств данных групп по наиболее типичным их представителям с акцентом на средства, функционирующие, в первую очередь, на IBM-совместимых ПК.

## 1 СРЕДСТВА ОБРАБОТКИ ТЕКСТОВОЙ ИНФОРМАЦИИ

Средства данного класса являются одной из составляющих компонент автоматизации учрежденческой деятельности в самом широком смысле ее понимания, позволяя пользователю с максимальными удобствами создавать высококачественные документы различного назначения посредством, в первую очередь, ПК и соответствующего ПО. В настоящее время существует достаточное разнообразие подобных систем, начиная от простейших текстовых Редакторов, ориентированных на работу с простыми ПК, и кончая специализированными издательскими системами, обеспечивающими работу большого издательского коллектива с использованием режима телеобработки и других современных издательских технологий.

Однако средства обработки текстовой информации, несмотря на их весьма широкий спектр, обеспечивают следующие основные функции:

- создание и редактирование текстового документа, включая:  
вставки, удаления, копирование, перемещение текста в документе, поиск и замену элементов документа, добавление в документ текстовой и графической информации;
- форматирование и распечатку документов с выбором размеров бумаги и форматов, а также с указанием числа копий и выводимой части документа;
- выравнивание документа и/или его отдельных частей по указанным границам с автоматической обработкой переносов строк;
- возможность создания документа по стандартному шаблону;
- использование различных шрифтов распространенных алфавитов и т.д.;
- размещение в документе таблиц, диаграмм, рисунков и т.д., а также ряд других функций, состав которых определяется уровнем и назначением конкретного ПС. В настоящем разделе кратко остановимся на текстовых процессорах, широко используемых на IBM-совместимых ПК для организации обработки различного рода текстовой информации, акцентируя внимание на наиболее популярных пакетах - *Ms Word 6.0*, *Word 7.0*, *Word 97* фирмы *Microsoft*

### 1.1 Элементы работы в среде пакета Ms Word

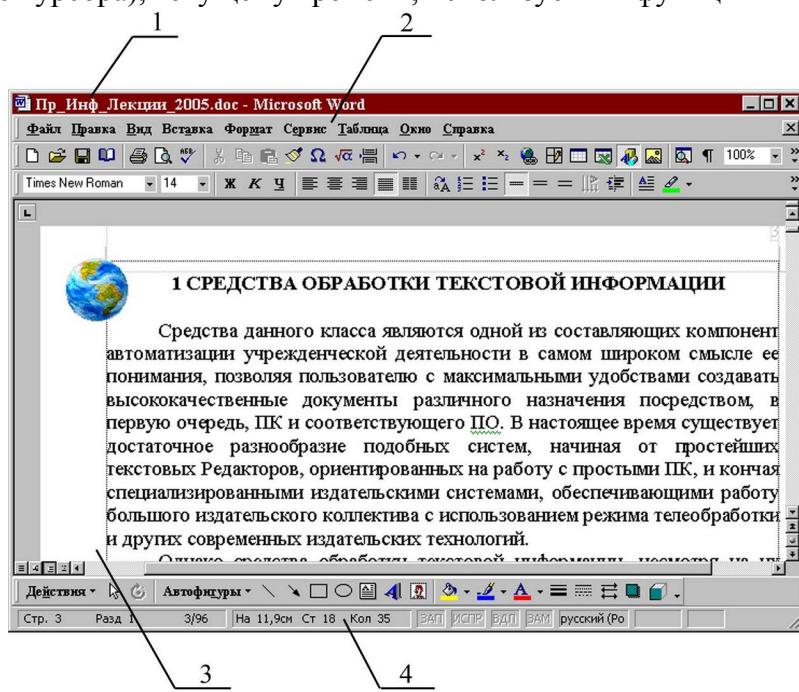
**1. Общие сведения.** Обработка **текстовых** документов - наиболее массовый вид прикладной деятельности на ЭВМ.

Пакет ***Ms Word 2000*** (в дальнейшем просто *Word* или *пакет*) представляет собой относительно более ранних средств данного типа достаточно большую программную систему.

Перед использованием Word рекомендуется закрывать ненужные при работе с ним Windows-приложения и удалять из ОП резидентные программы, например Norton Commander.

**2. Главные окно и меню пакета Word.** После активации пиктограммы **Word** двойным щелчком мыши производится загрузка резидентной части пакета с выходом на его главное окно. Главное окно пакета (рисунок 1.1) содержит четыре основных поля:

- первая строка (1) содержит имя пакета и имя активного в данный момент документа, а также кнопки для управления режимом визуализации окна (увеличения, уменьшения, сведения к пиктограмме), завершения работы с пакетом и др.;
- вторая строка (2) содержит главное меню пакета (ГМП) и управляющие кнопки, аналогичные упомянутым; детальное их описание можно найти в литературе по пакету или Windows;
- третье поле (3) представляет собой собственно рабочее окно текущего документа, через которое пользователь не только имеет возможность просматривать весь документ, но и проводить сам процесс создания/редактирования его;
- наконец, четвертая строка (4) содержит справочную информацию по: текущему документу (номер просматриваемой страницы, общее число страниц, местоположение курсора), текущему времени, используемым функциям пакета и др.



*Рисунок 1.1 – Главное меню пакета MS Word*

Дополнительно к указанным рабочее окно документа обрамляется рамками, содержащими указатели, позволяющие посредством активации их мышью перемещать окно по текущему документу.

**Главное меню пакета** (ГМП) содержит 9 групп функций. Для выбора требуемой функции пакета активируется одинарным щелчком мыши соответствующая группа ГМП, раскрывая подокно, содержащее список функций.

Прежде всего, средства меню «Справка» предоставляют не только справочную информацию по всем возможностям пакета, но и предлагают демонстрационные примеры, позволяющие лучше уяснить основные возможности пакета и принципы их использования.

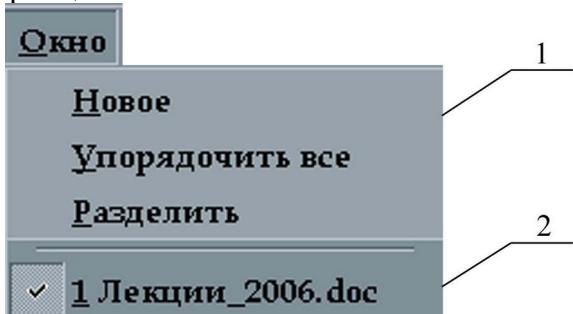


Рисунок 1.2 – Меню «Окно»

Меню «Окно» (рисунок 1.2) содержит две секции: (1) функции (*Новое*, *Упорядочить все*, *Разделить*) реорганизации окон пакета, включая разделение окон на подокна и создание новых окон с одинаковыми документами; и (2) *список документов*, находящихся в рабочей области пакета и к которым можно легко обращаться, активируя их окна через соответствующие элементы данного списка.



Рисунок 1.3 – Меню «Правка»

При необходимости перенести некоторый блок информации из неактивного документа открывается подокно, в списке документов (вторая секция) выбирается мышью нужный документ с его активацией и открытием окна и в нужном месте выбирается необходимый блок информации. Посредством функции копирования (*Ctrl+C*) меню «Правка» (рисунок 1.3) выбранный блок копируется в *системный буфер обмена*, вновь через список документов меню «Окно» активируется предыдущее окно (документ) и по функции «*Вставить*» (*Ctrl+V*) меню «Правка» в нужное место документа копируется хранящаяся в буфере информация (смотри рисунок 1.3).

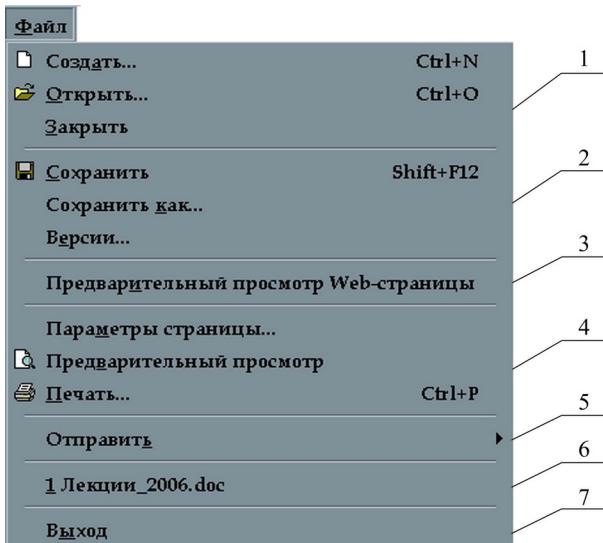


Рисунок 1.4 – Меню «Файл»

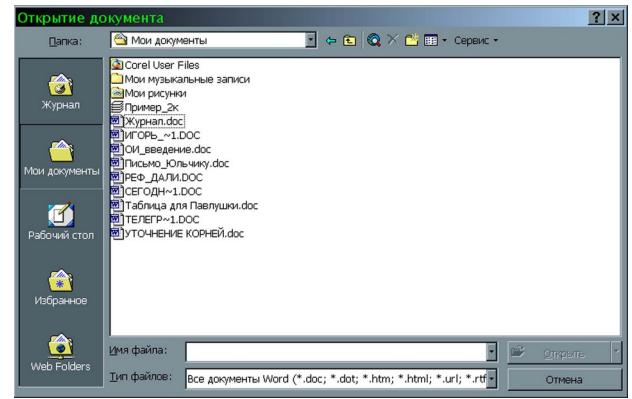


Рисунок 1.5 – Меню «Открытие документа»

Средства меню «Файл» ГМП разбиты на 7 секций по своему функциональному назначению (рисунок 1.4). Первая секция (1) содержит функции: создания нового (*Создать*; *Ctrl+N*), открытие существующего (*Открыть*; *Ctrl+O*) и закрытие текущего (*Закрыть*) документа.

При открытии существующего файла открывается окно «*Открытие документа*» (рисунок 1.5), поля и переключатели которого позволяют выбирать нужный файл из предлагаемого списка, обеспечивать поиск нужного файла, определять его формат по расширению имени и т.д.

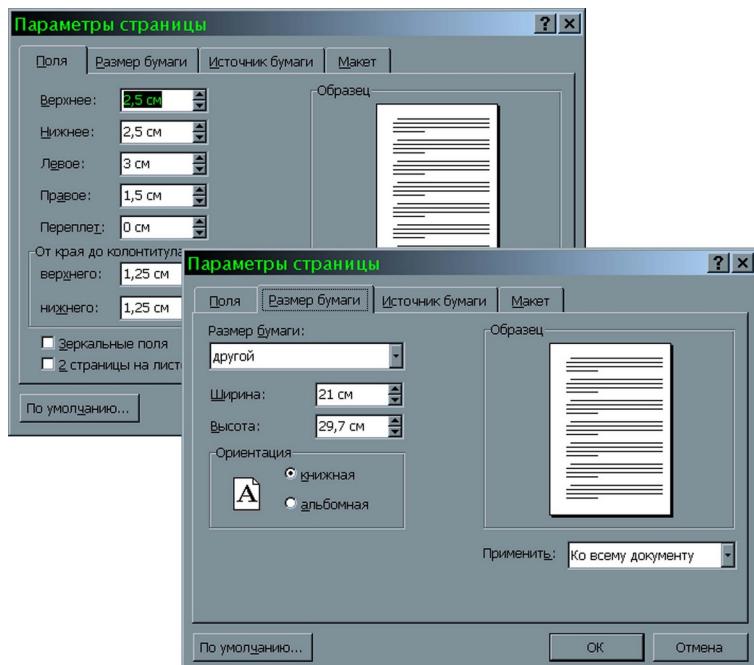
Вторая секция (2) содержит функции сохранения: текущего документа с обновлением исходного файла (*Сохранить*, *Ctrl+S*), текущего документа в новом файле и/или в новом формате (*Сохранить как*) и функцию «*Версии*». По функции «*Сохранить как*» предоставляется возможность не только сохранять текущий документ согласно новому спецификатору файла, но и в нужном формате, определяемом списком из 23 допустимых, включая txt- и WordPerfet-форматы.

Средства третьей секции (3) позволяют предварительно просмотреть созданную Web-страницу.

Средства четвертой секции (4) связаны с: установкой параметров страниц и просмотра текущего документа, обеспечением вывода текущего документа на печать.

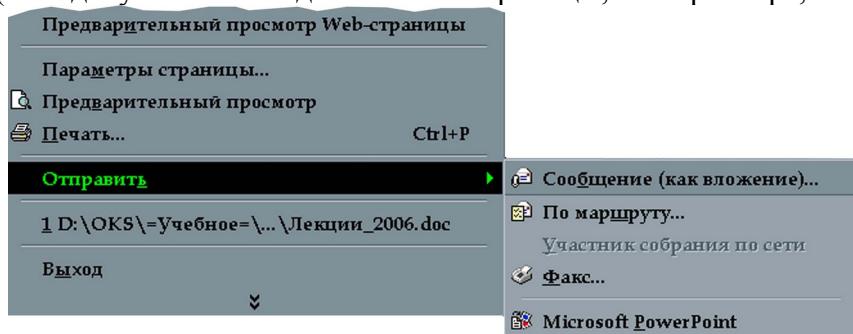
По функции «*Предварительный просмотр*» предоставляется возможность постранично просматривать файл, отслеживая его форматирование относительно заданных выходных параметров: размеры страницы и полей, расположение нумерации, сносок, типа принтера и др.

По функции «*Параметры страницы*» открывается окно, поля которого позволяют определять размеры выходной страницы текущего документа при выводе на печать, поля и режим расположения документа на странице, нумерации страниц и т.д (рисунок 1.6).



*Рисунок 1.6 – Меню «Файл», подменю «Параметры страницы»*

По функции «Печать» (*Ctrl+F*) открывается стандартное для Windows окно, позволяющее определять режим непосредственного вывода документа на печать (весь документ или отдельные его страницы, тип принтера, количество копий и т.д.).



*Рисунок 1.7 - Меню «Файл», подменю «Отправить»*

Средства пятой секции (5) меню «Файл» (*Отправить*) обеспечивают отправку подготовленного документа по электронной почте, факсом и т.д (рисунок 1.7).

В шестой секции (6) находятся имена последних открытых или созданных файлов.

Последняя седьмая секция (7) обеспечивает выход из среды пакета в Windows-среду; при этом запрашивается санкция на сохранение текущего документа (*Да*), без сохранения (*Нет*) и отмену выхода (*Отмена*) из пакета.

## 1.2 Создание, редактирование и печать документов.

После активации пиктограммы, *Word* выходит на главное окно пакета, содержащее ГМП. Если до того пакет не использовался в требуемых пользователю условиях, необходима его предварительная настройка, основные элементы которой сводятся к следующему.

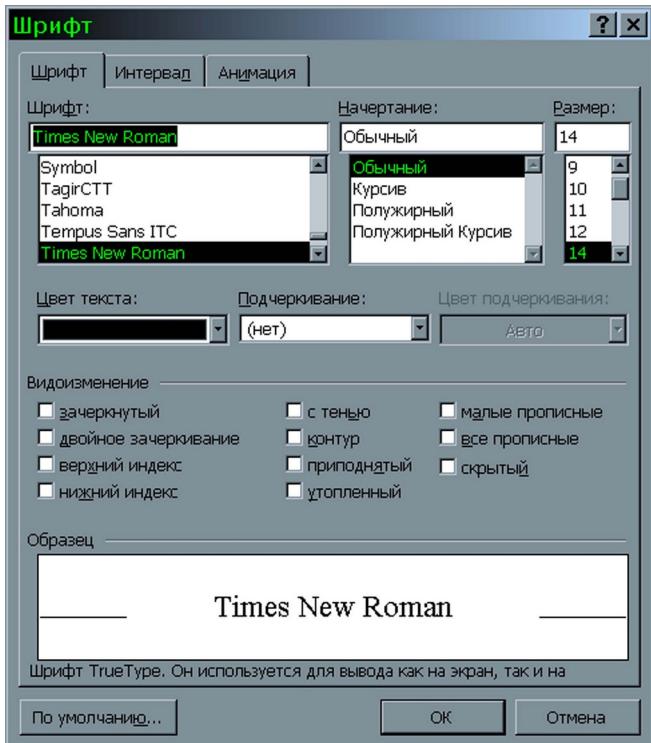


Рисунок 1.8 – Установка шрифта: меню «Формат», подменю «Шрифт»

После этого определяется режим нумерации страниц создаваемого документа (если таковая требуется), что обеспечивают средства меню «Вставка» подменю «Номера страниц». При этом стиль оформления нумерации можно определять посредством функции «Формат» (рисунок 1.9).

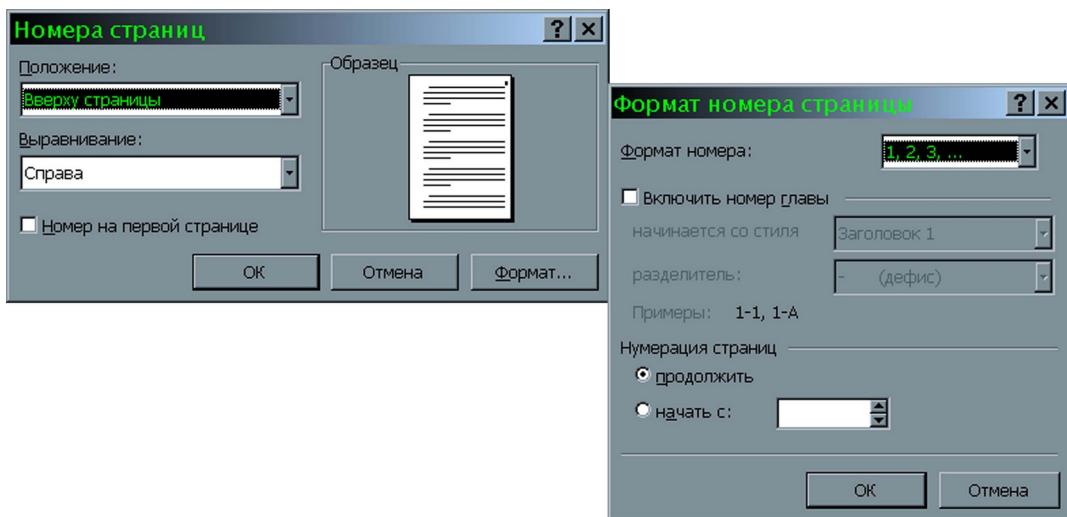


Рисунок 1.9 – Подменю «Номера страниц», формат номера страницы

Формат выходной страницы документа определяется при помощи меню «Файл», подменю «Параметры страницы» (смотри рисунок 1.6), открывающей соответствующее окно с опциями и кнопками. По опции

Прежде всего, определяется используемый шрифт; из практических соображений рекомендуется выбирать шрифт *Times New Roman* размера 14 и *Normal*-стиля (рисунок 1.8).

Если нажать на кнопку «По умолчанию» данная установка сохраняется в пакетном файле *normal.dot*, делая ее глобальной для всего последующего использования пакета.

Установка отображается, как правило, и в строке-меню окна пакета.

«Размер бумаги» выбирается формат А4 (210x297 мм), а по опции «Поля» определяются размеры полей (верхнее и нижнее – 2,5 см, левое 3 см, правое – 1,5 см). Сделанные установки по кнопке «По умолчанию» определяются глобальными с сохранением их в пакетном файле *normal.dot*.

Несколько сложнее обстоит дело при необходимости импортировать документы, подготавливаемые в среде приложений MS-DOS. В этом случае выполняется, следующая цепочка операций. В меню «Файл» подменю «Открыть» в качестве типа открываемых файлов необходимо установить «Все файлы». В появившемся меню «Преобразование файла» выбираем вид кодировки документа. В окне «Просмотр» появляется фрагмент документа в выбранной кодировке (рисунок 1.10), что позволяет визуально контролировать правильность выбора.

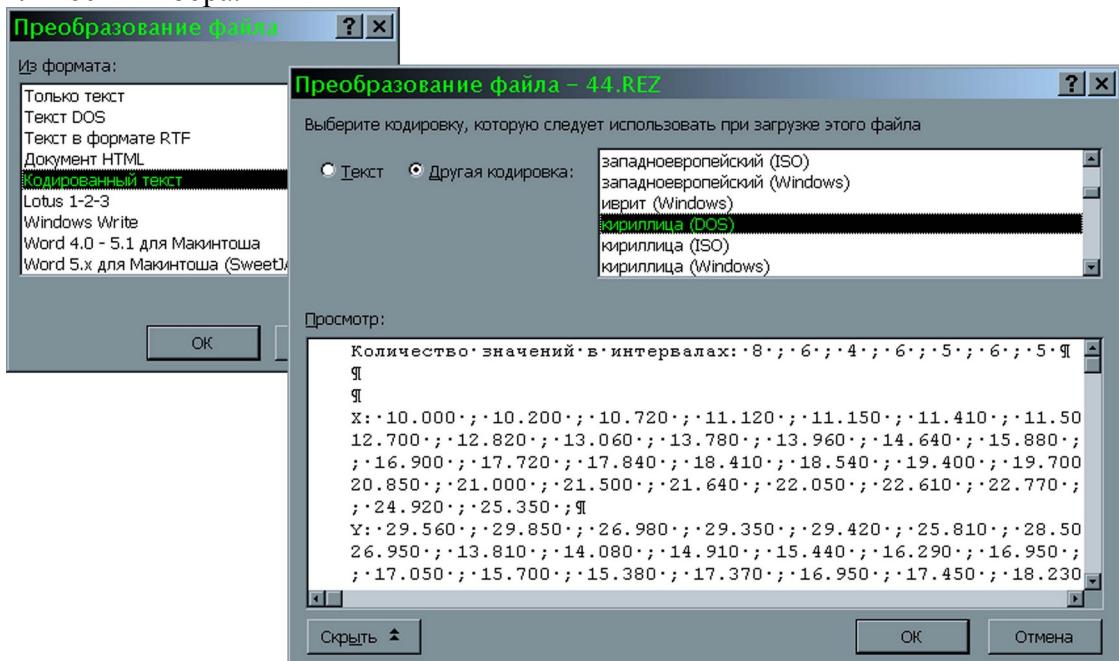


Рисунок 1.10 – Импорт документа в кодировке, отличной от MS Word

### 1.3 Настройка панелей инструментов Ms Word

Очевидно, что чем удобнее расположены рабочие панели и иконки инструментов на них, тем быстрее и эффективнее вы будете работать с программой. Начиная с версии *Word 2000* в этом текстовом процессоре реализован принцип автоматически настраиваемых меню (технология *IntelliSense*). Технология заключается в том, что система самостоятельно отслеживает, какими инструментами вы пользуетесь, и именно их оставляет на панелях меню.

В ниспадающих меню появляются сначала наиболее часто используемые команды (по стрелочке можно открыть полный список). Это нововведение доступно также и в версии *Word 2002*, однако, как показывают отзывы пользователей, подобная «излишняя интеллектуальность» не всем по нраву и некоторых раздражает ситуация, когда на панели ищешь кнопку,

которая только что была, а потом почему-то была убрана системой без вашего на то согласия.

Чтобы отказаться от данной услуги, необходимо выполнить команду *Сервис → Настройка* и выбрать вкладку *Параметры*. Если вы хотите всегда показывать полное меню, поставьте флажок в соответствующем пункте (рисунок 1.11).

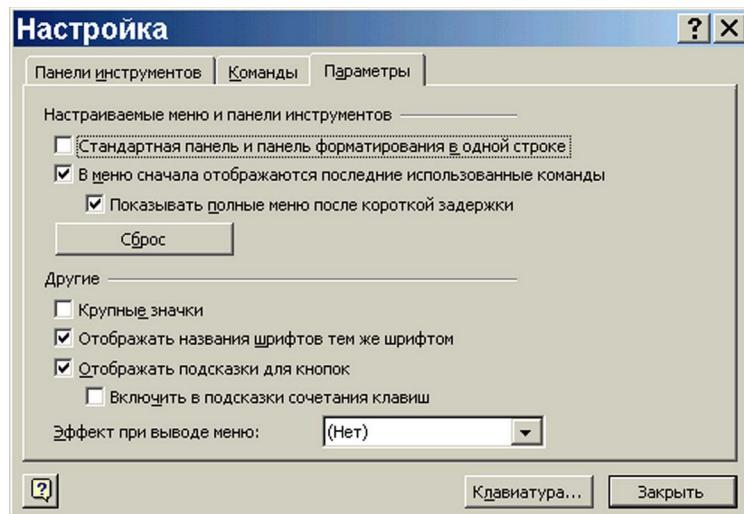


Рисунок 1.11 – Меню «Настройка», закладка «Параметры»

В этой же вкладке вы можете настроить, будут ли у вас отображаться панели Стандартная и Форматирование в одну строку или в две, варьируя параметр «Стандартная панель ...».

Для того чтобы просмотреть все доступные вам панели, перейдите во вкладку «Панели инструментов». Выставляя флажки в окошках, вы можете вывести на экран различные панели и ознакомиться с их функциональностью (рисунок 1.12).

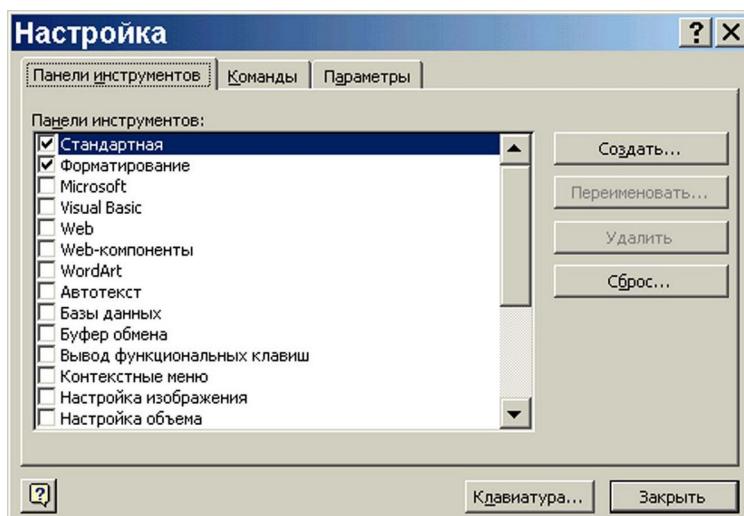
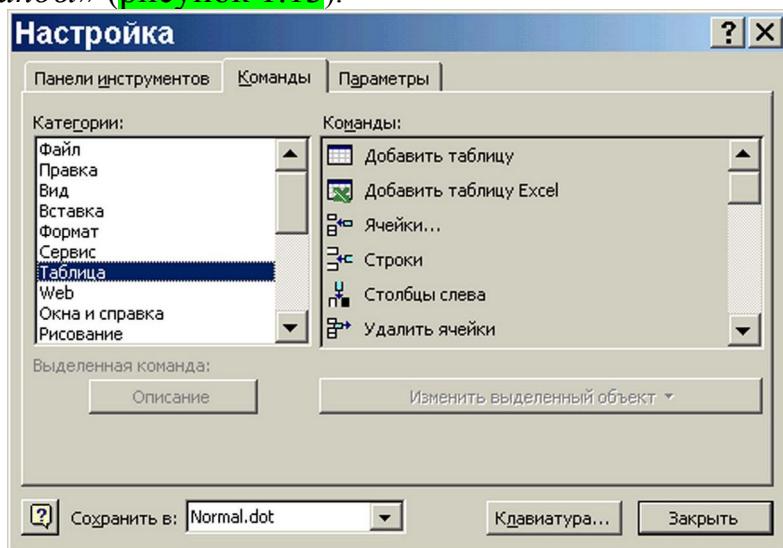


Рисунок 1.12 – Меню «Настройка», закладка «Панели инструментов»

Панели можно свободно перемещать по экрану. Например, если вы хотите за счет горизонтальной составляющей увеличить размер экрана по вертикали, любую панель можно переместить в боковую область. Для этого необходимо зацепить мышью за левый край панели (при этом возникает четырехнаправленная стрелка) и оттащить, куда пожелаете.

В тот момент, когда на экране находится панель «Параметры» (см. рисунок 1.11), вы можете перемещать иконки панелей инструментов в режиме drag-and-drop (независимо от того, какая вкладка открыта). Так, можно перетащить ненужный инструмент (удерживая левую кнопку мыши) на поле текста - он исчезнет из панели.

В принципе, кнопки можно также добавлять на любую панель по вашему желанию. Выполните команду *Сервис → Настройка* и выберите вкладку «Команды» (рисунок 1.13).



*Рисунок 1.13 – Меню «Настройка», закладка «Команды»*

Данная вкладка дает доступ ко всем иконкам-командам. Просмотреть их очень легко: выбираете категорию, затем щелкаете по доступным иконкам и читаете их описание. Для того чтобы добавить на какую-либо рабочую панель иконку-команду, выберите нужную и перетащите с панели «Команды» на выбранную панель. При перетаскивании иконки на панели будет возникать вертикальный разделительный знак, показывающий, между какими иконками будет помещена данная кнопка при отпускании мыши.

Помимо перемещения кнопок внутри панелей вы можете также изменить кнопки на такие, которые доступны в каталоге, и даже нарисовать собственные.

Выполните команду *Сервис → Настройка*, выберите вкладку «Команды» и щелкните правой кнопкой мыши по той иконке, внешний вид которой вы хотите изменить. В результате появится меню (рисунок 1.14), в котором следует выбрать пункт «Выбрать значок для кнопки» (сменить изображение на кнопке).

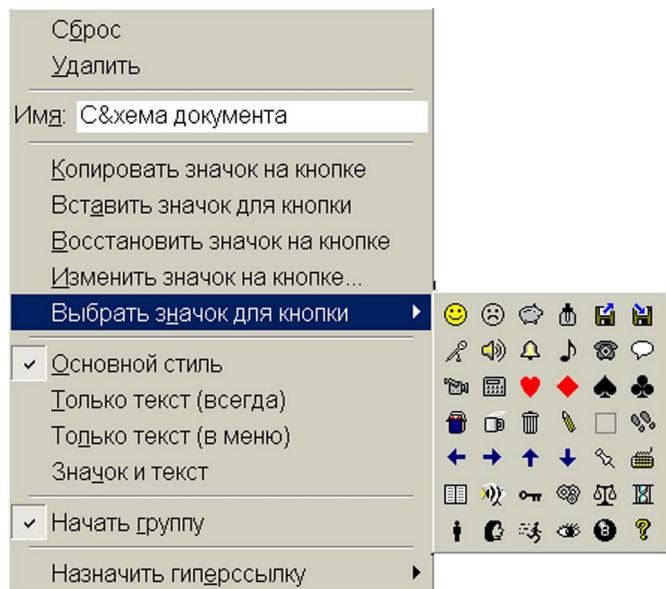


Рисунок 1.14 – Меню выбора значка

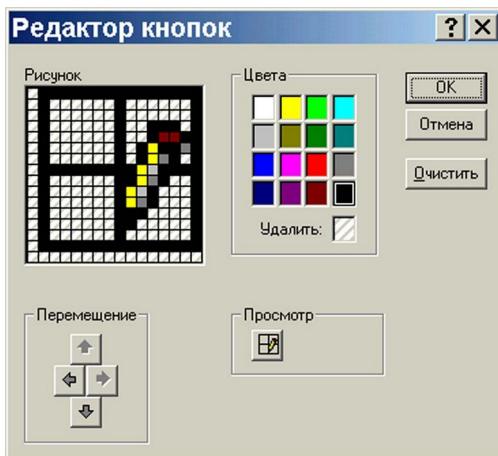


Рисунок 1.15 – Редактор изменения значков

Также можете нарисовать собственный значок или отредактировать существующий. На рисунке 1.15 показан встроенный редактор, который достаточно удобен для этой цели.

Целесообразно будет сделать свою собственную (пользовательскую) панель и поместить на нее те иконки-команды, которые вам необходимы в первую очередь.

#### 1.4 Программа Acrobat

В настоящее время при интенсивном развитии компьютерной телекоммуникации весьма актуальной является проблема использования так называемых **электронных** публикаций, а также подготовки такого типа материалов для широкого распространения в глобальных информационных компьютерных сетях. Для этих целей существует ряд развитых средств, из которых можно выделить популярный пакет **Acrobat** фирмы **Adobe Systems Inc.** Пакет обеспечивает доступ к электронным документам PDf-формата (*Portable Document Format*) в их оригинальной форме независимо от

компьютерной платформы, позволяя отыскивать, читать, редактировать, печатать и создавать файлы данного формата, широко используемого при организации систем электронных публикаций. Для нормального функционирования пакет относительно IBM-совместимых ПК требует процессор не ниже *Intel80386*, ОП не менее 4 Мбайт и доступной памяти на НМД не менее 4 Мбайт, а также *Windows* версии не ниже 3.1. В настоящее время пакет *Acrobat* поставляется в двух основных вариантах:

- *Acrobat Exchange*, обеспечивающий все основные потребности массового пользователя для работы в системах электронных публикаций и включающий средства: создания, редактирования (*Writer*), просмотра (*Viewer*), чтения (*Reader*) и индексирования (*Catalog*) *PDF*-документов, а также поиска информации в индексированных документах.
- *Acrobat Pro*, дополнительно к предыдущему включает средства конвертации *PostScript-файлов* в *PDF-формат*, данное средство поддерживает как локальный, так и сетевой режимы работы ПК на платформах *Windows* и *Macintosh*.

## Лекция 2 (1 час)

*Средства обработки графической информации*

*Типы визуализации графических изображений*

*Основные методы формирования графических объектов*

## 2 СРЕДСТВА ОБРАБОТКИ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

Развитию программных *графических* средств способствовал целый ряд важных как *мотиваций*, так и *предпосылок*. *Мотивации* широкого использования графического представления информации, как наиболее информативного и понятного человеку, весьма разнообразны и стимулируются многочисленными приложениями (научными, инженерно-техническими, медицинскими, архитектурно-изобразительными, художественными, игровыми, организации человеко-машинного интерфейса и т.д.). Среди *предпосылок* следует отметить развитие аппаратных средств отображения графической информации (в первую очередь дисплеев *растрового* типа) и возможностей ЭВМ - объема ОП и производительности ЦП. Наряду с этим развитию графического ПО способствовало создание целого ряда подходов и методов компьютерной обработки графической информации (трассировка лучей, фрактальная геометрия, конструктивная геометрия сплошных тел и др.), позволивших разработать целый ряд интересных графических ПС различного назначения, как для *векторных*, так и для *растровых* дисплейных систем.

### 2.1 Типы визуализации графических изображений

Используемые методы обработки графической информации определяются аппаратными средствами отображения ее на экране/плоттере. В настоящее время дисплейные системы делятся на два основных типа: *векторные* и *растровые*. В *векторных* системах световой луч движется по экрану вдоль рисуемой по определенному алгоритму линии. Тогда как в *растровых* системах нужный объект воспроизводится посредством последовательного сканирования световым лучом его *шаблона*, т.е. без вычерчивания каждой линии непрерывным движением. Таким образом, между обоими типами визуализации ГО лежит принципиальное различие: (1) *векторный* подход характеризуется (*динамическим* формированием на экране объекта по его программному описанию, сформированному посредством *графических примитивов*), (2) *растровый* подход отображает на экран весь объект целиком на основе его *шаблона*, созданного посредством *графических примитивов* и находящегося в *видеопамяти* дисплея.

На дисплейных системах *векторного* типа работа с ГО включали, как правило, следующие основные этапы: (1) программное создание ГО посредством графических примитивов, (2) формирование его снимка

проекции из определенной точки наблюдения и отображение данного **снимка** на экран дисплея, (3) программное редактирование ГО Созданное для векторных дисплеев графическое ПО, воспринимая ряд *растровых* примитивов (на пример массивов пикселов), работает в системе *мировых* координат с заданными пользователем объектами.

Принципиально иной подход использует *рисующее* графическое ПО, ориентированное на работу с *растровыми* дисплейными системами и получившее весьма широкое распространение в связи с использованием именно *растровых* дисплеев на всех основных типах ПК. Такое ПО работает с ГО, не являющимися объектами некоторого координатного пространства, а представляющими собой отдельные пиксели или **их** совокупности в некоторой **пиксельной матрице** - стандартном листе для рисования. В этом случае выводимое на экран изображение представляет собой 2-мерный массив **пикселов** - элементов графического изображения, несущих информацию о **яркости** и **цвете** элементарного участка изображения, совокупность которых составляет изображение в целом. Для представления одного **пикселя** требуется до двух байт информации. В настоящее время **пиксельные матрицы** характеризуются размерностью 1024x768 пикселов, определяя достаточно высокую разрешающую способность дисплейных систем для ПК.

Сформированное в пиксельной матрице изображение хранится в **видеопамяти** дисплея и выводится на экран в режиме *регенерации*. *Рисование* в цвете, реализуемое посредством **манипуляций** пикселями такой матрицы, в определенной мере подобно приготовлению фотографии путем модификации областей эмульсионного слоя пленки в отличие от обычного экспонирования ее посредством фотоаппарата, наведенного на некоторый ГО. Тогда как методика работы с ГО в среде *векторных* систем напоминает именно процесс обычного фотографирования. Таким образом, в отличие от *векторного* **растровая** технология позволяет создавать рисующие ПС, работающие независимо от аппаратных средств на нижних уровнях обработки графической информации.

В настоящее время разработаны и поставляются на рынок весьма разнообразные ПС, ориентированные на различные приложения, многие из которых упоминались выше.

## **2.2 Основные методы формирования графических объектов**

Каждый разрабатываемый **графический** пакет ищет свою нишу в •среде исследователей, инженеров, дизайнеров, художников, чертежиков, рисующих детей и т.д. Существующие пакеты используют следующие **основные** (с точки зрения пользователя) технологические методы создания *графического* изображения-объекта (ГО):

- **Команда-параметры** - этим методом выбора команд и задания им соответствующих параметров пользуются известный пакет *AutoCAD* и созданные на его основе многие другие системы автоматизированного проектирования. Метод занимает свою нишу в основном в среде разработчиков средств технического черчения;

- **Команда-инструмент** - этим методом выбора команд и/или рисовальных инструментов пользуются пакеты типа *Paintbrush*;

метод нашел свою нишу в среде рисующих детей и дизайнеров, редактирующих *битовые* ГО, полученные сканированием или вводом с магнитных носителей. Данный метод использует также известный графический пакет *Adobe Photoshop*, имеющий расширенные возможности выделения областей и цветового редактирования;

- **Команда-символ-инструмент** - этим методом выбора команд, инструментов и готовых графических изображений (*символов*) пользуются многочисленные пакеты типа *CorelDRAW*, нашедшие применение, главным образом, в *иллюстративных* целях (рекламные заставки, фирменные знаки, оформительские материалы и т.д.), а также в качестве относительно простых графических редакторов.

Разделение технологических методов на такие группы в значительной мере условно и лишь отражает основную тенденцию, т.к. любой пакет использует в большей или меньшей мере задание параметров и инструментов в своем графическом интерфейсе. Очевидно, ниша *художественного* рисования на экране остается почти незанятой, ибо получение *кривых* линий вытягиванием или построением по касательным в значительной мере непривычно для традиционной работы художника, а физическим рисованием посредством инструментов трудно создать на экране сложный фрагмент или узор.

Существует ряд численных методов, использующих специальные алгоритмы, генератор псевдослучайных чисел и итерации, с помощью которых можно программным путем, *задавая параметры*, создавать фрагменты достаточно сложных ГО для их последующего конструирования и редактирования.

Приемы работы с графическими редакторами рассматриваются в специальных курсах, поэтому на этом обзор графических ПС закончим.

*Лекция 3 (2 часа)*  
*Табличная обработка информации (электронные таблицы)*  
*Элементы работы в среде табличного пакета MS Excel*

### **3. Табличная обработка информации (электронные таблицы)**

С появлением класса ПК широкое распространение получила группа пакетов обработки *электронных таблиц*, позволяющих решать широкий круг научно-технических, планово-экономических, учетно-статистических и других задач, для которых *исходные данные и результаты* обработки могут быть представлены в *табличной* форме или в самой сути которых лежит *табличное* представление информации.

Под *электронными таблицами* будем понимать ПС обработки крупно форматных *электронных динамических* таблиц. Средства данного типа обрабатывают *таблицы*, состоящие из строк и столбцов, на пересечении которых располагаются клетки. Каждая клетка такой таблицы может содержать числовое значение, формулу или текст. Клетки идентифицируются именами, состоящими из номера строки и номера столбца, например A55, G50, S3 и т.д. Значение в числовой клетке таблицы может быть записано или вычислено по ассоциируемой с данной клеткой формуле; в формуле могут присутствовать ссылки на другие клетки таблицы. Каждый раз при изменении значения в клетке таблицы в результате записи в нее нового значения с консоли ЭВМ пересчитываются и значения всех тех клеток, которые содержат величины, зависящие от изменяемой клетки. Возможность запоминания текста в привязке к клеткам таблицы используется для присваивания заголовков столбцам, названий строкам и т.д. Электронные таблицы, в частности, весьма удобны для реализации на ПК, поддерживающих режим быстрого и гибкого манипулирования цветными изображениями на экране монитора.

Общей характерной чертой всех электронных таблиц является использование экрана в качестве окна обзора таблицы целиком либо по выбранным ее участкам; если число строк и столбцов слишком велико, то окно обзора можно перемещать по таблице по горизонтали и вертикали, обозревая ее невидимые части. Для изменения какого-либо значения достаточно поместить курсор в соответствующую клетку таблицы, визуализируемую на экране, и записать с консоли в нее основное значение. При этом обеспечиваются хранение в памяти ПК и просмотр на экране монитора таблиц большой размерности; размещение крупных работ (макросов) содержащих ссылки на другие клетки документа и встроенные функции; отображение на экране значений, вычисляемых по формулам, хранящимся в клетках документа; автоматическая корректировка результатов документа при изменениях содержимого клеток, на которые в формулах имеются ссылки; графическое отображение информации и т.д. Электронные

таблицы оказываются весьма эффективным способом обработки различного рода информации, которую можно представлять в табличной форме (научная, деловая, коммерческая и т.д.). Среди средств данного типа наиболее известными и популярными являются пакеты Lotus 1-2-3, Quattro Pro, SuperCalc и Ms Excel, кратко рассматриваемый ниже.

### **3.1 Элементы работы в среде табличного пакета Ms Excel**

1. *Общие сведения.* Обработка *табличных* документов - наиболее массовый вид прикладной деятельности на ЭВМ, в первую очередь на ПК, в различного рода организациях, офисах, учреждениях и т.д. И среди средств, обеспечивающих данный вид деятельности, одну из лидирующих позиций по праву занимает широко известный пакет *MS Excel* фирмы *Microsoft*, версия 5.0 которого весьма распространена па IBM-совместимых ПК и Macintosh. Пакет поставляется как в виде самостоятельного средства, так и в составе *интегрированного* средства *Microsoft Office*, имеющего широкое распространение среди организаций учрежденческого типа. В настоящее время на рынок поставляется версия 7 пакета, работающая под управлением *Windows*, однако мы будем вести изложение именно относительно версии 5.0, которая представляет собой хорошо апробированное ПС и обеспечивает потребности весьма широкого круга пользователей, включая решение задач подготовки высококачественных материалов к печати. Более того, переход на версию 7 из среды рассматриваемой версии не представляет затруднений, но, на наш взгляд, возможности версии 5.0 пакета окажутся избыточными для массового пользователя.

Пакет *MS Excel 2000* (в дальнейшем просто *Excel* или *пакет*) представляет собой относительно более ранних средств данного типа достаточно большую программную систему (так, только *SXE-файл* пакета имеет объем порядка 4 Мбайт), требующую значительных ресурсов ПК. В результате совокупного функционирования указанной программной системы у пользователя весьма актуальными могут стать вопросы оптимизации работы, а то и возможности выполнения ее вообще. Так как *Excel* является *третьим* уровнем упомянутой выше цепочки ПС на пути к конечному пользователю, то оптимизируя каждую из ее составляющих, можно оптимизировать и всю работу в целом.

1. *Главные окно и меню пакета Excel.* После активации пиктограммы *Excel* двойным щелчком мыши производится загрузка *резидентной* части пакета с выходом на его *главное окно*, общий вид которого представлен на **рисунке 3.1.**

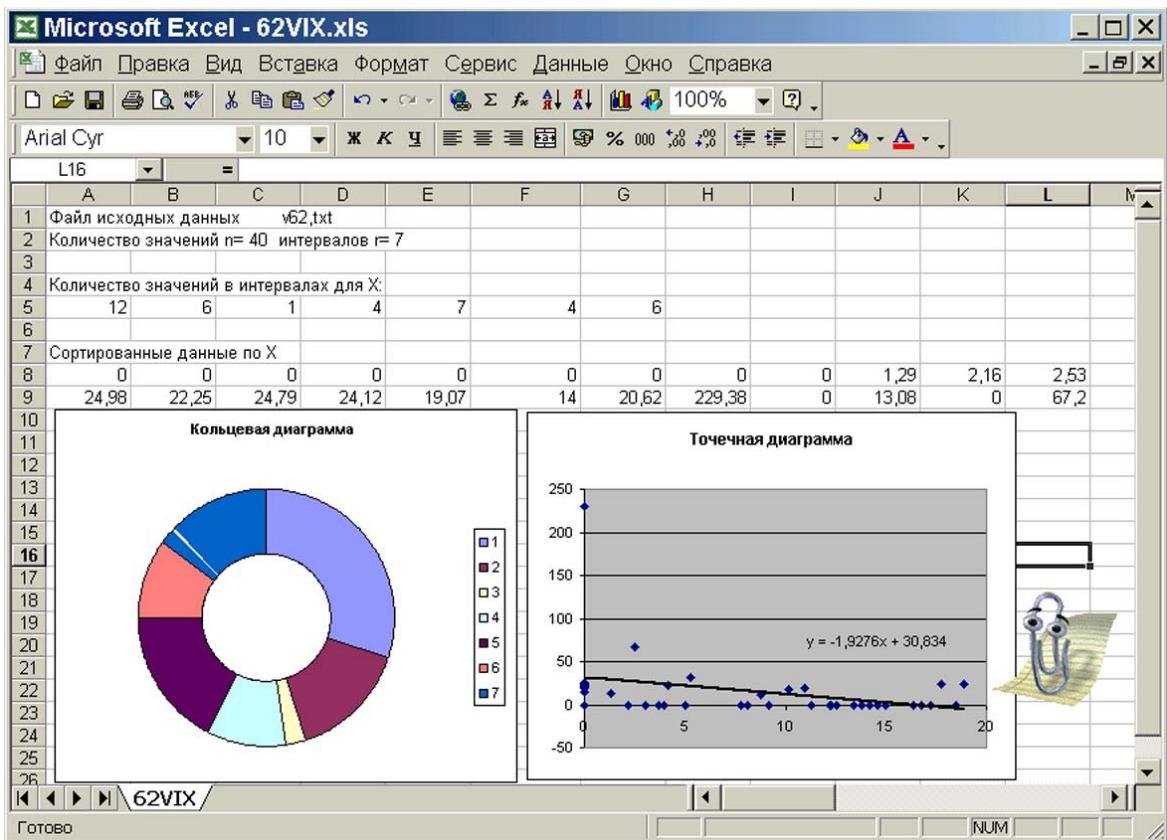


Рисунок 3.1. – Главное окно пакета Microsoft Excel

Главное окно пакета в минимальном составе содержит четыре основных поля:

- первая строка содержит имя пакета и имя активного в данный момент документа, а также кнопки для управления режимом визуализации окна (увеличения, уменьшения, сведения к пиктограмме), завершения работы с пакетом и др.;

- вторая строка содержит главное меню пакета (ГМП) и управляющие кнопки, аналогичные упомянутым; детальное их описание можно найти в литературе по пакету *Ms Excel* или оболочке *Windows*;

- третье поле представляет собой собственно рабочее окно текущего документа, через которое пользователь не только имеет возможность просматривать весь документ, но и производить сам процесс создания/редактирования его; рабочее окно обрамлено координатной разметкой клеточной сетки таблицы, позволяя легко адресовать конкретные клетки таблицы;

- наконец, четвертая управляющая строка содержит рычажки для перемещения окна просмотра текущей страницы влево/вправо и подключения на другие страницы текущего документа.

Дополнительно к указанным рабочее окно документа обрамляется рамками, содержащими указатели, позволяющие посредством активации их

мышию перемещать окно по текущему документу (рисунок 3.1). Наряду с *обязательными* по выбору пользователя могут визуализироваться *специальные компоненты*, строки которых могут располагаться в верхней/нижней части *главного окна пакета*. Для повышения уровня интерфейса со средствами пакета в состав доступных в оперативном режиме компонент главного окна рекомендуется включать *формульную строку* (ФС), *строку-меню* (для ряда приложений пакета таких строк может быть несколько) и *информационную строку* (ИС). На рисунке 3.1 представлено *главное окно пакета*, содержащее как *основные* его поля, так и указанные три *дополнительные* строки, соответственно 4-я (ФС), 3-я (строка-меню) и *последняя* (ИС) строки главного окна пакета. Для большинства пользователей такая организация *главного окна пакета* вполне достаточна, обеспечивая приемлемый размер *рабочего окна документа*. Кратко рассмотрим их состав и назначение.

*Строка-меню* (одна или несколько) обеспечивает *оперативный* доступ к средствам пакета посредством находящихся в ней кнопок-пиктограмм соответствующих средств и функций. Их состав и вопросы комплектования из стандартно поддерживаемых пакетом пиктограмм нужных пользователю строк-меню аналогичны рассмотренному выше случаю.

*Информационная* строка является последней строкой главного окна пакета; в случае готовности пакета к вводу информации она отображает *Ready-* состояние, в случае редактирования текущего документа - *Edit-состояние*. Наряду с этим строка динамически отображает *текущий* режим работы с пакетом, а также различного рода сообщения (информационные, диагностические, подсказки, установочные и т.д.).

*Формульная строка* (4-я на рисунке 3.1) состоит из *пяти* полей (*слева направо*), имеющих соответственно следующую смысловую нагрузку:

- окно индикации координат активной в данный момент клетки таблицы;

-  - кнопка отмены последней операции ввода (эквивалента *Esc*-клавише);

-  - кнопка подтверждения ввода данных (эквивалента *Enter*-клавише);

-  - кнопка активации выбора необходимой для обработки таблицы функции;

- рабочее поле, через которое сканируется/редактируется содержимое клетки.

При активации кнопки ФС открывается специальное окно, обеспечивая доступ к встроенным функциям пакета, сгруппированным по их назначению и позволяющим производить вычисления, используя в качестве значений своих *аргументов* конкретные данные и/или имена клеток таблицы. Все

последующее изложение будет соотноситься с рисунке 3.1, отражающим конкретное состояние *главного окна пакета* с рассматриваемым в дальнейшем простым примером, *иллюстрирующим* наиболее часто используемые средства *Excel-пакета* для обработки различного рода документов *табличного* типа. Однако прежде всего кратко рассмотрим основные возможности пакета, предоставляемые пользователю посредством функций ГМП.

*Главное меню пакета* (ГМП) содержит 9 групп функций, предоставляющих весьма широкие возможности для создания, редактирования и вычислений в документах *табличного* вида. Организация ГМП пакета во многом аналогична ГМП рассмотренного ранее *Word-пакета* - то же количество *функциональных* групп, идентичные названия для них (и функциональное наполнение каждой из них организовано подобным же образом с учетом двух аспектов: необходимости обработки *табличной* информации, включая расширенные *вычислительные* операции над *клетками* табличных объектов. По сути дела, в отличие от *текстового* процессора *Word* (наделенного простыми вычислительными операциями в рамках создаваемых в его среде *табличных* объектов; формула-функция *Table-* группы) пакет *Excel* является процессором обработки *табличной информации с развитыми* средствами редактирования и создания в его среде *текстовых и графических* объектов.

Для выбора требуемой *функции* пакета активируется *одинарным* щелчком мыши соответствующая группа ГМП, раскрывая *подокно*, содержащее список функций (условно сгруппированных по *секциям* согласно основному их назначению) с приписанными им клавишами *быстрой* активации (минута ГМП). Ниже будем предполагать, что мы располагаем некоторым готовым *документом* (текущим, находящимся в рабочей области пакета и/или дисковом файле), и все обсуждаемые пакетные средства будем пока рассматривать относительно этого документа, не акцентируя особого внимания на вопросах его создания. Такой документ представляет собой одну или несколько *квадратных таблиц*, строки которой нумеруются цифрами {1-16 384}, а столбцы - латинскими буквами и их парными комбинациями {A-Z, AA-AZ, BA-BZ, CA-CZ, ..., IA--IV}. Каждая клетка такой таблицы нумеруется (именуется) номером столбца и номером строки, на пересечении которых она находится, например: G51, S31, A8, V55, K1 и т.д. *Клетки* могут содержать текстовую, числовую, графическую информацию, а также формулы и даже целые таблицы. Пакет позволяет производить над клетками таблицы операции подобно тому, как это делают языки программирования над *переменными*, т.е. в качестве переменной выступает *клетка* таблицы.

Прежде всего, средства меню «*Справка*» предоставляют не только справочную информацию по всем возможностям пакета, но и предлагают

демонстрационные примеры, позволяющие лучше уяснить основные возможности пакета и принципы их использования.

Средства меню «Справка» разбиты на четыре секции (рисунок 3.2): (1) получение справочной информации по пакету; принцип доступа к такой информации полностью соответствует соглашениям *Windows*; (2) краткие обзор по пакету и курс лекций с демонстрацией его основных возможностей; (3) справочная информация по соответствиям между командами пакетов *Excel*, *Multiplan* и *Lotus 1-2-3*, а также (4) краткая информация по сопровождению *Excel*; справка по пакету и лицензионная информация; по опции *System Info* можно дополнительно получать информацию по текущему состоянию основных ресурсов ПК, в первую очередь ОП. Из средств данной группы наиболее часто используемыми являются функции *первой* ее секции.

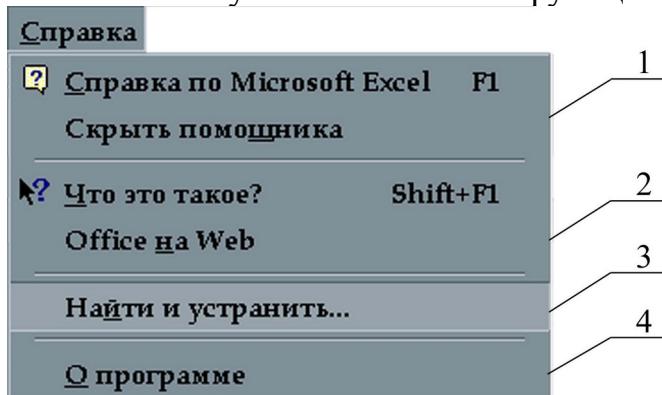


Рисунок 3.2 – Меню «Справка»

Группа Window содержит три секции, функции которых позволяют: (1) управлять окнами активных в среде пакета документов (открывать новые, реорганизовывать, скрывать, визуализировать); (2) производить *реорганизацию* окон пакета, включая разделение окон по *выделенной* клетке таблицы; и (3) обеспечивать доступ к *списку документов*, находящихся в рабочей области пакета и к которым можно легко обращаться, активируя их окна через соответствующие элементы данного списка - *активный* в текущий момент документ в списке помечается маркером. Средства данной группы весьма эффективны в случае необходимости *совместного* редактирования нескольких различных документов или различных частей одного достаточно большого документа, позволяя просматривать разные его области. Принцип совместной обработки документов во многом подобен случаю, рассмотренному для текстового *Word*-пакета.

Средства меню «Файл» ГМП разбиты на 7 секций (рисунок 3.3) по своему функциональному назначению и в целом обеспечивают работу с файлами, содержащими *документы* как основные *объекты* обработки в среде пакета.

По умолчанию документы *Word-формата*, имеют *xl\*-расширение имени* и, если не оговорено противного, пакет *автоматически* начинает отыскивать именно такие файлы, при загрузке проверяя их формат на корректность.

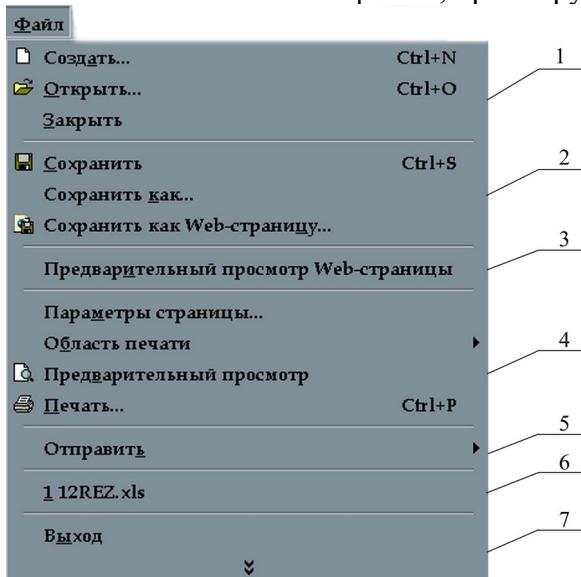


Рисунок 3.3 – Меню «Файл»

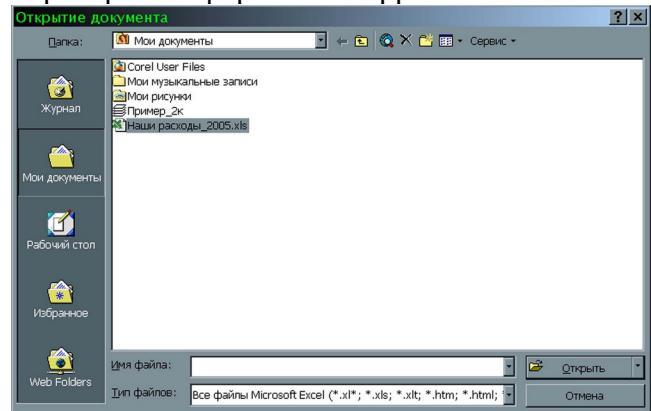


Рисунок 3.4 – Открытие существующего документа

*Первая секция (1)* (смотри рисунок 3.3) содержит функции: *создания нового* (*Создать*, *Ctrl+N*), *открытие* существующего (*Открыть*; *Ctrl+O*) и *закрытие* текущего (*Закрыть*) документа.

При открытии существующего файла (рисунок 3.4) открывается окно «*Открытие...*», поля и переключатели которого позволяют выбирать нужный файл из предлагаемого списка, обеспечивать поиск нужного файла, определять его формат по расширению имени и т.д. В частности, поддерживается использование текстовых файлов, а также файлов, созданных в среде пакетов *dBase*, *Lotus 1-2-3*, *Quattro Pro*, *Ms Works* и др. Структура окна соответствует соглашениям оболочки *Windows* и аналогична другим *Windows*-приложениям.

*Вторая секция (2)* содержит функции сохранения: текущего документа с обновлением исходного файла (*Сохранить*; *Ctrl+S*), текущего документа, в новом файле и/или в новом формате (*Сохранить как*) и текущего документа как Web-страницу (*Сохранить как Web-страницу*). По функции *Сохранить как* предоставляется возможность не только сохранять текущий документ согласно новому идентификатору файла, но и в нужном формате, определяемом списком из ряда допустимых, включая форматы пакетов *Lotus 1-2-3*, *Quattro Pro*, *dBase*, систем *Macintosh*, *MS-DOS*, *OS/2* и др. Данная возможность полезна при последующем использовании такого файла в среде других ПС.

Средства *третьей* секции группы (3) обеспечивают предварительный просмотр Web-страницы.

Средства четвертой секции (4) связаны с обеспечением вывода текущего документа на печать, параметрами страницы.

По функции «Предварительный просмотр» (рисунок 3.5) предоставляется возможность постранично просматривать файл, отслеживая его форматирование относительно заданных выходных параметров: размеров страницы и полей, Расположения нумерации, сносок, типа принтера и др. Вообще говоря, функция поддерживает режим *WYSIWYG* - каким видится, таким и выводится.

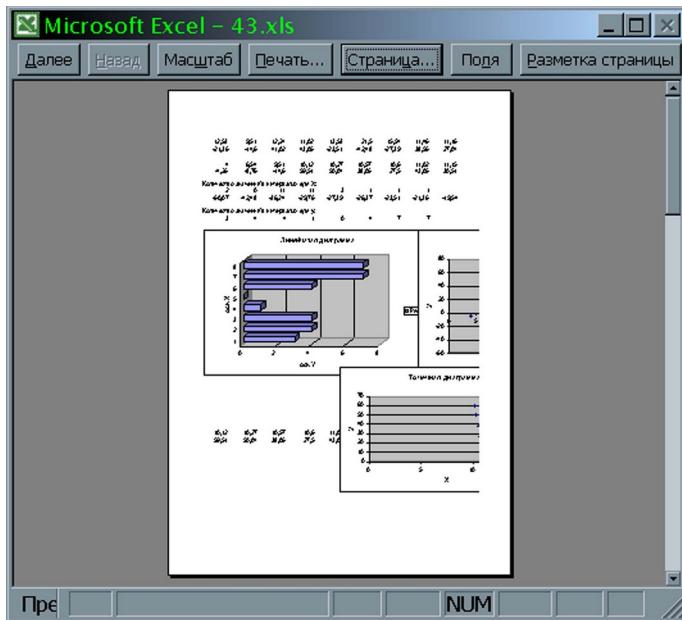


Рисунок 3.5 – Просмотр документа MS Excel

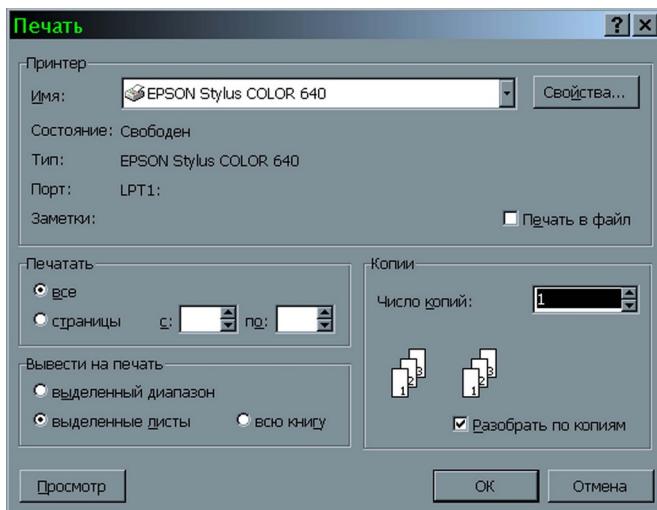


Рисунок 3.6 – Меню «Печать»

По функции «Параметры страницы» открывается окно, поля которого позволяют определять размеры выходной страницы текущего документа при выводе на печать, поля и режим расположения документа на странице, нумерацию страниц и т.д. По функции «Печать» (*Ctrl+P*) открывается окно печати, позволяющее определять режим непосредственного вывода документа на печать (весь документ или отдельные его страницы-таблицы, тип принтера, количество копий и т.д.) (рисунок 3.6).

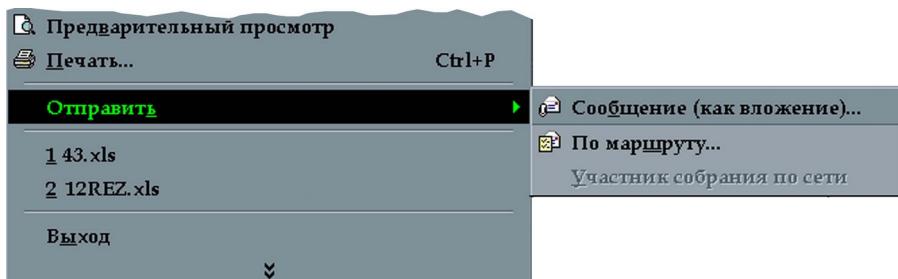


Рисунок 3.7 – Меню «Файл», функция «Отправить»

Средства пятой секции (5) (рисунок 3.7) меню «Файл» обеспечивают отправку созданного документа по электронной почте и т.д.

В шестой секции (6) (смотри рисунок 3.3) находятся имена последних открытых или созданных файлов.

Последняя седьмая секция (7) обеспечивает выход из среды пакета в Windows-среду; при этом запрашивается санкция на сохранение текущего документа (*Да*), без сохранения (*Нет*) и отмену выхода (*Отмена*) из пакета.

Перед обсуждением средств следующей группы ГМП поясним одно из **ключевых** понятий редактирования текущего документа - *выделенный блок*. Прежде всего, в качестве *выделенного блока* может быть любая компонента документа: отдельная клетка таблицы, прямоугольная совокупность клеток, таблица, ГО и т.д. Выделение блока клеток или отдельной клетки производится фиксацией мышью его *начала* с последующим перемещением ее *курсора* (визуализируемого в области активной таблицы документа) вперед/назад, в результате чего выбираемый блок *выделяется*. ГО и другие подобные объекты выбираются установкой на них мыши с последующим щелчком, в результате чего визуализируется обрамляющая их прямоугольная рамка. Прекращено фиксации мыши устанавливает конец *выделенного блока*. С *выделенным блоком* работает целый ряд функций как *редактирования*, так и *вычислительных*.

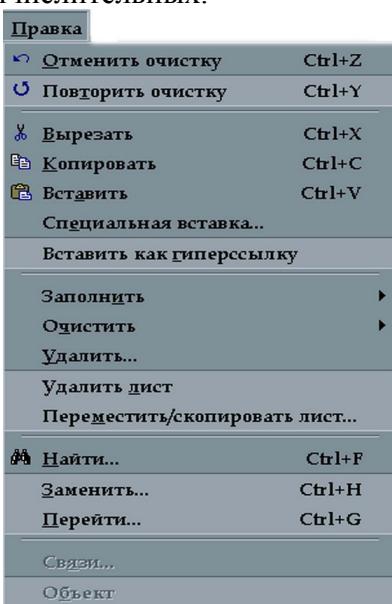


Рисунок 3.8 – Меню «Правка» Рисунок 3.9 – Функция «Очистить»



Средства меню «Правка» (рисунок 3.8) во многом дублируют соответствующие средства одноименной группы Word-пакета с очевидными изменениями (учитывая специфику табличной организации документов) и обеспечивают ряд важных функций редактирования текущего документа.

Так, опция «Очистить» (рисунок 3.9) позволяют удалять из выделенных клеток таблицы текущего документа любое содержимое (данные, текст, формулы и т.д.) и приписанные им форматы.

Функции поиска, перехода и замены (рисунок 3.10) ассоциируются с содержимым клеток таблицы. Функция «Найти» позволяет в документе найти любой интересующий элемент. Функция «Заменить» позволяет заменить один элемент документа на любой другой.

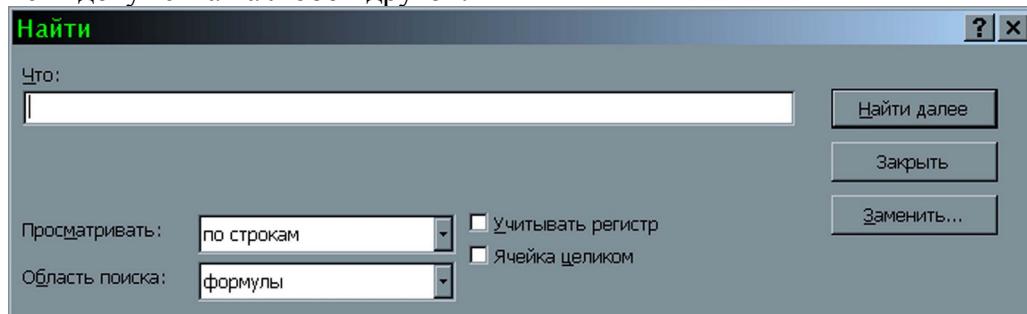


Рисунок 3.10 – Меню «Найти»

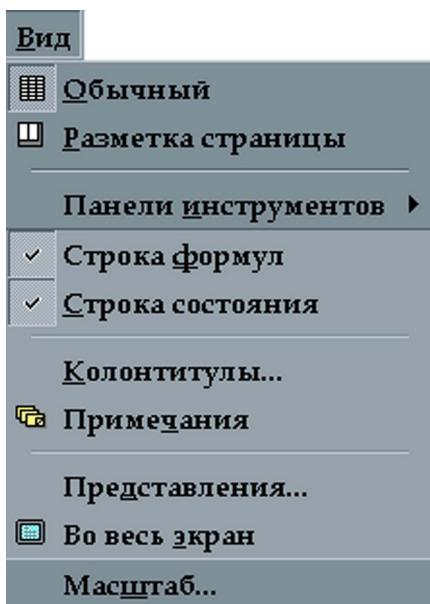


Рисунок 3.11 – Меню «Вид»

Средства меню «Вид» (рисунок 3.11) предназначены для выбора режима просмотра текущего документа и определения визуализируемого набора средств пакета, позволяя: (1) визуализировать/скрывать формульную строку (Строка формул), информационную строку (Строка состояния), реорганизовывать набор строк-меню пакета и определять режим их визуализации на экране (Панели инструментов); (2) отводить под таблицу документа все окно (Обычный) экрана (оставляя лишь ГМП и, если определена видимой, информационную строку), а также (3) изменять масштаб изображения текущего документа (Масштаб).

Средства меню «Вставка» (рисунок 3.12) позволяют производить вставки необходимых *объектов* (другие документы, макросы, рисунки, данные из БД и др.), а также *линий раздела страниц* и др.

Группа включает функции, разбитые по их основному назначению на *четыре* секции, обеспечивающие следующие *основные* средства: (1) вставка в таблицу строк, столбцов или выбранных клеток (*Ячейки*, *Строки*, *Столбцы*); (2) добавление нового листа таблицы в текущий документ (*Лист*), помещение на текущей или новой странице документа *диаграммы* (*Диаграмма*), полученной на основе *данных* документа; (3) установка в документе *жесткой* линии раздела страниц (*Разрыв страницы*), активизация выбора встроенной функции пакета (*Функция*) для вставки ее или результата ее вычисления в клетку таблицы, а также обеспечение работы с именами (*Имя*); (4) вставка в клетку рисунка/объекта (*Рисунок/Объект*) из указанного файла и др. Большинство из указанных функций располагает *целым рядом опций*, позволяющих выполнять достаточно широкий спектр операций по обеспечению работы с помещаемыми в документ объектами и конструкциями различного типа.

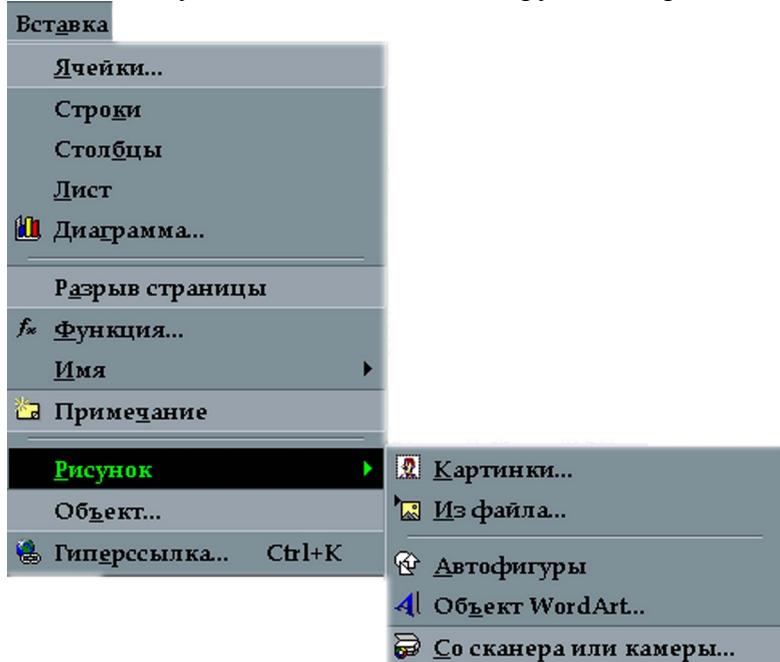


Рисунок 3.12 – Меню «Вставка»

Основным назначением функций меню «Формат» (рисунок 3.13) является *форматирование выделенных блоков* текущего документа.

По своему основному назначению функции группы образуют *две секции*, позволяющие проводить следующие основные процедуры: (1) изменять параметры строк и столбцов (высоту, ширину и т.д), переименовывать текущую страницу документа (*Лист*), а также изменять режим ее видимости; (2) производить автоматическое форматирование выбранного блока клеток таблицы согласно заданному формату из списка поддерживаемых пакетом (*Автоформат*) или обеспеченных пользователем,

изменять стили (форматы чисел, шрифты, центровка и др.) выделенной клетки (*Стиль*).

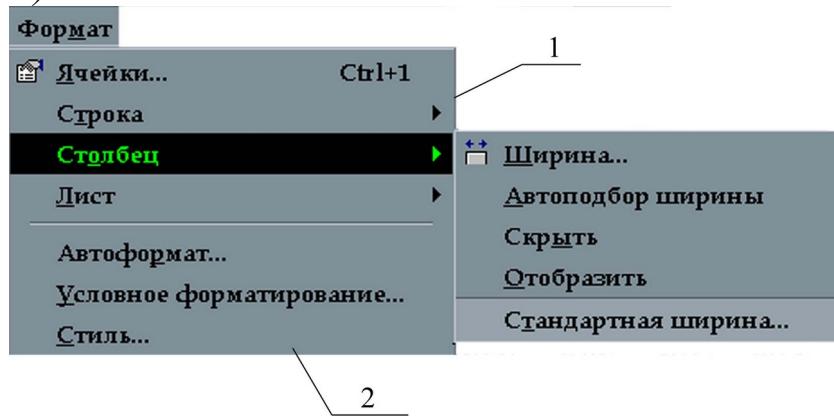


Рисунок 3.13 – Меню «Формат»

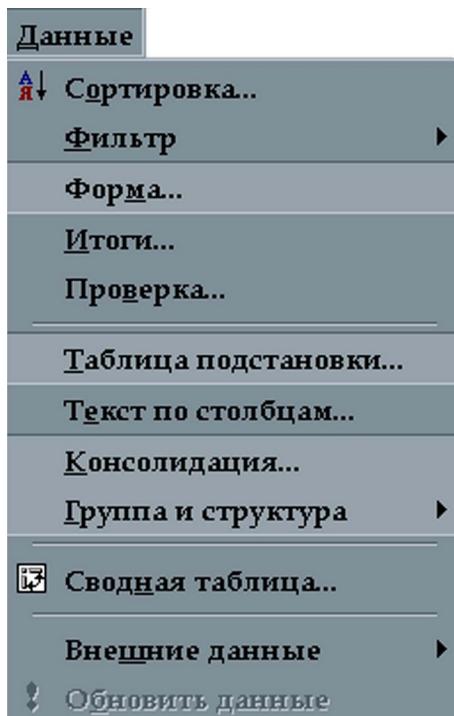


Рисунок 3.14 – Меню «Данные»

Средства меню «Данные» (рисунок 3.14) обеспечивают работу с данными, помещаемыми в клетки таблицы, позволяя: сортировать выделенные клетки согласно заданному критерию (*Сортировка*); обеспечивать фильтрацию данных таблицы согласно заданным *тестирующим* условиям (*Фильтр*); обеспечивать работу с записями базы данных (*Форма*); создавать в выбранной клетке таблицу (*Сводная таблица*); реорганизовывать текст в выделенных клетках в столбцы (*Текст по столбцам*); определять режим визуализации деталей выделенных клеток, группировать/разгруппировать их и т.д. (*Группа и структура*) и целый ряд других полезных функций обработки таблиц.

Наконец, средства меню «Сервис» (рисунок 3.15) предоставляют дополнительные возможности по работе с текущим документом, переопределению установок ряда основных параметров и характеристик пакета, а также по автоматизации операций пользователя. Функции данной группы по их основному назначению можно классифицировать по *пяти* секциям: (1) орфографическая проверка текстовой информации документа

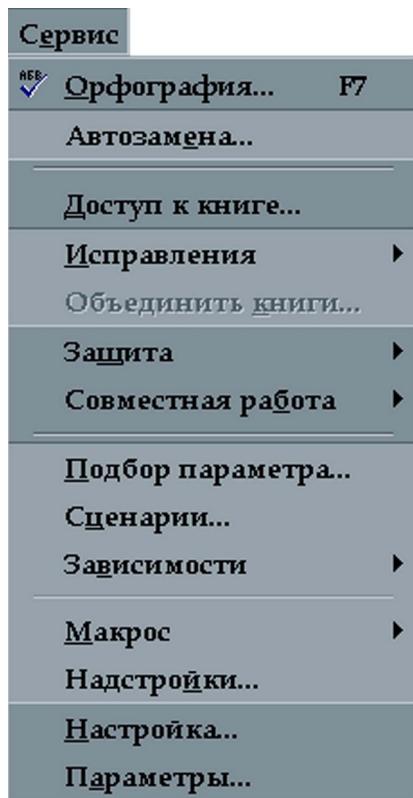


Рисунок 3.15 – Меню «Сервис»

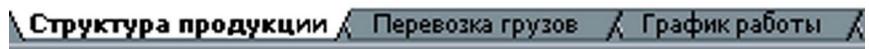
(*Орфография*) и детальная проверка методов получения результатов в клетках таблицы (*Автозамена*) с визуализацией использованных операций, функций, формул и т.д.; (2) определение доступа к книге; (3) обеспечение парольной защиты содержимого, объектов или схемы вычислений для отдельной таблицы или всего документа в целом (*Защита*); (4) обеспечение работы с мощным аппаратом макросов (*Макрос*) и (5) переопределение основных опций параметров и характеристик пакета (*Настройка*, *Параметры*) опции главного окна пакета, режимов вычислений и редактирования, перехода от форматов *Excel* к форматам *Lotus 1-2-3*, форматов диаграмм, палитры, общие опции и др.) и ряд других важных функций.

Оперативный доступ к большинству *основных* и *дополнительных* средств пакета обеспечивает *графический* интерфейс на основе соответствующих им *кнопок-пиктограмм*, сгруппированных в специальные *строки-меню*, кратко рассмотренные выше. В стандартную поставку входит 13 таких строк-меню с достаточно удачно подобранным наполнением их кнопками-пиктограммами (*средствами*).

### 3.2. Элементы работы в среде пакета Excel

Пакет представляет собой достаточно мощное средство, обеспечивающее разнообразную обработку документов табличного типа. Однако мы остановимся лишь на самых общих его возможностях и принципах применения. Файлы, с которыми работает пакет, имеют *xl\**-расширения, а собственно сами *Excel*-документы располагаются в *XLS*-файлах, если не определено противного. Каждый *Excel*-документ состоит из необходимого числа страниц, каждая из которых содержит таблицу, диаграмму, модули *Visual Basic* или другие объекты. Переключение между страницами текущего документа производится посредством соответствующих кнопок управляющей строки окна пакета (рисунок 3.16). В качестве названий кнопок используются *пакетные* идентификаторы (*n* -

номер страницы) или *имена*, присвоенные пользователем в процессе создания документа.



*Рисунок 3.16 – Кнопки управления окнами пакета MS Excel*

Для *выделения (выборки)* нужной клетки таблицы текущего документа на нее устанавливается курсор и производится однократный щелчок мышью (пакет допускает в качестве управляющего устройства и традиционную клавиатуру, но данный вопрос здесь не рассматривается по причине практического отсутствия современных ПК без очень удобного и эффективного устройства управления курсором типа мыши).

*Выбранная* клетка выделяется *обрамляющей* ее рамкой и в нее можно вводить необходимую информацию. В момент ввода активизируются поля ФС, отображая в первом поле имя (координаты) выбранной (активной) клетки, а в последнем поле - само содержимое вводимой (или уже находящейся в клетке) информации. Через -кнопку ФС (или по *ESC*-клавише) производится *отмена* ввода информации, а по -кнопке (или по клавише *Enter*) подтверждается ввод набранной информации в активную клетку. Содержимое выделенной (*активной*) клетки ; можно *редактировать* либо через ФС, либо непосредственно в самой , клетке, что во многих случаях значительно удобнее. Во *втором* случае на нужную клетку устанавливается курсор и производится двойной щелчок мышью. В результате этого поля ФС получают отмеченное выше содержимое (для *активированной* клетки), в *информационной* строке идентифицируется *Edit*-состояние, а в самой клетке визуализируется текстовый курсор, допуская редактирование ее содержимого аналогично способу, рассмотренному выше для случая *Word*-пакета.

Завершение редактирования определяется активацией -кнопки в ФС или по клавише *Enter* с возвратом пакета в основное состояние пакета; при этом в ФС скрываются управляющие (, , ) кнопки.

Независимо от типа решаемой в среде пакета задачи работа с пакетом в общем случае сводится к следующей общей технологии. Прежде всего, *выбирается* описанным способом *клетка* (или группа клеток) таблицы и активируется нужная функциональная группа ГМП или пиктограмма строки меню, обеспечивающая соответствующую обработку клетки. Для выбора прямоугольного блока клеток можно протянуть по ним курсор, зафиксировав начало в *левой верхней* клетке блока и прекратив фиксацию в его *нижней правой* клетке. Блок в пакете идентифицируется этими *концевыми* клетками. После выбора *клетки* или *блока* клеток активизируется, Пользователь получает доступ к *средствам* пакета, обеспечивающим соответствующую обработку выбранных до этого клеток таблицы текущего документа.

Например, скопировать по функции копирования содержимое выбранного блока клеток в буфер обмена (выбранный блок после выполнения функции окружается *циркулирующей рамкой*), затем выбрать нужную клетку таблицы текущей или другой страницы документа (используя переход к ней по переключателям управляющей строки) и по функции «*Вставка*» произвести копирование содержимого СБО в выбранную клетку, как *начальную* клетку скопированного блока клеток.

Наряду с описанным пакет позволяет и непосредственно производить копирование содержимого *выбранных* клеток таблицы путем *перетаскивания* выделенного блока в нужное место текущей таблицы, зафиксировав его рамку указателем, как это делается в большинстве приложений. Оперативный доступ к средствам пакета обеспечивают строки-меню, содержащие соответствующие наборы пиктограмм, о которых говорилось выше в контексте как *Excel*, так и *Word*-пакетов.

Для изменения *стиля* измерения валютных единиц выбираются клетки с необходимыми числовыми данными и активируется в строке-меню *\$-пиктограмма*, в результате чего всем значениям выбранных клеток будет предшествовать знак *доллара*; для округления денежных величин используются соответствующие пиктограммы строки-меню и т.д. В результате решается весьма простая задача: в таблице размещаются определенного типа данные и производится подведение итогов по столбцам с определенным форматированием результатов - типичная *бухгалтерская* или *учетная* задача. Наиболее массовые задачи, решаемые в среде *Excel*-пакета, сводятся к следующим *основным* операциям: производится *выборка* клеток текущей таблицы с помещением в них текстовой и числовой информации, затем осуществляется выбор и применение необходимых функций, манипулирование с клетками таблицы, использование оперативных средств пакета через строки-меню и другие более специальные процедуры, включая создание диаграмм, помещение в документ объектов из других приложений и т.д. Созданный документ сохраняется для последующего использования и/или редактирования в дисковом файле, выводится на печать полностью или частично, а также может передаваться по *Email-now* или на *факс* удаленного абонента, используя телекоммуникационные средства *Wmdows*-среды или другого приложения.

## Лекция 4 (2 часа)

*Средства численных и символьных вычислений*

*Принципы работы с пакетами математических вычислений*

*Назначение MathCAD*

*Знакомство с MathCAD*

*Интерфейс пользователя*

### **4. Средства численных и символьных вычислений**

В настоящее время ПС, ориентированные на решение **математических** задач (где под *математической* понимается задача, алгоритм которой может быть описан в терминах того или иного раздела математики), весьма обширны и условно могут быть дифференцированы на пять уровней: (1) *встроенные* средства различной степени развития той или иной системы программирования; (2) *специальные ЯВУ*; (3) *узкоспециальные*, (4) *специальные* и (5) *общие* пакеты. К первому уровню могут быть отнесены такие СП, как *Basic*, *C*, *PL/I*, *Pascal-XSi* и др., ко второму - *Fortran*, *ISETL* и др. Третий уровень может быть, представлен как библиотеками математических подпрограмм, так и узкоспециальными пакетами *MacMath*, *Phaser*, *VossPlot*, *Eureka* и др. К четвертому уровню можно отнести такие пакеты, как *S-Plus*, *XploRe*, *SAS*, *Dynamics*, *Macsyma* и др. Пятый уровень представляют три основных математических пакета *MathCAD*, *REDUCE* и *MATLAB*.

В указанной литературе приведены достаточно подробное описание этих популярных пакетов, особенности их эксплуатации и применения при решении различного типа задач математического характера.

Наконец, современное развитие компьютерных технологий, ориентированных на создание интегрированных пакетов **multimedia**-технологии привело к появлению нового уровня математических пакетов, из которых наиболее известными являются пакеты **Maple** фирмы Maple Software (Канада) и **Mathematica** фирмы Wolfram Research (США). Данные пакеты, превосходя по целому ряду важных показателей упомянутые средства 5-го уровня, вместе с тем наследуют ряд их стандартов как пионеров-эталонов ПС данного типа, что легко прослеживается при более детальном их рассмотрении.

Пакет **Maple** поддерживает как числовые, так и символьные вычисления, позволяя весьма эффективно решать задачи из многих разделов современной математики и математических задач в указанном нами понимании из других областей. На наш взгляд, пакет **Maple V** является лидером среди ПС данных типа и уровня. Другим интересным средством, поддерживающим все основные типы вычислений, является пакет **Mathematica**, разработанный большой группой математиков и программистов. Данный пакет позволяет достаточно эффективно

производить численные (матричные операции, интегрирование, преобразование Фурье, нахождение корней, минимаксные задачи, линейное программирование, различные математические функции и др.) и символьные (алгебраические преобразования, работа с полиномами, интегрирование, решение уравнений, матричные операции, работа со списками и др.) вычисления и другие виды работ рассмотренные ниже. Пакет является весьма популярным ПС для решения задач математического характера в различных областях современного естествознания, а также в учебных целях.

#### **4.1 Принципы работы с пакетами математических вычислений**

При работе с математическим пакетом пользователь формирует вычислительный алгоритм и поясняющее его описание создавая *документ*, который можно затем сохранять для последующего использования, выполнять полностью или его отдельные части (*секции*). Для названия его будем использовать термин "*Math-документ*".

Возможности пакета позволяют исследователю работать в его среде с проблемой подобно тому, как он это делает за своим рабочим столом. В дальнейшем полученный *документ* можно модифицировать как в вычислительной части, так и в части имеющейся в нем *текстовой, графической* и *иллюстративной* информации. При этом допускается включать в документ чертежи и рисунки, полученные в среде ряда известных *вычислительных* и/или CAD-пакетов. Готовый документ может быть выведен на экран дисплея, принтер и/или плоттер или сохранен в дисковом файле.

Главные характеристики пакета можно условно разбить на *группы*:

интерфейс с пользователем, вычислительные возможности, работа с текстово-графической информацией и *multimedia*-возможности. При этом в процессе общения с пакетом в интерактивном режиме пользователь получает следующие *основные* возможности:

- в любом доступном месте текущего документа помещать текст, математические конструкции, *графические объекты (TO)* и рисунки, диаграммы, схемы и т.д.;
- легко вводить ГО и достаточно, сложные математические конструкции, помещая их на свободные места экрана (*сканируемой части текущего документа*);
- произвольно редактировать текст, ГО, математические конструкции, а также текущий документ в целом;
- вводить по мере необходимости функции, управляющие выполнением *текущего документа (ТД)* и многими основными ресурсами ПК;
- выводить копию всего или части текущего документа на принтер, плоттер или в дисковый файл, а также передавать объекты через СБО в другие ПС;

- изменять *глобальные* или *локальные* форматы результатов вычислений и/или ГО текущего документа, а также основные характеристики пакета;
- запрашивать выполнение функций систем *MS-DOS* или *Windows*;
- импортировать объекты из других ПС в широком диапазоне их форматов;
- обеспечивать связь с целым рядом популярных ПС на уровне функциональных средств и/или данных (*MathLink-протокол*).

**Из «вычислительных» возможностей пакета можно выделить следующие:**

- проведение различных вычислений с высокой степенью точности;
- алгебраические и численные вычисления производных и интегралов;
- решение систем алгебраических, дифференциальных и разностных уравнений;
- поддерживает целый ряд функций матричных и векторных вычислений;
- поддерживает вычисления как в действительной, так и в комплексной области;
- располагает широким набором *встроенных* математических функций;
- использование функций, определяемых самим пользователем;
- достаточно развитые средства доступа к файлам данных на дисках,
- широкий набор *Мат-документов* и программ, поставляемых с пакетом для решения различных математических задач в среде пакета;
- встроенный язык программирования задач в среде самого пакета (*Math-язык*).

Наконец, пакет предоставляет пользователю широкие возможности по работе с *текстовой*, *графической* и *илюстративной* информацией. Это обеспечивается как средствами создания и редактирования данных типов информации, так и поддержкой пакетом широких наборов шрифтов, типов дисплеев, принтеров и плоттеров.

Как следует уже из перечисленного, пакеты предоставляет весьма широкие возможности любому пользователю для решения математических задач из различных прикладных областей.

## 4.2 Назначение MathCAD

MathCAD является математическим редактором, позволяющим проводить разнообразные научные и инженерные расчеты, начиная от элементарной арифметики и заканчивая сложными реализациями численных методов. Пользователи MathCAD - это студенты, ученые, инженеры, разнообразные технические специалисты. Благодаря простоте применения, наглядности математических действий, обширной библиотеке встроенных функций и численных методов, возможности символьных вычислений, а также превосходному аппарату представления результатов (графики самых разных типов, мощных средств подготовки печатных документов и Web-страниц), MathCAD стал наиболее популярным математическим приложением. MathCAD 2001, в отличие от большинства других современных математических приложений, построен в соответствии с принципом WYSIWYG ("What You See Is What You Get" - "что Вы видите, то и получите"). Поэтому он очень прост в использовании, в частности, из-за отсутствия необходимости сначала писать программу, реализующую те или иные математические расчеты, а потом запускать ее на исполнение. Вместо этого достаточно просто вводить математические выражения с помощью встроенного редактора формул, причем в виде, максимально приближенном к общепринятым, и тут же получать результат. Кроме того, можно изготовить на принтере печатную копию документа или создать страницу в Интернете именно в том виде, который этот документ имеет на экране компьютера при работе с MathCAD. Создатели MathCAD сделали все возможное чтобы пользователь, не обладающий специальными знаниями в программировании (а таких большинство среди ученых и инженеров), мог в полной мере приобщиться к достижениям современной вычислительной науки и компьютерных технологий. Для эффективной работы с редактором MathCAD достаточно базовых навыков пользователя. С другой стороны, профессиональные программисты (к которым относит себя и автор этих строк) могут извлечь из MathCAD намного больше, создавая различные программные решения, существенно расширяющие возможности, непосредственно заложенные в MathCAD.<sup>3</sup>

В соответствии с проблемами реальной жизни, математикам приходится решать одну или несколько из следующих задач: - ввод на компьютере разнообразных математических выражений (для дальнейших расчетов или создания документов, презентаций, Web-страниц); - проведение математических расчетов; - подготовка графиков с результатами расчетов; - ввод исходных данных и вывод результатов в текстовые файлы или файлы с базами данных в других форматах; - подготовка отчетов работы в виде печатных документов; - подготовка Web-страниц и публикация результатов в Интернете; - получение различной справочной информации из области математики.

Со всеми этими (а также некоторыми другими) задачами с успехом справляется MathCAD: - математические выражения и текст вводятся с помощью формульного редактора MathCAD, который по возможностям и простоте использования не уступает, к примеру, редактору формул, встроенному в Microsoft Word; - математические расчеты производятся немедленно, в соответствии с введенными формулами; - графики различных типов (по выбору пользователя) с богатыми возможностями форматирования вставляются непосредственно в документы; - возможен ввод и вывод данных в файлы различных форматов; - документы могут быть распечатаны непосредственно в MathCAD в том виде, который пользователь видит на экране компьютера, или сохранены в формате RTF для последующего редактирования в более мощных текстовых редакторах (например, Microsoft Word); - возможно сохранение документов в формате Web-страницы, причем создание файлов с рисунками происходит автоматически; - символьные вычисления позволяют мгновенно получить разнообразную справочную математическую информацию, а система помощи, Центр Ресурсов и встроенные электронные книги помогают быстро отыскать нужную справку или пример тех или иных расчетов.

Таким образом, следует хорошо представлять себе, что в состав MathCAD входят несколько интегрированных между собой компонентов - это мощный текстовый редактор для ввода и редактирования, как текста, так и формул, вычислительный процессор - для проведения расчетов согласно введенным формулам, и символьный процессор, являющийся, по сути, системой искусственного интеллекта. Сочетание этих компонентов создает удобную вычислительную среду для разнообразных математических расчетов и, одновременно, документирования результатов работы.

### 4.3 Знакомство с MathCAD

*После того как MathCAD 2001 установлен на компьютере и запущен на исполнение, появляется основное окно приложения, показанное на рисунке 4.1. Оно имеет ту же структуру, что и большинство приложений Windows. Сверху вниз располагаются заголовок окна, строка меню, панели инструментов (стандартная и форматирования) и рабочий лист, или рабочая область, документа (worksheet). Новый документ создается автоматически при запуске MathCAD. В самой нижней части окна находится строка состояния. Не забывая о сходстве редактора MathCAD с обычными текстовыми редакторами, вы интуитивно поймете назначение большинства кнопок на панелях инструментов. При запуске на переднем плане также появляется диалоговое окно Tip of the Day (Совет Дня), которое можно убрать, нажав кнопку Close (Закрыть), на нее указывает курсор на рисунке При запуске MathCAD также можно наблюдать еще одно окно -*

*Resource Center (Центр Ресурсов), которое является, по сути, отдельной программой путеводителем по возможностям MathCAD 2001, снабженной множеством примеров решения самых различных математических, физических и инженерных задач. В своей работе вы можете либо не обращать на него внимания, либо пользоваться как хорошим дополнением к справочной системе.*

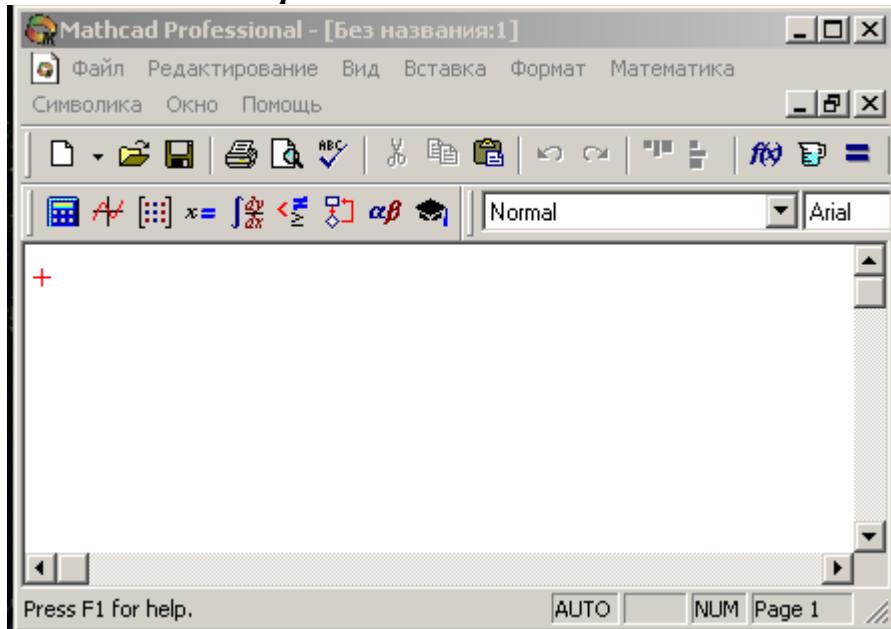


Рисунок 4.1 – Основное окно MathCad

- новый.

- открыть

- сохранить

- печать

- просмотр перед печатью

- проверка орфографии

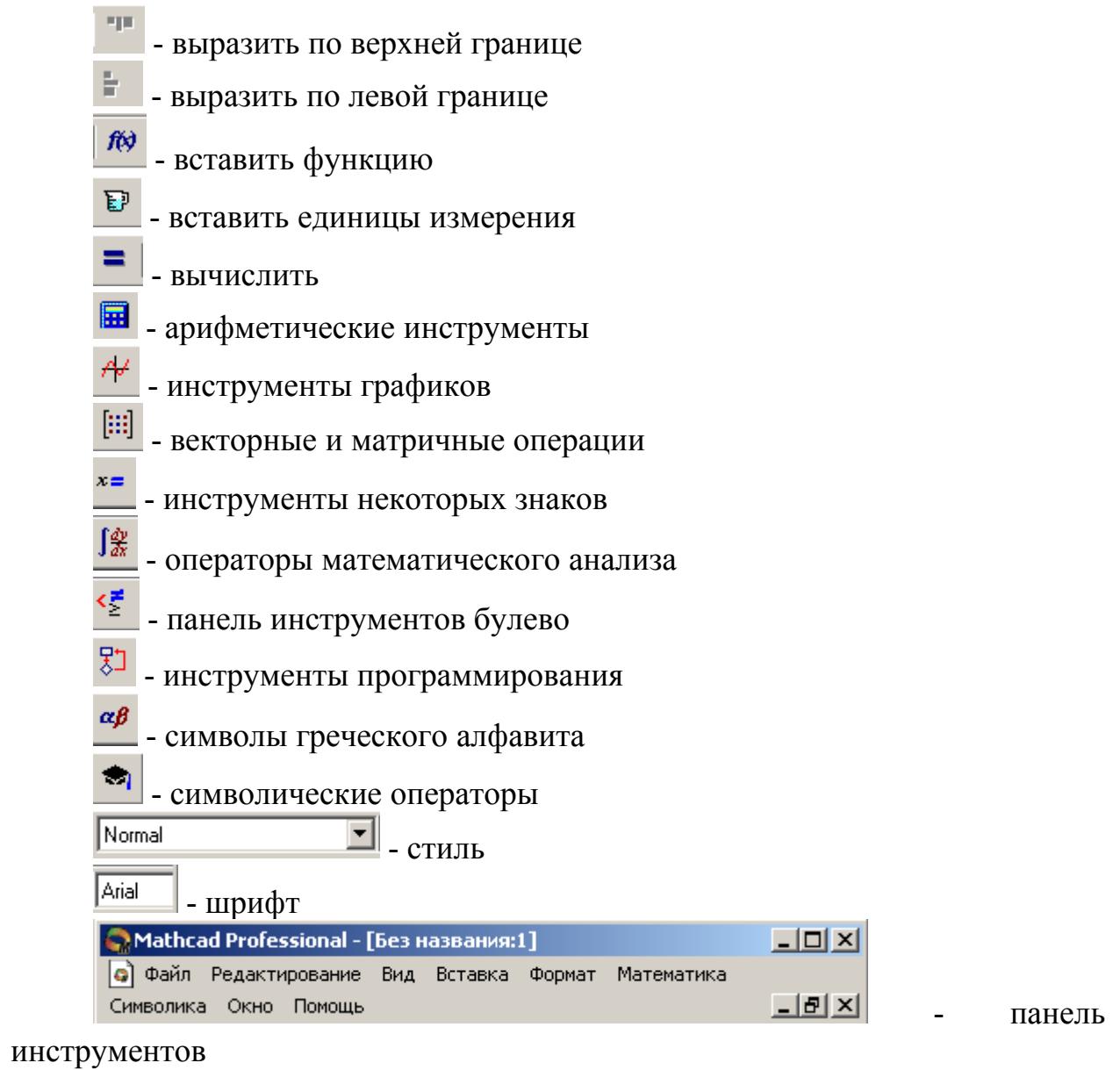
- вырезать

- копировать

- вставить

- отмена

- возврат



#### 4.4 Интерфейс пользователя

В MathCAD интерфейс пользователя интуитивен и сходен с другими приложениями Windows. Его составные части:

- верхнее меню, или строка меню (menu bar);
- панели инструментов (toolbars) Standard (Стандартная) и Formatting(Форматирование);
- панель инструментов Math (Математика) и доступные через нее дополнительные математические панели инструментов;
- рабочая область (worksheet);
- строка состояния (status line, или status bar);
- всплывающие, или контекстные, меню (pop-up menus, или context menus);
- диалоговые окна, или диалоги (dialogs).

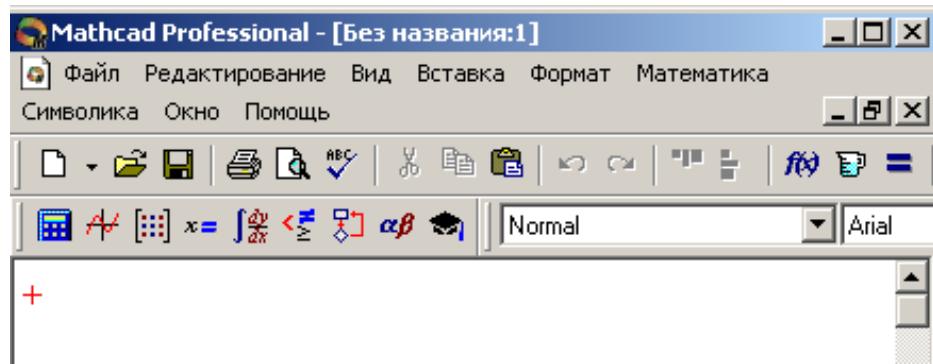
Большинство команд можно выполнить как с помощью меню (верхнего или контекстного), так и панелей инструментов или клавиатуры.

##### Меню

Строка меню располагается в самой верхней части окна MathCAD. Она содержит девять заголовков, щелчок мышью на каждом из которых приводит к появлению соответствующего меню с перечнем сгруппированных по действию команд: - File (Файл) - команды, связанные с созданием, открытием, сохранением, пересылкой по электронной почте и распечаткой на принтере файлов с документами; - Edit (Правка) - команды, относящиеся к правке текста (копирование, вставка, удаление фрагментов и т. п.); - View (Вид) - команды, управляющие внешним видом документа в окне редактора MathCAD, а также команды, создающие файлы анимации; - Insert (Вставка) - команды вставки различных объектов в документы; - Format (Формат) - команды форматирования текста, формул и графиков; - Math (Математика) - команды управления вычислительным процессом; - Symbolics (Символика) - команды символьных вычислений; - Window (Окно) - команды управления расположением окон с различными документами на экране; - Help (Справка) - команды вызова контекстно-зависимой справочной информации, доступа к Центру Ресурсов, опции Совета Дня и сведений о версии и программы.

### **Панели инструментов**

Панели инструментов служат для быстрого (в один щелчок мыши) выполнения наиболее часто применяемых команд. Все действия, которые можно выполнить с помощью панелей инструментов, доступны и через верхнее меню. На [рисунке 4.2](#) изображено окно MathCAD с тремя основными панелями инструментов, расположенными непосредственно под строкой меню. Кнопки в панелях сгруппированы по сходному действию команд: - Standard (Стандартная) - служит для выполнения большинства операций, таких как действия с файлами, редакторская правка, вставка объектов и доступ к справочным системам; - Formatting (Форматирование) - для форматирования (изменения типа и размера шрифта, выравнивания и т. п.) текста и формул; - Math (Математика) - для вставки математических символов и операторов в документы. Группы кнопок на панелях инструментов разграничены по смыслу вертикальными линиями - *разделителями*. При наведении указателя мыши на любую из кнопок рядом с кнопкой появляется *всплывающая подсказка* - короткий текст, поясняющий назначение кнопки. Наряду со всплывающей подсказкой, более развернутое объяснение готовящейся операции можно отыскать на строке состояния.



*Рисунок 4.2 - Основные панели инструментов*

Панель Math (Математика) предназначена для вызова на экран еще девяти панелей, с помощью которых, собственно, и происходит вставка математических операций в документы. В прежних версиях MathCAD эти математические панели инструментов назывались *палитрами* (palettes) или *наборными панелями*. Чтобы показать какую-либо из них, нужно нажать соответствующую кнопку на панели Math . Перечислим назначение математических панелей (рисунок 4.3):

- Calculator (Калькулятор) - служит для вставки основных математических операций, получила свое название из-за схожести набора кнопок с кнопками типичного калькулятора;
- Graph (График) - для вставки графиков;
- Matrix (Матрица) - для вставки матриц и матричных операторов;
- Evaluation (Выражения) - для вставки операторов управления вычислениями;
- Calculus (Вычисления) - для вставки операторов интегрирования, дифференцирования, суммирования;
- Boolean (Булевы операторы) - для вставки логических (булевых) операторов;
- Programming (Программирование) - для программирования средствами MathCAD;
- Greek (Греческие символы) - для вставки греческих символов;
- Symbolic (Символика) - для вставки символьных операторов.

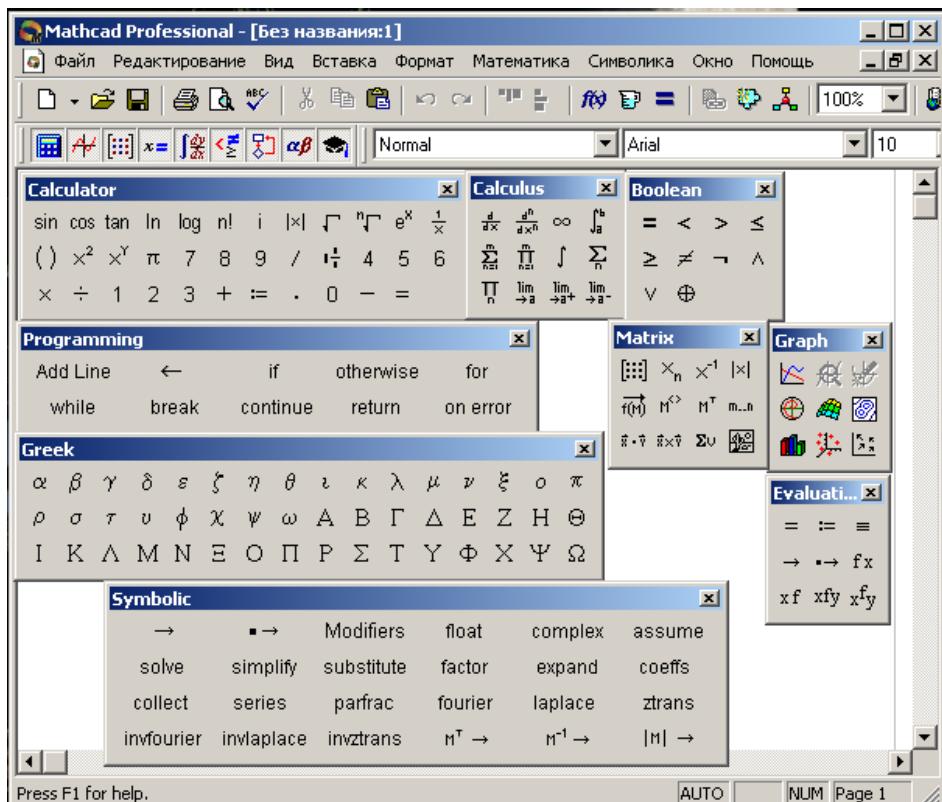


Рисунок 4.3 - Математические панели инструментов

При наведении указателя мыши на многие из кнопок математических панелей появляется всплывающая подсказка, содержащая еще и сочетание "горячих клавиш", нажатие которых приведет к эквивалентному действию. Ввод действий с клавиатуры часто удобнее нажатия кнопок панелей инструментов, но требует большего опыта.

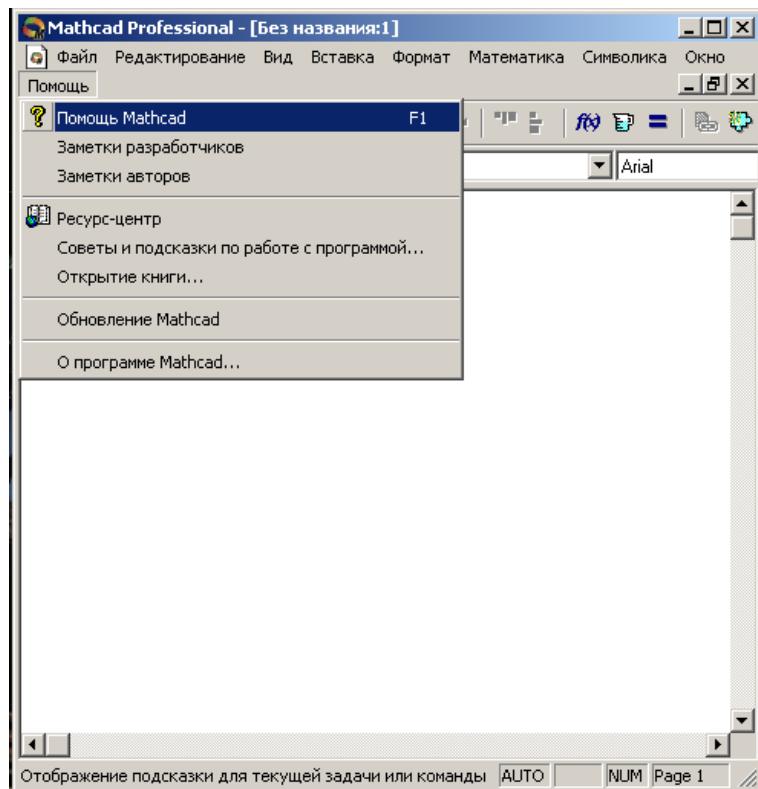
#### Строка состояния

В нижней части окна MathCAD, под горизонтальной полосой прокрутки, НЕ большинстве рисунков этой главы видна *строка (линия) состояния* (рисунок 4.4). На ней отображается самая основная информация о режиме редактирования разграниченная разделителями (слева направо): - контекстно-зависимая подсказка о готовящемся действии; - режим вычислений: автоматический (AUTO) или задаваемый вручную (Calc F9); - текущий режим раскладки клавиатуры CAP; - текущий режим раскладки клавиатуры NUM; - номер страницы, на которой находится курсор



Рисунок 4.4 - Страна состояния

Чтобы показать или скрыть строку состояния, выполните команду View/Status Bar (Вид / Страна состояния).



*Рисунок 4.5 - Меню Help*

Если в какой-либо момент работы с MathCAD вам потребовалась помощь, выберите Помощь/MathCAD помощь, либо нажмите клавишу <F1> (рисунок 4.5).

## Лекция 5 (2 часа)

*Системы управления базами данных (СУБД)*

*Основные положения*

*Архитектура систем управления базами данных*

*Иерархическая и сетевая даталогические модели СУБД*

*Реляционные даталогические модели СУБД*

### **5. Системы управления базами данных (СУБД)**

Основные идеи современной информационной технологии базируются на концепции **баз данных** (БД), ранее упоминаемых достаточно часто без какого-либо их детального пояснения. Согласно данной концепции основой информационной технологии являются **данные**, организованные в БД, адекватно отражающие реалии действительности в той или иной предметной области и обеспечивающие пользователя актуальной информацией в соответствующей предметной области. Первые БД появились уже на заре 1-го поколения ЭВМ, представляя собой отдельные файлы данных или их простые совокупности. По мере увеличения объемов и структурной сложности хранимой информации, а также расширения круга потребителей информации определилась необходимость создания удобных и эффективных систем интеграции хранимых данных и управления ими. В конце 60-х годов это привело к созданию первых **коммерческих систем управления базами данных** (СУБД), поддерживающих организацию и ведение БД. Перед обсуждением последующего материала нам потребуется ряд основных понятий, используемых в информационных системах различного назначения.

#### **5.1 Основные положения**

База данных (БД) в строгом смысле слова представляет собой совокупность взаимосвязанных файлов данных определенной организации. БД, как правило, включает целый ряд файлов, но может состоять и из единственного файла. Данные, составляющие БД, отражают характеристики объектов и их отношений в соответствующей прикладной области. Каждый файл, входящий в БД, содержит определенное число записей (изменяемое в процессе функционирования БД), отражающих ту или иную сторону предметной области, на которую ориентирована БД. Как правило, файлы БД содержат большое число однотипных записей. Записи, в свою очередь, состоят из полей, представляющих определенные типы информации об объектах. Поле является наименьшей информационной единицей, непосредственно доступной в записи. Если файл БД содержит  $n$  однотипных записей (имеющих одинаковую структуру полей и их смысловую нагрузку), то запись файла состоит из фиксированного набора (кортежа) полей А1-А $k$ , каждое из которых содержит в общем случае различного типа информацию.

При наличии БД прикладные программы могут использовать ее информацию (записи и их поля) для решения конкретных задач в прикладной области, на которую ориентирована данная БД (рисунок 5.1).

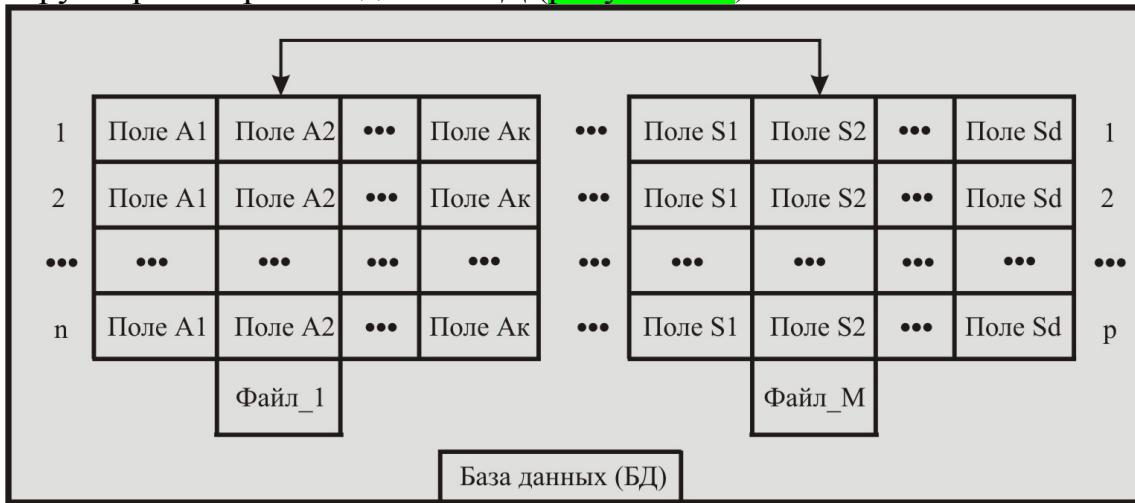


Рисунок 5.1 – Файловая организация баз данных (файлы, записи, поля)

Пользователями БД являются *четыре* основные категории потребителей ее информации и/или поставщиков информации для нее: (1) **конечные пользователи**, (2) **программисты** и **системные аналитики**, (3) **персонал** поддержки БД в актуальном состоянии и (4) **администратор** БД. Хорошо спроектированные *системы управления* БД (СУБД), используют развитые *графические* интерфейсы и поддерживают системы отчетов, отвечающие специфике пользователей указанных *четырех* категорий. В этом случае персонал поддержки БД и конечные пользователи могут легко осваивать и использовать СУБД для обеспечения своих потребностей без какой-либо специальной подготовки, т.е. *специфика функционирования* данных систем скрыта от пользователя. Более того, хорошо спроектированные СУБД предоставляют опытному пользователю средства для создания собственных БД-приложений, не требуя от него специальной программистской подготовки. **Конечным** пользователям для обеспечения доступа к информации БД предоставляется графический интерфейс, как правило, в виде системы окон с функциональными меню, позволяющими легко получать необходимую информацию на экран и/или принтер в виде удобно оформленных отчетов.

**Программисты** и **системные аналитики** используют СУБД совершенно в ином качестве, обеспечивая разработку новых БД-приложений, поддерживая и модифицируя (при необходимости) уже существующие. Для данной группы пользователей СУБД требуются средства, обеспечивающие указанные функции (создание, отладка, редактирование и т.д.). Пользователи **третьей** категории нуждаются в интерфейсе, как правило, графическом для обеспечения задач поддержания БД в актуальном состоянии. Эти пользователи состоят в штатах подразделений функциональных и/или

обработки информации, обеспечивающих прикладную область, и отвечают за актуальное состояние соответствующей ей БД (контроль текущего состояния, удаление устаревшей информации, добавление новой и т.д.). **Программисты** выполняют своего рода *посреднические* функции между БД и **конечными** пользователями. И если на первых этапах развития БД-технологии они составляли весьма многочисленную группу пользователей, то в процессе развития СУБД и, прежде всего, массового использования **ПК** эта категория сходит на нет. Особую и ответственную роль выполняет **администратор**, отвечающий как за актуальность находящейся в БД информации, так и за корректность функционирования и использования БД и СУБД.

В случае больших БД может быть достаточно много **конечных** пользователей, ряд *программистов* и несколько *администраторов* БД; в случае небольших БД (что особенно характерно для ПК) все эти функции могут обеспечиваться одним человеком. Важные функции выполняет *администратор* БД, отвечающий за выработку требований к БД, ее проектирование, реализацию, эффективное использование и сопровождение. Необходимость в таком специалисте вытекает из принципа *независимости* данных, а также диктуется важностью БД в деятельности организаций и более крупных объединений - поставщиков и потребителей информации БД. *Администратор* БД взаимодействует с пользователями в определении требований к базе в процессе выработки требований к системе в целом, пользуется языком описания данных для определения БД в процессе проектирования системы, взаимодействует с программистами, которые создают ПО, использующее доступ к БД, отвечает за загрузку БД информацией в процессе реализации системы, контролирует работоспособность БД, используя соответствующие *программные* и *аппаратные* средства, и определяет, когда следует реорганизовывать данные в базе или начать работы по созданию новой, более совершенной БД. В целом, функции *администратора* БД сводятся к поддержанию целостности БД, необходимого *уровня* защиты ее данных и эффективности. Среди его *наиболее важных обязанностей* - *согласование* конфликтующих требований, которое требуется достаточно часто, ибо БД обслуживает, как правило, целый ряд различных прикладных процессов.

Как уже отмечалось, БД представляет собой *совокупность* логически взаимосвязанных *файлов данных* определенной организации;

для определения и обращения к такой файловой совокупности используются средства *системы управления* БД (СУБД). СУБД представляет собой совокупность *лингвистических* и *программных* средств, предназначенных для создания, ведения и совместного использования БД многими пользователями. Тогда как под *системой* БД понимается СУБД с *наполненной* соответствующей информацией БД, управляемой ее средствами. Это означает, *во-первых*, что совокупность файлов БД определяется

посредством схемы, не зависящей от программ, которые к ней обращаются, и, *во-вторых*, что она реализована на основе ВП *прямого доступа*. Использование СУБД обеспечивает лучшее управление данными, более совершенную организацию файлов и более простое обращение к ним по сравнению с обычными способами хранения информации. Вследствие более совершенных механизмов доступа БД, как правило, имеют более сложную организацию, чем обычные файлы, объединяя *данные*, ранее хранящиеся во многих отдельных файлах. Размер и сложность не являются определяющими характеристиками БД - наличие СУБД для ПК и даже в среде ряда пакетов (например, табличных процессоров, интегрированных и др.) приводит к созданию большого числа относительно простых и небольших БД, достоинством которых (при наличии соответствующих СУБД) являются простота определения и доступа к данным. Под *банком данных* (БнД) понимается система *лингвистических, программных, аппаратных и организационных* средств, основанная на БД-технологии и предназначенная для *централизованного накопления и коллективного использования* данных в той или иной прикладной области. Тогда как *система обработки информации* (СОИ) реализует автоматизированный сбор, обработку и хранение информации, включая соответствующие лингвистические, программные, аппаратные, организационные средства и обслуживающий их персонал.

Под *целостностью* БД понимается *актуальное* состояние ее данных, отражающих состояние некоторой реальной прикладной области и подчиняющихся правилам *непротиворечивости*. Под *языком* БД понимается один или совокупность языков, обеспечивающих описание данных, манипулирование с данными. Конкретный язык БД всегда ассоциируется с конкретной СУБД. СУБД представляет собой средства обработки на языке базы данных, позволяющие обрабатывать обращения к БД, поступающие от прикладных программ и/или конечных пользователей, и поддерживать *целостность* БД. Таким образом, СУБД имеет свойства, характерные как для *компиляторов*, так и для ОС, однако по сравнению с *первыми* обеспечивается более высокий уровень *абстрагирования*, что оказывается очень полезным как для *программистов*, так и для *конечных пользователей*.

## 5.2 Архитектура систем управления базами данных

Большинство современных СУБД включает следующие пять *основных* компонент обеспечивающих работу с БД широкому кругу прикладных процессов (**рисунок 5.2**):



Рисунок 5.2 – Принципиальный состав основных компонент СУБД

Остановимся несколько детальнее на каждой из этих компонент. В общем случае **лингвистическое** обеспечение СУБД включает ряд языков, обеспечивающих интерфейс с БнД пользователям различного уровня. Из данных **языковых** средств можно выделить две основные группы: **языки описания данных** (ЯОД) и **языки работы с БД** (ЯрБД). Группа ЯОД предназначена для описания *структур данных* и *отношений* между ними, поддерживаемых СУБД. Каждая конкретная СУБД располагает своим ЯОД, но все ЯОД поддерживают один и тот же набор основных функций. В зависимости от своего назначения ЯОД можно разделить, в свою очередь, на **языки описания БД** и **языки описания внешних данных**. Если *первые* предназначены для описания состава и логической организации БД (т.е. собственно самой хранящейся информации), то *вторые* - для описания *внешней* по отношению к БД информации (входной, выходной, сообщений и т.д.).

При необходимости написания прикладных программ работы с БД используется **ЯУД**, позволяющий организовывать интерфейс (*обмен информацией*) пользовательских программ, написанных на ЯВУ, с БД. Как правило, ЯУД базируется на том же ЯВУ, на котором создаются БД-приложения (например, *Pascal, C, Cobol*).

Языки группы ЯУЗ относятся к языкам 4-го поколения, ибо позволяют непосредственно запрашивать (на языке, близком к естественному) необходимую информацию из БД, минуя целую цепочку требуемых для этого процедур. Базируясь на лексиконе естественных языков в соответствующей прикладной области, ЯУЗ значительно легче в освоении, чем традиционные ЯВУ 3-го поколения (*Basic, Pascal, C, Cobol* и др.).

Стандартом среди языков данной группы является SQL- *структурированный язык запросов*.

Набор *общих утилит* (смотри рисунок 5.2) предназначен для обеспечения наиболее часто используемых процедур работы с данными и файлами БД (редактирование данных, удаление записей, создание новых файлов и т.д.) Как правило, *общие утилиты* обеспечивают интерфейс с пользователем на уровне командного языка и используются персоналом ведения БнД в случае больших БД, функционирующих на мощных мини-ЭВМ или ЭВМ общего назначения. На ПК эти средства используются непосредственно его пользователем и интерфейс с ними упрощен.

*Генератор приложений* представляет собой компоненту СУБД, обеспечивающую пользователя средствами создания БД-приложений без их программирования. На основе описания задачи *генератор* из программных модулей, находящихся в специальных библиотеках, *компилирует* (собирает и редактирует) программную систему, обеспечивающую решение поставленной задачи.

*Генераторы отчетов* позволяют выводить результаты работы с БнД в виде *отчетов*, оформленных по требованию пользователя, используя достаточно простой язык (командный, табличный и др.).

Одной из важных особенностей СУБД является их *многофункциональность*, диапазон которой определяется степенью ее *функциональной полноты*.

*Функционально полная* СУБД должна включать в свой состав средства, обеспечивающие нужды пользователей различных категорий и уровней на всех этапах *жизненного* цикла БнД (проектирование, разработка и эксплуатация), который для данного типа систем может составлять несколько десятков лет (*IMS, IDMS* и др.).

*Функционально полную* СУБД можно определять следующим составом ее функциональных характеристик:

- поддерживаемая системой даталогическая модель;
- средства администратора БД;
- средства разработки БД-приложений;
- интерфейсы с пользователями и другими БД-приложениями;
- интерфейсы с другими СУБД;
- средства обеспечения сетевой и распределенной обработки информации.

Само же понятие *функциональной полноты* СУБД постоянно расширяется с учетом новых информационных технологий.

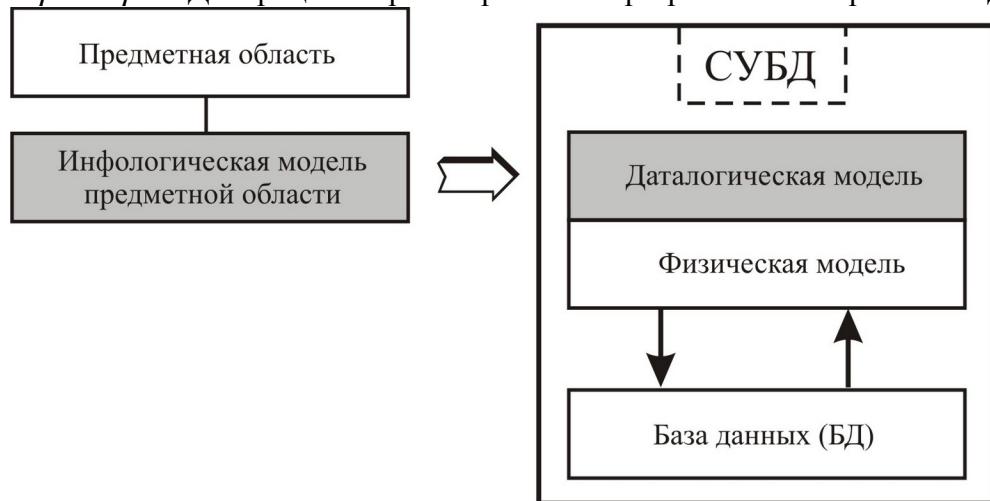
В настоящее время реализация СУБД традиционными программными технологиями и аппаратными средствами не удовлетворяет многие прикладные области - потребителей и/или поставщиков информации БД/БЗ. В первую очередь, это относится ко многим задачам САПР, управления БД/БЗ в режиме *реального* времени, работы с *дедуктивными* БД и др.

Особенно актуальной проблема становится при реализациях интегрированного управления БД и БЗ. Поэтому одним из основных современных подходов к обеспечению необходимой эффективности управления информационными ресурсами в совокупности с использованием быстро развивающейся технологией управления БД является применение подхода на основе *специализированных ВС - машинах баз данных и знаний* (МБДЗ), интенсивно исследуемых и разрабатываемых во многих ведущих странах. При этом под МБДЗ понимается автономная ВС, реализующая функционально полный **набор** процедур управления БД/БЗ и предназначенная для поддержки развитых интерфейсов конечных пользователей и прикладных программ в абонентских ЭВМ, связанных каналами связи с *первой*.

В силу функционального назначения СУБД имеют как *совпадающие* по функциональным возможностям, так и *существенно различные* характеристики, зависящие от ряда конкретных обстоятельств и, в первую очередь, от поддерживаемой *даталогической модели* и используемого класса ЭВМ (ПК, мини- и ЭВМ общего назначения).

### 5.3 Иерархическая и сетевая даталогические модели СУБД

Каждая БнД содержит и обрабатывает информацию из конкретной **прикладной области**, представляющей интерес для определенных приложений. Описание *предметной области* без акцента на ее последующие БнД - реализации определяет **инфологическую модель** предметной области (**рисунок 5.3**). Инфологическая модель является исходной для построения даталогической модели БД и служит промежуточной моделью для специалистов *предметной области* (для которой создается БнД) и *администратора БД* в процессе проектирования и разработки конкретной БнД.



*Рисунок 5.3 – Принципиальная организация системы обработки информации (СОИ) на основе БД-технологии*

Под *даталогической* понимается *модель*, отражающая логические взаимосвязи между элементами данных безотносительно их содержания и физической организации. При этом *даталогическая* модель разрабатывается с учетом конкретной реализации СУБД, а также с учетом специфики конкретной *предметной области* на основе ее *инфологической* модели. Для конкретной реализации *даталогической* модели проектируется *физическая* модель, (отображающая *первую* на конкретные программные и аппаратные средства ОС, внешняя память, работа с данными на физическом уровне и т.д.). Наполненная конкретной информацией *физическая* модель и составляет собственно БД. Система, обеспечивающая соответствующее совместное функционирование указанных компонент, и составляет суть конкретной СУБД.

Современные СУБД допускают целый ряд классификаций в зависимости от уровня их рассмотрения (в целом либо по совокупностям их функциональных характеристик): по интерфейсу с пользователем в зависимости от поддерживаемых моделей, по назначению и режимам функционирования, по способу обработки информации и т.д. Мы кратко остановимся на моделях *даталогического* уровня, который берется за основу большинства современных классификаций СУБД.

Обычно различают три класса СУБД, обеспечивающих работу *иерархических, сетевых и реляционных* моделей. Однако *различия* между этими классами *постепенно* стираются, причем, видимо, будут появляться и другие классы, что вызывается, прежде всего, интенсивными работами в области *баз знаний* (БЗ) и объектно-ориентированной инфотехнологией, о которой будет идти речь ниже. Поэтому традиционной классификацией пользуются все реже, но мы пока будем придерживаться именно ее, как наиболее устоявшуюся. Каждая из указанных моделей обладает характеристиками, делающими ее наиболее удобной для конкретных приложений. Одно из основных *различий* этих моделей состоит в том, что для *иерархических* и *сетевых* СУБД их структура часто не может быть изменена после ввода данных, тогда как для *реляционных* СУБД структура может изменяться в любое время. С другой стороны, для больших БД, структура которых остается длительное время неизменной, и постоянно работающих с ними приложений с интенсивными потоками запросов на БД-обслуживание именно *иерархических* и *сетевых* СУБД могут оказаться наиболее эффективными решениями, ибо они могут обеспечивать более быстрый доступ к информации БД, чем *реляционные* СУБД.

Однако прежде чем переходить непосредственно к *иерархическим* СУБД, кратко рассмотрим *файловую* модель, неправомерно относимую довольно часто к СУБД. *Файловая* модель представляет собой набор файлов данных определенной структуры, но связь между данными этих файлов отсутствует. Естественно, программные средства работы с таким образом организованной *инфобазой* могут устанавливать связь между данными ее

файлов, но на *концептуальном* уровне файлы модели являются *независимыми*. Системы, обеспечивающие работу с *файловыми инфобазами*, называют *системами управления файлами* (СУФ) и они оказываются весьма эффективными во многих приложениях. СУФ используются на всех классах ЭВМ, но особенно они распространены для обработки информации на ПК. *Файловые системы* легко осваиваются, достаточно просты и эффективны в использовании и, как правило, для работы с ними используются *простые языки запросов* либо и вовсе ограничиваются набором программ-утилит. Такие системы обычно поддерживают работу с небольшим числом файлов, содержащих ограниченное число записей с небольшим количеством полей.

**Иерархические** модели СУБД имеют *древовидную* структуру, когда каждому узлу структуры соответствует один *сегмент*, представляющий собой поименованный линейный кортеж *полей данных*. Каждому сегменту (кроме *Sl-корневого*) соответствует один *входной* и несколько *выходных* сегментов (рисунок 5.4). Каждый *сегмент* структуры лежит на единственном иерархическом пути, начинающемся от *корневого* сегмента.

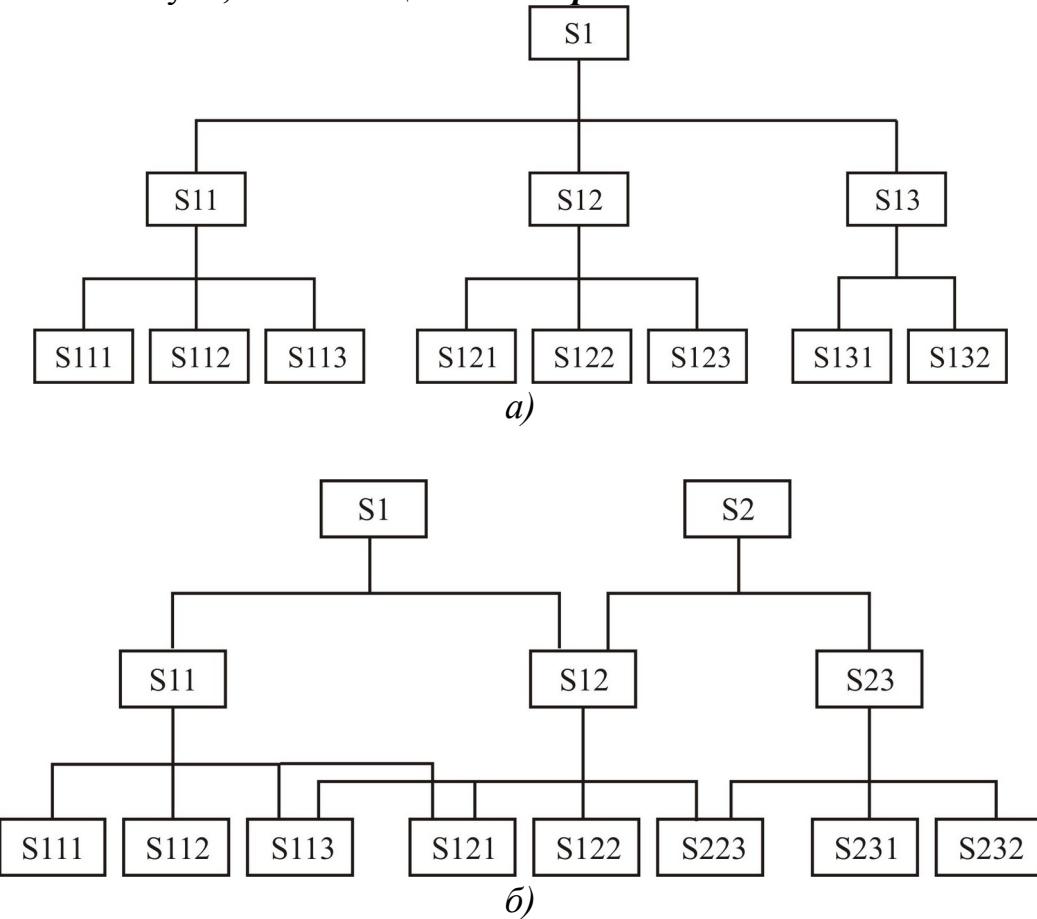


Рисунок 5.4 – Структура иерархической (а) и сетевой (б) СУБД

Для описания такой логической организации данных ЯОД достаточно предусматривать для каждого сегмента данных только идентификацию *входного* для него сегмента. Так как в иерархической модели каждому

входному сегменту данных соответствует  $N$  выходных, то такие модели весьма удобны для представления отношений типа  $1:N$  в *предметной* области. Следует отметить, что в настоящее время не разрабатываются СУБД, поддерживающие на концептуальном уровне только *иерархические* модели. Как правило, использующие *иерархический* подход системы допускают связывание *древовидных* структур между собой и/или установление связей внутри них. Это приводит к *сетевым* даталогическим моделям СУБД. К основным недостаткам *иерархических* моделей следует отнести: неэффективность реализации отношений типа  $N:N$ , медленный доступ к сегментам данных нижних уровней иерархии, четкая ориентация на определенные типы запросов и др. В связи с *этими* недостатками ранее созданные *иерархические* СУБД подвергаются существенным модификациям, позволяющим поддерживать более сложные типы структур и, в первую очередь, *сетевые* и их модификации.

**Сетевая** даталогическая модель СУБД во многом подобна *иерархической*: если в иерархической модели (смотри рисунок 5.4а) для каждого сегмента записи допускается *только* один входной сегмент при  $N$  выходных, то в *сетевой* модели для сегментов допускается *несколько* выходных сегментов наряду с возможностью наличия сегментов без входов с точки зрения иерархической структуры. На рисунке 5.4б представлен простой пример *сетевой* структуры, полученной на основе *модификации* иерархической структуры (смотри рисунок 5.4а). Графическое изображение структуры связей сегментов в такого типа моделях представляет собой *сеть*. Сегменты данных в *сетевых* БД могут иметь множественные связи с сегментами старшего уровня. При этом направление и характер связи в *сетевых* БД не являются столь очевидными, как в случае *иерархических* БД. Поэтому имена и направление связей должны идентифицироваться при описании БД средствами ЯОД.

Таким образом, под *сетевой* СУБД понимается система, поддерживающая сетевую организацию: любая запись, называемая записью *старшего* уровня, может содержать данные, которые относятся к набору других записей, называемых записями *подчиненного* уровня. Возможно обращение ко всем записям в наборе, начиная с записи старшего уровня. Обращение к набору записей реализуется по указателям. В рамках сетевых СУБД легко реализуются и иерархические даталогические модели. *Сетевые* СУБД поддерживают сложные соотношения между типами данных, что делает их пригодными во многих различных приложениях. Однако пользователи таких СУБД ограничены связями, определенными для них разработчиками БД-приложений. Более того, подобно иерархическим *сетевые* СУБД предполагают разработку БД-приложений *опытными* программистами и системными аналитиками.

Для работы с записями в БД CODASYL-модели используется DML-язык манипулирования данными. Сетевая CODASYL-модель накладывает

меньше ограничений на логическую структуру БД, чем другие сетевые модели, что существенно облегчает отображение предметной области в даталогическую модель СУБД и расширяет круг ее приложений.

Среди недостатков сетевых СУБД следует особо выделить проблему обеспечения сохранности информации в БД, решению которой уделяется повышенное внимание при проектировании *сетевых* БД. Примерами известных *сетевых* СУБД являются: *DBMS, IDS, TOTAL, IDMS*,

#### **5.4 Реляционные даталогические модели СУБД**

СУБД *реляционного* типа являются наиболее распространенными на всех классах ЭВМ, а на ПК занимают доминирующее положение. *Реляционной* называется СУБД, в которой средства управления БД поддерживают *реляционную* модель данных. Концепция *реляционной* модели была предложена в 1970г. Е. Коддом и имеет большое значение в деле организации работы с БД. Данная модель позволяет определять: (1) операции по запоминанию и поиску данных; (2) ограничения, связанные с обеспечением целостности данных. Модель основана на математическом понятии *отношения*, расширенном за счет значительного добавления специальной терминологии и развития соответствующей теории. В такой модели общая структура данных (*отношение*) может быть представлена в виде таблицы, в которой каждая строка значений (*кортеж*) соответствует *логической записи*, а заголовки столбцов являются названиями *полей* (*элементов*) записи. Операции запоминания и поиска делятся на две группы: операции на *множествах* (объединение, пересечение, разность, произведение) и *реляционные* операции (выбрать, спроектировать, соединить, разделить). Любой *язык* манипулирования данными, обеспечивающий все эти операции, является *реляционно полным*. Для увеличения эффективности работы во многих СУБД *реляционного* типа приняты ограничения, соответствующие строгой *реляционной модели*. Многие *реляционные* СУБД представляют файлы БД для пользователя в табличном формате - с *записями* в качестве строк и их *полями* в качестве столбцов. В табличном виде информация воспринимается значительно легче. Однако в БД на физическом уровне данные хранятся, как правило, в файлах, содержащих последовательности записей. Основным преимуществом *реляционных* СУБД является возможность связывания на основе определенных соотношений файлов БД. Со *структурной* точки зрения реляционные модели являются более простыми и однородными, чем иерархические и сетевые. В *реляционной* модели каждому объекту предметной области соответствует одно или более отношений. При необходимости определить связь между объектами явно, она выражается в виде отношения, в котором в качестве атрибутов присутствуют идентификаторы взаимосвязанных объектов. В

*реляционной* модели объекты предметной области и связи между ними представляются одинаковыми информационными конструкциями, существенно упрощая саму модель.

Файл авторов публикаций БД			
Автор	Адрес	Телефон	Число публ.
•••	•••	•••	•••
Иванов	Москва	935-45-80	160
Петров	Тула	49-35-87	180
Сидоров	Тверь	3-45-27	75

Файл публикаций РБД				
Назв. публикации	Автор	Тип публ.	Дата	Объем в п.л
Основы ...	Иванов	статья	06.03.2004	2,5
Проблема .	Петров	книга	23.09.2001	35
Теория ...	Сидоров	статья	02.07.2005	3,8
•••	•••	•••	•••	•••

Рисунок 5.5 – Простой пример, иллюстрирующий принцип реляционной модели

СУБД считается *реляционной* при выполнении следующих двух предложенных еще Э.Коддом: (1) поддерживает *реляционную* структуру данных и (2) реализует по крайней мере операции *селекции, проекции и соединения* отношений. В последующем был создан целый ряд *реляционных* СУБД, в той или иной мере отвечающих данному определению. Многие СУБД представляют собой существенные расширения *реляционной* модели, другие являются *смешанными*, поддерживая несколько даталогических моделей (например, уже упомянутые NOMAD, ИНТЕРБАЗА).

Для работы с данными реляционные СУБД используют как языки, основанные на *реляционной* алгебре, так и другие типы языков, включая *процедурные* (Cobol, С и др.) и *непроцедурные* (Prolog, Lisp).

Суть *реляционной* СУБД можно пояснить на следующем простом примере (рисунок 5.5). В некоторой *реляционной* БД (РБД) имеются два файла *авторов* и *публикаций*, каждый из которых содержит определенное число записей, состоящих из фиксированного числа полей (соответственно 4 и 5), представляющих данные по соответствующим элементам предметной области (рисунок 5.5). Можно сказать, что определены два *отношения* (файла), имеющие общий элемент - значения поля *Автор*. Операции *реляционной* алгебры могут объединять два типа записей по этому общему

элементу. Например, в результате соединения запись **Петров** может представится в следующем виде:

*Петров <Тула> <49-*

*35-87><180><Проблема...><книга><23.09><35>.....*

т.е. к сведениям об авторе добавляются сведения о всех его публикациях, имеющихся в РБД. Связь между записями допускается по нескольким полям, позволяя образовывать достаточно сложные операции. Поля данных, связывающие вместе две записи, могут быть уникальными для данной пары, но могут дублироваться и во многих других записях. Они могут повторяться неоднократно, связывая между собой записи. Аналогичным образом можно проиллюстрировать выполнение в *реляционной* модели операций *проекции* и *селекции*.

*Реляционная* СУБД должна четко отслеживать взаимосвязи записей в БД во избежание потери или искажения информации. С этой целью СУБД постоянно пересчитывает число связей для каждой записи БД в прямом и обратном направлениях, что требует существенных временных затрат для больших БД. Простота и стройность реляционной алгебры делают ее весьма привлекательной для организации *реляционных* БД, что мы и видим, прежде всего, для класса ПК. Однако в действительности *реальные* данные предметной области не укладываются в указанную модель (например, отношения могут содержать повторяющиеся записи и т.д.). Поэтому наряду с сугубо реляционными существуют и другие даталогические модели СУБД и их различные модификации и сочетания, обеспечивая широкий круг решаемых на их основе информационных, коммерческих, управлеченческих, финансовых, вычислительных и других типов задач. Из наиболее известных примеров реляционных СУБД можно отметить такие, как: RAPPORT, dBase, DB/2, ORACLE, QBE, INGRES, SQL/DS, Paradox, R:Base и ряд других.

Массовое развитие класса ПК оказало весьма существенное влияние на развитие инфотехнологии и БД-технологии в частности, привнося элементы последней в массовую инфотехнологию. Прежде всего, этому способствовало развитие мощной индустрии по созданию разнообразных СУБД для ПК. Если создание СУБД для ЭВМ общего назначения и (в значительной мере) **мини-ЭВМ** занимало длительный промежуток времени и число таких коммерческих СУБД было невелико - практически весь их перечень был на слуху у специалистов по компьютерной инфотехнологии, то с появлением класса ПК наряду с мощным развитием для них ПС различного назначения начали быстро появляться СУБД. При этом БД-технология начала активно проникать и в ПС другого назначения (электронные таблицы, интегрированные и статистические пакеты и т.д.). К БД-технологии были приобщены широкие круги пользователей ПК. Во многих разработках для ПК начали применяться собственные СУБД различных организаций и назначения. Так при разработке пакета *Metrolog* для первого отечественного ПК ИСКРА 226 нами была создана хорошо зарекомендовавшая себя СУБД

смешанного типа, поддерживающая *файловую* и *иерархическую* даталогические модели. На наш взгляд, ряд причин способствовал такому массовому использованию БД-технологии:

- массовое использование ПК в приложениях, предопределяющих работу с БД;
- резкое уменьшение цикла разработки ПС из-за персонального характера работы;
- наличие достаточно развитых системных и инструментальных средств;
- наличие внешней памяти большой емкости на "винчестерах".

Эти и другие причины обеспечили как широкий спрос на СУБД для ПК, так и хорошие предпосылки для его быстрого удовлетворения. Наряду с мощными фирмами, специализирующимися на разработке коммерческих СУБД для ПК (*Ashton-Tate*, *Microrim*, *Fox Software*, *Nantucket* и др.), к разработкам и/или адаптации уже готовых СУБД для ПК приступили и крупные фирмы, ранее ориентированные в этой области на приложения к ЭВМ других классов (*Oracle*, *IBM*, *Relational Technology* и др.). Все это способствовало интенсивному проникновению БД-технологии в массовую инфообработку. С другой стороны, широкое использование ПК в весьма обширном спектре прикладных областей способствовало выдвижению к СУБД целого ряда **актуальных** требований и, в первую очередь, по повышению уровня интерфейсов с пользователем и другими приложениями.

Разработанное в настоящее время большое число различного назначения СУБД позволяет создавать и эксплуатировать системы БД на всех классах и типах ЭВМ, поддерживая различные даталогические модели и обеспечивая нужды широкого круга приложений. В табл. 12 приведена сводка из 20 наиболее популярных СУБД.

**Таблица 5.1** представляет наиболее популярные СУБД с краткой их характеристикой; в ней приняты следующие сокращения: *O* - общего назначения и *M* - мини-ЭВМ. При этом можно с полной уверенностью определить двух явных лидеров на рынке СУБД: *dBase* и ей подобные системы для класса ПК и *ORACLE* для мини - и общего назначения ЭВМ.

Средства современных СУБД настолько разнообразны, что способны удовлетворить потребности самого широкого круга пользователей - от *профессионала* в области разработки систем БД различных типа и назначения до *пользователя*, не обладающего достаточным уровнем компьютерной грамотности.

Таблица 5.1 – Популярные СУБД и их характеристики

№	СУБД	Класс ЭВМ	Тип модели БД	Язык запросов
1	Adabas	О, М, ПК	Сетевая	Natural, SQL
2	Clipper	ПК	Реляционная	Собственный
3	DataBase	ПК	Реляционная	DQL
4	dBase	ПК	Реляционная	DQL
5	DB Vista	ПК	Сетевая	SQL
6	FFS File	ПК	Файловая	Собственный
7	FoxBase+	ПК	Сетевая	Собственный
8	FoxPro	ПК	Сетевая	Собственный
9	IDMS	О	Сетевая	Собственный
10	IMS/VS	О	Иерархическая	CICS
11	INGRES	М, ПК	Реляционная	SOL, QUEL
12	ORACLE	О, М, ПК	Реляционная	SQL
13	Paradox	ПК	Реляционная	PAL, QBE
14	O&A	ПК	Файловая	Собственный
15	R:Base	ПК	Реляционная	SQL, собственный
16	Reflex	ПК	Файловая	Собственный
17	SQL/DS	О	Реляционная	SQL, QBE
18	TOTAL	О	Сетевая	DML
19	ПАЛЬМА	О, М, ПК	Реляционная	KРЕЗ
20	ИНТЕРБАЗА	ПК	Смешанная	SQL/IB, QBE/IB

В первую очередь, это относится к СУБД, созданным для класса ПК. Эти СУБД характеризуются не только своим количеством, но и функциональным разнообразием: от простых файловых систем до функционально полных СУБД, в основном *реляционного* типа. Многие из коммерческих СУБД поддерживают многопользовательскую работу и работу в *сетях* ЭВМ, как *локальных*, так и *глобальных*. К средствам, непосредственно относящимся к СУБД, можно отнести и многочисленные средства их окружения: генераторы и конверторы данных и программ, компиляторы языков программирования БД-приложений, генераторы создания различного назначения и уровня интерфейсов с БД в рамках традиционных ЯВУ и т.д.

Такое многообразие инструментальных и прикладных средств по СУБД позволяет выбирать наиболее адекватные нуждам пользователя, обеспечивая эффективное использование вычислительных ресурсов и существенное сокращение сроков разработки конкретных БД-технологий. В подавляющем большинстве СУБД для ПК ориентированы на интерактивный режим работы с пользователем, широко используя удобные и дружелюбные системы интерфейсов на основе простых и понятных меню. В СУБД, поддерживающих языки программирования БД-приложений, средства такого интерфейса избавляют пользователя от необходимости знания синтаксиса языка для обеспечения требуемых функций. Ряд популярных СУБД

предусматривают несколько уровней интерфейса, обеспечивающих работу с ними различной квалификации пользователей (*dBase IV, Paradox, R:Base* и др.). Большое внимание уделено эффективной системе Help-информации по СУБД, включающей электронные краткие обучающие *курсы* с демонстрацией наиболее часто используемых приемов работы с конкретным пакетом.

Интенсивное расширение компьютерной инфотехнологии ставит перед дальнейшим развитием СУБД целый ряд новых требований, во многом связанных с вопросами стандартизации. Это относится не только к СУБД, но и к ПС других типов. В отношении же СУБД это прежде всего относится к стандартизации *эталонной* модели управления данными, предусматривающей четкую классификацию основных вопросов стандартизации СУБД в зависимости от функциональных особенностей и уровня описания данных на разных стадиях проектирования. Можно предполагать, что *последующее* развитие СУБД будет ориентироваться на рекомендации международных стандартов относительно языков БД и средств доступа к удаленным БД, а также *интерфейсов* с системами программирования, ориентированными на ИИ-проблематику. Новые интересные аспекты БД-технологии появляются на основе объектно-ориентированной технологии программирования и обработки информации.

## Лекция 6 (2 часа)

*Объектно-ориентированные СУБД (ООСУБД)*

*Сложные объекты*

*Понятие идентичности объектов*

*Понятие инкапсуляции*

*Понятие класса и типа*

### **6. Объектно-ориентированные СУБД (ООСУБД)**

В настоящем параграфе рассматриваются основные концепции, понятия, черты и характеристики *объектно-ориентированных систем управления БД* (ООСУБД) в контексте объектно-ориентированных программирования и технологий. В последние годы в результате проникновения идеологии ООП в СУБД интенсивные разработки теоретического и прикладного характера ведутся по созданию различного назначения ООСУБД. Ввиду не совсем устоявшейся в этом направлении терминологии отметим основные черты и характеристики, определяющие СУБД как **объектно-ориентированную**. При этом по мере необходимости проводятся сопоставления с рассмотренной выше концепцией ООП.

Характеристики ООСУБД подразделяются на три *определяющие* группы:

- **базовые**, определяющие принадлежность СУБД к *объектно-ориентированному* классу;
- **по выбору**, позволяющие улучшать ООСУБД, но не являющиеся *базовыми*,
- **открытости**, позволяющие пользователю делать осознанный выбор из ряда одинаково приемлемых реализаций ООСУБД.

В первую очередь, ООСУБД должна удовлетворять двум критериям: быть СУБД в ее классическом понимании и быть *объектно-ориентированной системой* (ООС), т.е. в определенной степени она должна быть совместимой с современными объектно-ориентированными ЯВУ. **Первый** критерий включает следующие пять характеристик, присущих классической СУБД: сохранность и реентерабельность данных, развитое управление внешней памятью, возможность совмещения обработки и поиска данных, поддержка средств восстановления и возможность быстрого доступа к БД по запросу пользователя. Отмеченные характеристики в той или иной мере обсуждались выше. **Второй** критерий предполагает наличие следующих **восьми** характеристик, присущих собственно *объектно-ориентированной* технологии: понятие сложных объектов, идентичность объектов, инкапсуляция, типы или классы, наследование, настройка (сочетающаяся с отложенным присвоением), расширяемость и вычислительная полнота. Характеристики первого критерия хорошо известны пользователям традиционных СУБД (*INGRES*, *dBase*, *R:Base*, *IMS*, *Reflex* и др.), поэтому

кратко остановимся на характеристиках *второго* критерия, соотнося их с уже рассмотренными вопросами ООП-технологии.

**1. Сложные объекты** строятся из более простых путем применения к ним **конструкторов**. В качестве *простых* используются такие объекты, как: целые и действительные числа, символы, символьные строки любой длины, булевы величины и, возможно, другие первичные типы. В качестве **конструкторов** сложных объектов (*объектных конструкторов*) могут выступать: кортежи, множества, списки, массивы, таблицы и др. В качестве **минимального** набора объектных конструкторов для ООСУБД определяются: *множество*, *кортеж*, *список* или *массив*. **Множество** дает естественную *возможность* представления определенного набора объектов из имеющейся обширной совокупности; тогда как *кортеж* позволяет представлять определенные свойства объекта. При этом кортежи и множества имеют особое значение, получив *широкое* применение в качестве *объектных конструкторов* в *реляционных БД* (РБД). *Список* или *массив* играют важную роль при установлении *порядка* среди элементов множества. Более того, указанные *типы объектных конструкторов* играют важную роль во многих приложениях (векторно-матричные задачи, задачи анализа временных рядов и др.).

**Объектные конструкторы** должны удовлетворять **принципу ортогональности**: любой конструктор может применяться к любому объекту. Например, конструкторы РБД не обладают данным свойством (так конструктор *множества* может применяться только к *кортежам*, а конструктор *кортежей* - только к *первичным типам*). Наряду с этим для обеспечения работы со сложными объектами требуются также соответствующие **операторы**, которые оперируют с объектом как с единым целым. Таким образом, **операции** над сложным объектом должны распространяться *транзитивно* и на все его составляющие компоненты.

**2. Понятие идентичности объектов** давно существует в языках программирования и относительно недавно в **СУБД**. Суть данного понятия состоит в том, что существование объекта *не зависит* от его значения. В этом плане существует два понятия **эквивалентности** объектов:

(1) два объекта *идентичны* (*одинаковы*) или (2) *равны* (имеют одно и то же значение). Эти понятия влекут за собой соответственно две *импликации*: *разделение* объектов и их *обновление*. *Разделение* предполагает возможность разделения двумя объектами некоторой их *общей* компоненты, тогда как в случае *обновления* объектов *общие* компоненты двух объектов обновляются *независимо* при обновлении любого из них. **Идентичность** объектов является мощной *первой* операцией над данными, которая может быть положена в основу операций над множеством, кортежем сложных объектов, а также над рекурсивными сложными объектами.

Поддержка понятия *идентичности* объектов влечет за собой наличие таких операций, как: присваивание значений объекту, копирование объектов,

а также проверка на идентичность и/или равенство объектов. Следует отметить, что подход на основе идентичности присущ и современным ЯВУ, а именно: каждый программный объект идентичен в своей области определения и может быть обновлен. Данная *идентичность* определяется либо именем (*идентификатором*) объекта, либо его *местоположением* в памяти. Однако это понятие является совершенно новым для чисто РБД, в которых *отношения* базируются только на значениях объектов.

3. Понятие *инкапсуляции* связано с необходимостью как прояснить различие между *спецификацией* и *реализацией* операции, так и с применением *модульного* принципа организации ПС. На *инкапсуляцию* существует две точки зрения: (1) со стороны ЯВУ (именно здесь зародилось это понятие) и (2) адаптированная со стороны СУБД точка зрения. В *первом* случае понятие *инкапсуляции* восходит к абстрактным типам данных, когда *объект* имеет *интерфейсную* и *обрабатывающую* части. *Интерфейсная* часть специфицирует множество операций, которые могут быть выполнены над объектом. Тогда как *обрабатывающая* часть, в свою очередь, состоит из данных и процедур: *данные* определяют состояние или представление объекта, а *процедуры* на некотором ЯВУ (например *Turbo-Pascal*, *C++*, *SmallTalk* и др.) описывают применение к объекту каждой допустимой для него операции.

Распространение данного понятия на СУБД состоит в том, что объект *инкапсулирует* как *программу*, так и *данные*. Таким образом, предполагается *единая* модель для *данных* и их *обработки*, а сама информация может быть *скрытой* от пользователя. Операции, не специфицированные в *интерфейсной* части объекта, не могут выполняться. Данное ограничение распространяется на операции как *обновления*, так и *поиска* данных. *Инкапсуляция* в ООСУБД обеспечивает принцип *логической независимости* данных: изменение способа реализации типа не требует изменения самих прикладных программ, использующих данный тип. Следовательно, прикладные программы защищены от изменений способов реализации на нижних уровнях ООСУБД. *Полная инкапсуляция* состоит в том, что в объекте видимыми являются только допустимые над ним операции, а его *данные и реализации операций скрыты* внутри. Однако существует целый ряд случаев, когда *инкапсуляции* не требуется и использование ООСУБД может быть существенно упрощено, если система допускает при определенных условиях отмену инкапсуляции. Следовательно, механизм *инкапсуляции* в ООСУБД должен быть предусмотрен, но его использование должно быть достаточно гибким.

4. В ООС существуют две основные категории, поддерживающие соответственно понятие *класса* и *типа*. К *первой* категории можно отнести *SmallTalk* - подобные системы (*SmallTalk*, *Vision*, *GemStone* и др.) и все созданные на основе Zi'sp-языка системы (*Flavors*, *G-Base*, *Orion*, *Lore* и др.). Тогда как ко *второй* категории можно отнести такие системы, как: *C++*,

*Trellis, VBase, Simula* и др. Понятие **типа** в ООС суммирует общие черты множества объектов с одинаковыми характеристиками. Оно соответствует понятию типа абстрактных данных. Тип состоит из двух частей: *интерфейсной* и *обрабатывающей*. Для пользователя видна только **интерфейсная** часть, содержащая список допустимых для данного типа операций с их *сигнатурой* (типы входных параметров и результата). Тогда как **обрабатывающая** часть скрыта от пользователя и состоит, в свою очередь, также из двух частей: *данных* и *процедур*. Часть **данных** описывает собственно внутреннюю структуру данных объекта и в зависимости от возможностей ООС данная часть может быть более или менее сложной. **Процедуры** поддерживают выполнение операций из *интерфейсной* части объекта.

В ЯВУ **типы** являются эффективным средством повышения продуктивности программиста путем обеспечения корректности разрабатываемых ПС. В этом случае **типы** используются, главным образом, в период компиляции программы с целью проверки ее корректности, имеют *специальный* статус и не могут модифицироваться в период выполнения программ.

Понятие **класса** отличается от понятия **типа** большей динамичностью и включает следующие два аспекта: *генерация* объекта и *принадлежность* объектов. **Генерация** используется для создания новых объектов **класса** посредством операции *New* или соответствующей модификацией некоторого представителя данного класса. В свою очередь, **принадлежность** определяет вхождение того или иного объекта в данный класс. **Классы** не используются для проверки корректности программ, а предназначены для организации работы с объектами. В большинстве систем, базирующихся на этом подходе, **классы** играют первостепенную роль и могут динамически обрабатываться, обновляясь или передаваясь в качестве параметров. Естественно, устанавливаются строгие соотношения между **классами** и **типами** (так, для обоих понятий используются в качестве идентификаторов *имена*) и в некоторых ООС различия между ними могут быть едва уловимыми. В любом случае, ООСУБД должна включать некоторый механизм структурирования данных, базирующийся на **классах** или **типах**. Следовательно, классическое понятие структуры БД заменяется **понятием**, базирующимся на множестве **классов** или множестве **типов**.

5. **Наследование** является весьма существенной характеристикой ООСУБД, имеющей два основных преимущества: (1) предоставляет мощные средства *моделирования* (на основе компактного и точного описания среды) и (2) помогает в выработке разделяемых *спецификаций* и *реализации* в приложениях. **Наследование** существенно облегчает также создание многократно используемых программ. На сегодня существует, по крайней мере, четыре **типа** наследования: подстановка, включение, ограничение и специализация. **Наследование подстановки** означает, что тип А наследуется

из типа *B*, если можно производить больше операций над объектом *A*-типа, чем *B*-типа. Данный тип наследования отражает *поведенческие*, а не *семантические* аспекты объектов. *Наследование включения* соответствует понятию классификации и устанавливает тот факт, что *A*-тип является *подтипом* *B*-типа, если каждый объект *A*-типа является также объектом *B*-типа. Данный тип наследования отражает *структурные*, а не *функциональные* аспекты объектов. *Наследование ограничения* образует *подкласс* класса *наследования включения*, а именно: *A*-тип является *подтипом* *B*-типа, если он состоит из всех объектов *A*-типа, удовлетворяющих заданному ограничению. Наконец, *наследование специализации* заключается в том, что *A*-тип является *подтипом* *B*-типа, если объекты *A*-типа являются объектами и *B*-типа, содержащими более специфическую информацию. Следует отметить, что с различной степенью полноты указанные четыре типа *наследования* поддерживаются всеми существующими на сегодня ООСУБД.

6. В целом ряде случаев необходимо располагать одним *именем*, используемым для различных *реализации* одной и той же операции. Например, используется операция GS(*X*), которая в зависимости от типа *X*-объекта обеспечивает *различные* алгоритмы обработки. Естественно, что в момент компиляции программы тип *X*-объекта неизвестен. Поэтому в традиционной технологии программист в своей программе обеспечивает проверку на *тип X-объекта (переменной)* и в зависимости от ее результата осуществляет *ветвление* вычислительного алгоритма. В условиях ООСУБД такая операция GS(*X*) определяется на уровне *объектного* (наиболее общего в системе) типа; однако реализация операции переопределяется (*настраивается*) для каждого из типов обрабатываемых *X*-объектов. В этом случае, применяя операцию GS(*X*), мы предоставляем возможность самой системе динамически отыскивать (*настраивать*) подходящую реализацию операции согласно типа *X*-объекта. Для обеспечения этой новой черты ООС не может связывать имена операций с программами их реализации в период компиляции. Следовательно, присваивание имен реализациям должно осуществляться *динамически*, т.е. поддерживается принцип *отложенного* присвоения имен операционным программам. Следует отметить, что несмотря на то, что *отложенное* присвоение существенно усложняет проверку *типов* (а в ряде случаев делает ее просто невозможной), *полностью* оно такую проверку не исключает.

7. В основу построения ООСУБД положено множество *предопределенных* типов, используемых пользователем для написания прикладных программ. Данное множество типов должно быть *расширяемым* в том смысле, что ООСУБД располагает средствами определения *новых* (*пользовательских*) типов, рассматриваемых системой наравне с *предопределенными* (*системными*) типами. При этом свойство

*расширяемости* не обязательно должно распространяться на типы **конструкторов данных** (кортежи, множества, списки и т.д.).

С точки зрения языков программирования свойство **вычислительной полноты** ООСУБД является очевидным: любую вычислительную процедуру можно выразить средствами языка системы. Однако с точки зрения СУБД это не совсем так, ибо уже SQL-язык (предназначенный для описания структурированных **запросов** к БД) не обладает **вычислительной полнотой**. **Свойство вычислительной полноты** существенно повышает возможности ООСУБД, предназначенных только для *хранения и поиска* данных, а также поддержки простых вычислений над **первичными** значениями данных.

8. Согласно устоявшимся на сегодня понятиям каждая СУБД, наследующая рассмотренные выше черты и характеристики, относится к классу ООСУБД. Однако наряду с **определяющими** ООСУБД могут обладать и **дополнительными** (по выбору) характеристиками, позволяющими в том или ином отношении улучшать систему. Некоторые из них имеют объектно-ориентированную природу (например *множественное наследование*), другие - черты сугубо традиционных СУБД (например реализацию *транзакций*). *Первые* из них преследуют цель повышения уровня *объектно-ориентированности* системы, не являясь для нее **базовыми**. *Вторые*, как правило, преследуют цели улучшения функциональных и эксплуатационных характеристик системы с точки зрения классических СУБД и пользователя без акцента на ее *объектно-ориентированном* аспекте. Большинство таких характеристик преследуют цели улучшения обслуживания для новых приложений ООСУБД (CAD/CAM, CAP, CASE, управление, финансы, делопроизводство и др.) и носят скорее *прикладную*, чем *технологическую* направленность. Так как многие современные ООСУБД ориентированы именно на эти новые приложения, то возникает ряд противоречий между их дополнительными чертами *прикладной* ориентации и *объектно-ориентированной* природой самих систем.

9. Современные ООСУБД, представляющие собой *четвертое поколение* в технологии управления данными, в коммерчески пригодном исполнении появились относительно недавно. Они поддерживают все рассмотренные выше черты и характеристики как традиционных СУБД, так и *объектно-ориентированной технологии* (OOT). В плане выразительных возможностей прикладной семантики ООСУБД существенно мощнее традиционных *реляционных* СУБД. Превосходный сравнительный анализ *реляционных* СУБД и ООСУБД можно найти в [359-362, 394-398]. Первыми разработчиками ООСУБД в 1990 г. явились фирмы *Object Design*, *OBJECT-Sciences*, *Objectivity*, *OntoLogic* и *Servio* (США). Наряду с ними ряд разработчиков классических *реляционных* СУБД расширили свои системы *объектно-ориентированными* средствами. Среди наиболее удачных разработок можно отметить, в первую очередь, ООСУБД *VERSANT* фирмы *Versant Object Technology*, которая поддерживает ряд широко используемых

стандартов, а также концепцию ООСУБД, рассмотренную выше. В качестве примера кратко рассмотрим вопросы VERSANT-технологии управления данными, базирующейся на ООСУБД VERSANT, на сегодня являющейся одним из стандартов *дефакто* для такого типа ООС.

10. На основе технологии разработан достаточно широкий спектр высокопроизводительных *объектно-ориентированных* ПС, позволяющих создавать гибкие *интегрированные информационные системы* (ИИС), включая в них уже существующие СУБД пользователя. Дополнительно к ООСУБД-технологии VERSANT-технология в целом поддерживает средства интегрирования однородных БД, набор языковых интерфейсов, наборы средств для администратора БД, системного и прикладного программистов, а также специальные средства, обеспечивающие нужды пользователя. Наряду с ООТ VERSANT-технология позволяет создавать пользовательские ИИС более производительными, гибкими и реактивными. При этом обеспечивается интегрирование существующих пользовательских РБД и ООСУБД VERSANT в единую *унифицированную* информационную систему; VERSANT-технология для любой БД системы допускает режим *клиент-сервер*, а для ООСУБД - *многопользовательский* и *многосерверный* режимы. Следовательно, информация такой системы может разделяться на весьма широкой основе пользователями, получающими доступ к любому количеству однородных распределенных БД, функционирующих на широком спектре программных и технических средств. При этом все различия между компонентами системы невидимы и несущественны (*инкапсулированы*) для конечного пользователя - относительно конечного пользователя такая ИИС представляется как единое целое.

Использование VERSANT-технологии позволяет повышать производительность прикладных программистов на порядок и более, а производительность самих ИИС почти на два порядка. Полученный опыт подтверждает, что современные ООСУБД обладают примерно на два порядка большей производительностью, чем *реляционные* СУБД при поиске сложной информации. Например, при использовании самой быстрой *реляционной* СУБД для поиска описания некоторого объекта, содержащего несколько тысяч элементов, потребовалось порядка 15 мин, тогда как ООСУБД VERSANT на эту же задачу потребовалось менее 10 с. На **рисунке 5.6** представлена общая 8-уровневая архитектура ИИС VERSANT, начиная с ООСУБД и кончая конечным пользователем.

На *первом* уровне находится собственно ООСУБД VERSANT, являющаяся на сегодня одной из наиболее развитых и мощных систем данного типа ООС и поддерживающая во всей полноте *определяющие* и *дополнительные* черты и характеристики, рассмотренные выше. Наряду с этим VERSANT поддерживает многопользовательский/многосерверный режим управления *распределенными* БД, позволяя сохранять, отыскивать и обновлять объекты в распределенных вычислительных системах.

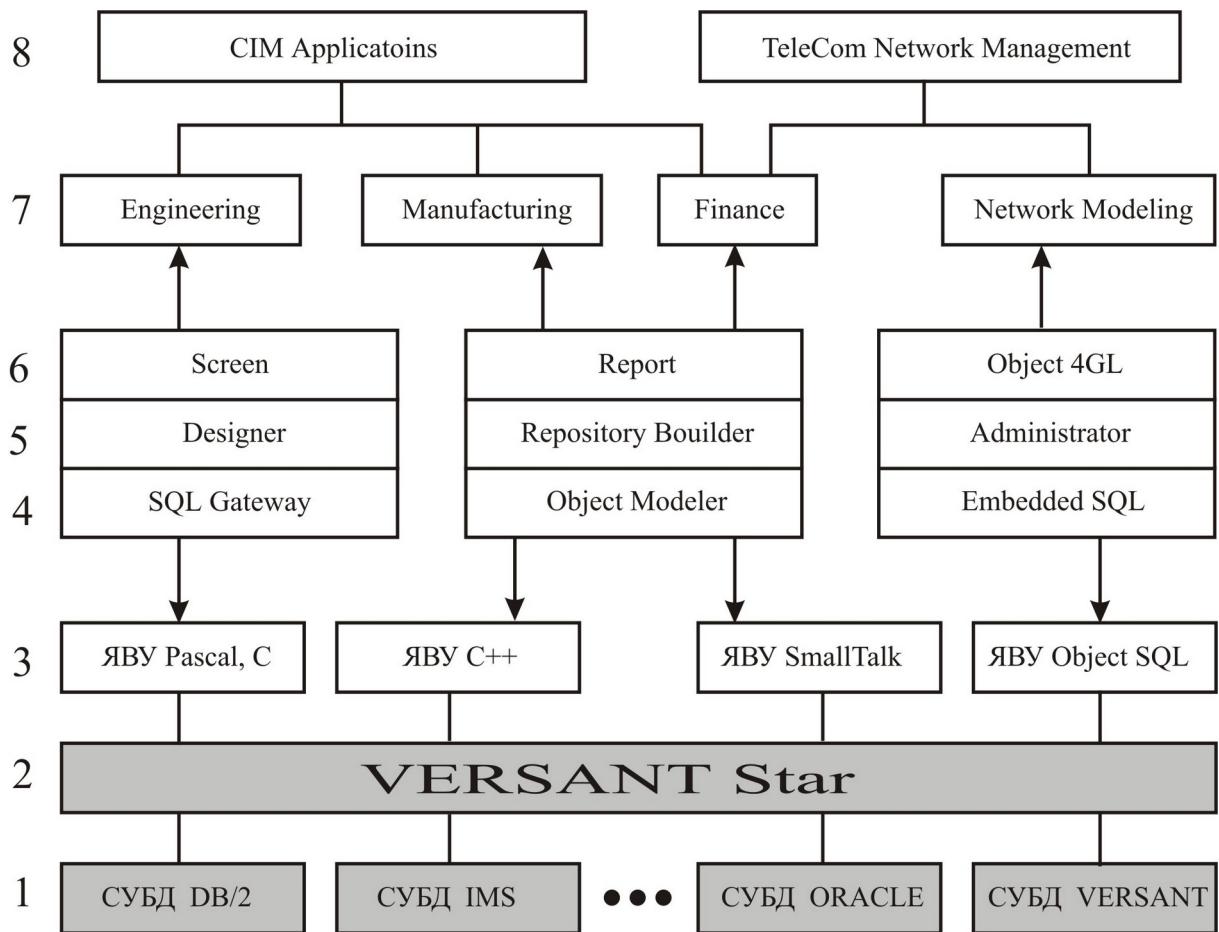


Рисунок 5.6 – Архитектура ИИС на основе VERSANT-технологии

**Второй** уровень содержит единственную компоненту VERSANT Star системы, осуществляющую интеграцию различных СУБД. Она управляет доступом к **объектам**, находящимся в системе БД *первого* уровня архитектуры. С этой целью она использует набор БД-драйверов, поддерживающих индивидуальные БД-форматы. По этой причине в настоящее время поддерживаются **только** наиболее популярные СУБД (*ORACLE, DB/2, IMS* и др.), но данный список постоянно расширяется. Этому способствует и переход к новому единому ASCII-стандарту для *реляционных* СУБД.

В отличие от некоторых ООСУБД система VERSANT не использует *единого* языка программирования и не требует от пользователя использования какого-либо конкретного компилятора или транслятора. Она позволяет для работы со своими БД использовать многие как *традиционные*, так и *объектно-ориентированные ЯВУ* (*третий* уровень; смотри рисунок 5.6). Достигается это посредством введения *языковых интерфейсов* в виде библиотек *классов* и *функций*. Посредством использования данных библиотек в своих прикладных программах пользователь получает доступ ко *всем* ресурсам VERSANT-системы на широком спектре различных ЯВУ. В настоящее время поддерживаются пять *базовых* языков: *Pascal, C, C++,*

*SmallTalk* и *Object SQL*. Последний является расширением SQL-стандарта, позволяющим управлять структурами объектно-ориентированных данных. В свою очередь, С-интерфейс может быть использован любым языком программирования, допускающим вызовы внешних С-функций (*Pascal*, *Fortran*, *Cohol* и др.). Это позволяет пользователю писать свои прикладные программы на подходящем ЯВУ, обеспечивая доступ к любой БД, интегрированной в *VERSANT*-систему. Следующие три уровня архитектуры определяют (3x3)-матрицу средств, предназначенных для облегчения разработки прикладных программ и администрирования интегрированной ООСУБД *VERSANT*-системы.

**Четвертый** уровень содержит средства, расширяющие возможности прикладных программ, написанных на ЯВУ уровня 3 и совместимых с ними. Компонента этого уровня *Object Modeler* является наиболее многоцелевым средством, позволяющим средствами графического интерфейса на экране определять объектно-атрибутные структуры данных. Затем эти структуры данных могут автоматически транслироваться в С-структуры, {C+*SmallTalk*} -классы или SQL-таблицы, позволяя существенно сокращать время разработки прикладных программ.

Средство *Object Modeler* оказывает неоценимую помощь в деле повышения продуктивности пользователя, так как позволяет только один раз определять структуры данных, а затем повторно использовать их во многих различных приложениях безотносительно **языка**, на котором они были запрограммированы, или БД, к которым организуется доступ. **Вторым** важным преимуществом данного средства является то, что оно может непосредственно использоваться системотехниками, воспитанными на традиционной технологии, а не на ООТ. Два других средства уровня 4 обеспечивают поддержку языка запросов SQL-стандарта. Так, на базе средства *SQL Gateway* (*шлюза*) пользователь в своих программах на языках C++ и/или *SmallTalk* может организовывать запросы в SQL-стандарте и автоматически получать результаты запросов в соответствующие **объекты** программы. Средство же *Embedded SQL* позволяет использовать предложения языка *Object SQL* для расширенного доступа к объектно-ориентированным структурам данных.

**Пятый** уровень **архитектуры** (смотри рисунок 5.6) включает средства для администратора БД, предназначенные для настройки и управления **распределенными** БД. Данные средства пригодны также и для прикладных программистов, но только *первое* из них обычно ими используется. Средство *Designer* представляет собой графический инструмент для программистов и администратора БД, позволяющий в рамках VERSANT-баз создавать и управлять *иерархиями классов*. Средство автоматически генерирует *определения* классов для языка C++ из их графического представления. Более того. *Designer* может просматривать БД, позволяя *пользователю* или *администратору* проверять (*просматривать*) *определения* классов в БД,

включая их атрибуты, отношения и методы. С его помощью можно проверять, изменять или удалять (т.е. *редактировать*) отдельные элементы в БД. Средство *Repository Builder* позволяет *администратору* БД определять и обслуживать *метаданные* из архива данных системы. По функционированию данное средство несколько напоминает предыдущее, но включает ряд специальных возможностей для обеспечения доступа именно к информации *архива* системы. Наконец, *Administrator* представляет собой набор средств *администратора* БД, поддерживающих: командный (*оперативный*) режим управления пользователями и доступом к БД, операции восстановления, а также выполнять другие стандартные сервисные функции по обслуживанию ООСУБД VERSANT-системы.

*Шестой* уровень включает средства, предназначенные для быстрой генерации мощных объектно-ориентированных прикладных программ на основе библиотек классов, поддерживаемых системой. Графическое *Screen*-средство ориентировано на разработку необходимых *графических* интерфейсов для приложений. Наряду с этим *Screen* имеет развитые средства для управления *широким* набором носителей информации. Например, можно вводить диаграммы, производить поиск в БД фотографий, карт или осуществлять другие процедуры (не поддерживаемые традиционными пакетами) по работе с *форматированной* информацией. Средство *Report* предназначено для подготовки отчетов и при необходимости может выполнять роль традиционного *генератора* отчетов. Наконец, *Object 4GL* представляет собой ЯВУ программирования БД, который может быть использован для сопоставки операций и объектов, созданных другими средствами системы. Следует отметить, что графические интерфейсы, отчеты и процедуры, созданные средствами *шестого* уровня, сохраняются в архиве системы (*Repository*) и становятся доступными для всех приложений в среде VERSANT-системы. Более того, по причине определения этих прикладных конструкций в терминах абстрактных категорий и атрибутов, они могут использоваться и с другими типами СУБД.

*Седьмой* уровень архитектуры (смотри рисунок 5.6) представляют *специализированные* приложения (*библиотеки классов*), специально адаптированные к нуждам той или иной области приложений VERSANT-системы (проектирование, конструирование, производство, строительство, сетевое моделирование и др.). Средства данного уровня стали возможными благодаря принципу *многократного использования*, поддерживаемому ООТ.

Наконец, на *восьмом* уровне *архитектуры* представлены фактические (*конечные*) приложения VERSANT-технологии. Используя средства *седьмого* уровня или создавая свои собственные, пользователь завершает создание конкретного *функционально полного* приложения VERSANT-системы. На рисунке 5.6 условно показано, как *некоторый* интегрированный пакет для проектирования в области разработки и производства интегральных схем (*CIM Applications*) может быть перенастроен на основе средств *седьмого*

уровня (*Engineering, Manufacturing, Finance*), используя их функциональные возможности, чтобы быстро связывать изменения проектных решений с производственными затратами и оценивать их влияние на себестоимость конечного изделия. Практическое использование VERSANT-технологии позволяет говорить о ее больших *потенциях* для разработчиков АСУП и АСУТП всех уровней.

Первоначально средства VERSANT-технологии были разработаны и эксплуатировались на платформах *Sun 3/4*, мини-ЭВМ *IBM RISC System/6000*, *DEC*-станциях, *HP 9000/400*, системах *Silicon Graphics*, *InterGraph 6000* и последующих сериях мультипроцессорных ЭВМ. В настоящее время указанные **ООС** функционируют также на платформах *Unix*, *Windows*, *OS/2* и *Macintosh*. Эти реализации делают VERSANT-технологию доступной для широкого круга пользователей IBM-совместимых ПК и существенно расширяют ее возможности для обеспечения распределенной обработки информации в условиях функционирования различного назначения ИИС.

## Лекция 7 (2 часа)

*Базы знаний и экспертные системы*

*Базы знаний и способы представления*

*Модель на базе логики*

*Фреймы, семантические сети*

*Экспертные системы*

*Преимущества использования экспертных систем*

*Особенности построения и организации экспертных систем*

*Основные режимы работы экспертных систем*

### **7.1. Базы знаний и способы представления**

*База знаний (knowledge base)* - совокупность знаний, относящихся к некоторой предметной области и формально представленных таким образом, чтобы на их основе можно было осуществлять рассуждения.

Базы знаний чаще всего используются в контексте *экспертных систем*, где с их помощью представляются *навыки и опыт экспертов*, занятых практической деятельностью в соответствующей области (например, в медицине или в математике). Обычно база знаний представляет собой совокупность правил вывода.

При проектировании модели представления знаний следует учитывать такие факторы, как:

- однородность представления и
- простота понимания.

Однородность представления приводит к упрощению механизма управления логическим выводом и управлением знаниями.

Простота понимания предполагает доступность понимания представления знаний и экспертам, и пользователем системы. В противном случае затрудняется приобретение знаний и их оценка.

Однако выполнить эти требования в равной степени, как для простых, так и сложных задач довольно трудно. В настоящее время для представления знаний используют следующие виды моделей:

- модель на базе логики;
- продукционная модель;
- модель семантической сети;
- модель, основанная на использовании фреймов и др.

### **7.2 Модель на базе логики**

Основная идея логического подхода состоит в том, чтобы рассматривать всю систему знаний необходимую для решения прикладных задач, как совокупность фактов (утверждений).

Факты представляются как формулы в некоторой логике (первого или высшего порядка, многозначной, нечеткой или др.) Система знаний отображается совокупностью таких формул и, представленная в ЭВМ, она образует БЗ.

Формулы неделимы и при модификации БЗ могут лишь добавляться или удаляться.

Логические методы обеспечивают развитый аппарат вывода новых фактов из тех, которые явно представлены в БЗ.

Основным примитивом манипуляции знаниями является операция вывода.

### Продукционная модель

Это один из наиболее часто используемых в экспертных системах способов представления знаний. Основная идея заключается в ассоциировании с соответствующими действиями набора условий в виде правил типа "если-то", называемых также продукциями:

ЕСЛИ условие ТО действия

"Если-то"-правила обычно оказываются весьма естественным выразительным средством представления знаний. Кроме того, они обладают следующими привлекательными свойствами:

- модульность: каждое правило описывает небольшой, относительно независимый фрагмент знаний;
- возможность инкрементного наращивания: добавление новых правил в базу знаний происходит относительно независимо от других правил;
- удобство модификации (как следствие модульности): старые правила можно изменять и заменять на новые относительно независимо от других правил;
- применение правил способствует прозрачности системы, т.е. способности к объяснению принятых решений и полученных результатов.

Однако продукционные системы не свободны от недостатков:

- процесс вывода менее эффективен, чем в других системах, поскольку большая часть времени при выводе затрачивается на непроизводительную проверку применимости правил;
- этот процесс трудно поддается управлению;
- сложно представить родовидовую иерархию понятий.

### Фреймы

Представление знаний, основанное на фреймах, является альтернативным по отношению к системам продукции: оно дает возможность хранить родовидовую иерархию понятий в базе знаний в явной форме. Фреймом называется структура для описания стереотипной ситуации, состоящая из характеристик этой ситуации и их значений, характеристики называются слотами, а значения - заполнителями слотов. Слот может содержать не только конкретное значение, но и имя процедуры, позволяющей вычислить его по заданному алгоритму, а также одну или несколько продукции (эвристик), с помощью которых это значение можно

найти. В слот может входить не одно, а несколько значений. Иногда слот включает компонент, называемый фасетом, который задает диапазон или перечень его возможных значений. Фасет указывает также граничные значения заполнителя слота.

Совокупность фреймов, моделирующая какую-нибудь предметную область, представляет собой иерархическую структуру, в которую фреймы соединяются с помощью родовидовых связей. На верхнем уровне иерархии находится фрейм, содержащий наиболее полную информацию, истинную для всех остальных фреймов. Фреймы обладают способностью наследовать значения характеристик своих родителей, находящихся на более высоком уровне иерархии. Значения характеристик фреймов могут передаваться по умолчанию фреймам, находящимся ниже них в иерархии, но если последние содержат собственные значения данных характеристик, то в качестве истинных принимаются именно они.

Наиболее ярко достоинства фреймовых систем представления знаний проявляются в том случае, если родовидовые связи изменяются нечасто и предметная область насчитывает немного исключений. Во фреймовых системах данные о родовидовых связях хранятся явно, т.е. так же, как и значения всех других типов. Значения слотов представляются в системе в единственном экземпляре, поскольку включаются только в один фрейм, описывающий наиболее общее понятие из всех тех, которые содержат слот с данным именем. Такое свойство систем фреймов дает возможность уменьшить объем памяти, необходимый для их размещения в компьютере. Еще одно достоинство фреймов состоит в том, что значение любого слота при необходимости может быть вычислено с помощью соответствующих процедур или найдено эвристическими методами.

Как недостаток фреймовых систем следует отметить их относительно высокую сложность, что проявляется в снижении скорости работы механизма вывода и в увеличении трудоемкости внесения изменений в родовидовую иерархию. Кроме того, во фреймовых системах затруднена обработка исключений.

### Семантические сети

Семантическая сеть, по мнению специалистов, - наиболее общий способ представления знаний, причем она появилась, по-видимому, ранее других. В ней понятия и классы, а также отношения и связи между ними представлены в виде сети. Семантическая сеть отображает совокупность объектов предметной области и отношений между ними, при этом объектам соответствуют вершины (или узлы) сети, а отношениям - соединяющие их дуги. В качестве объектов могут выступать события, действия, обобщенные понятия или свойства объектов. Свойства представляются в сети также в виде вершин и служат для описания классов объектов. Вершины сети

соединяются дугой, если соответствующие объекты предметной области находятся в каком-либо отношении.

Как и в системе, основанной на фреймах, в семантической сети могут быть представлены родовидовые отношения, которые позволяют реализовать наследование свойств от объектов-родителей. Это обстоятельство приводит к тому, что семантические сети приобретают большинство недостатков и достоинств представления знаний в виде фреймов. Но основное преимущество семантических сетей заключается в их наглядности и непосредственной связанности понятий через сеть, которая позволяет быстро находить связи понятий и на этой основе управлять принимаемыми решениями. Именно этот формализм был использован в данной работе в качестве формализма для представления знаний в БЗ. Основной недостаток сетей - сложность обработки исключений.

### **7.3 Экспертные системы**

В начале 80-х годов в исследованиях по искусственному интеллекту сформировалось самостоятельное направление, получившее название "экспертные системы" (ЭС). Основным назначением ЭС является разработка программных средств, которые при решении задач, трудных для человека, получают результаты, не уступающие по качеству и эффективности решения, решениям получаемым человеком-экспертом. ЭС используются для решения так называемых неформализованных задач, общим для которых является то, что:

- задачи не могут быть заданы в числовой форме;
- цели нельзя выразить в терминах точно определённой целевой функции;
- не существует алгоритмического решения задачи;
- если алгоритмическое решение есть, то его нельзя использовать из-за
  - ограниченности ресурсов (время, память).

Кроме того неформализованные задачи обладают ошибочностью, неполнотой, неоднозначностью и противоречивостью как исходных данных, так и знаний о решаемой задаче.

**Экспертная система** - это программное средство использующее экспертные знания для обеспечения высокоэффективного решения неформализованных задач в узкой предметной области. Основу ЭС составляет база знаний (БЗ) о предметной области, которая накапливается в процессе построения и эксплуатации ЭС. Накопление и организация знаний - важнейшее свойство всех ЭС.



*Рисунок 7.1 – Основные свойства*

Экспертные системы должны:

- хранить знания об определенной предметной области (факты, описания событий и закономерностей);
- уметь общаться с пользователем на ограниченном естественном языке (т.е. задавать вопросы и понимать ответы);
- обладать комплексом логических средств для выведения новых знаний, выявления закономерностей, обнаружения противоречий;
- ставить задачу по запросу, уточнять её постановку и находить решение;
- объяснять пользователю, каким образом получено решение.

Желательно также, чтобы экспертная система могла:

- сообщать такую информацию, которая повышает доверие пользователя к экспертной системе;
- «рассказывать» о себе, о своей собственной структуре.

Экспертные системы могут использоваться в различных областях - медицинской диагностике, при поиске неисправностей, разведке полезных ископаемых, выборе архитектуры компьютерной системы и т.д.

Знания являются явными и доступными, что отличает ЭС от традиционных программ, и определяет их основные свойства, такие, как:

1) Применение для решения проблем высококачественного опыта, который представляет уровень мышления наиболее квалифицированных экспертов в данной области, что ведёт к решениям творческим, точным и эффективным.

2) Наличие прогностических возможностей, при которых ЭС выдаёт ответы не только для конкретной ситуации, но и показывает, как изменяются эти ответы в новых ситуациях, с возможностью подробного объяснения каким образом новая ситуация привела к изменениям.

3) Обеспечение такого нового качества, как институциональная память, за счёт входящей в состав ЭС базы знаний, которая разработана в ходе взаимодействий со специалистами организации, и представляет собой текущую политику этой группы людей. Этот набор знаний становится сводом квалифицированных мнений и постоянно обновляемым

справочником наилучших стратегий и методов, используемых персоналом. Ведущие специалисты уходят, но их опыт остаётся.

**4)** Возможность использования ЭС для обучения и тренировки руководящих работников, обеспечивая новых служащих обширным багажом опыта и стратегий, по которым можно изучать рекомендуемую политику и методы.

Состав и взаимодействие участников построения и эксплуатации экспертных систем.

К числу основных участников следует отнести саму экспертную систему, экспертов, инженеров знаний, средства построения ЭС и пользователей.

**Экспертная система** это программное средство, использующее знания экспертов, для высокоэффективного решения задач в интересующей пользователя предметной области. Она называется системой, а не просто программой, так как содержит базу знаний, решатель проблемы и компоненту поддержки. Последняя из них помогает пользователю взаимодействовать с основной программой.

**Эксперт** - это человек, способный ясно выражать свои мысли и пользующийся репутацией специалиста, умеющего находить правильные решения проблем в конкретной предметной области. Эксперт использует свои приёмы и ухищрения, чтобы сделать поиск решения более эффективным, и ЭС моделирует все его стратегии.

**Инженер знаний** - человек, как правило, имеющий познания в информатике и искусственном интеллекте и знающий, как надо строить ЭС. Инженер знаний опрашивает экспертов, организует знания, решает, каким образом они должны быть представлены в ЭС, и может помочь программисту в написании программ.

**Средство построения ЭС** это программное средство, используемое инженером знаний или программистом для построения ЭС. Этот инструмент отличается от обычных языков программирования тем, что обеспечивает удобные способы представления сложных высокогоуровневых понятий.

**Пользователь** - это человек, который использует уже построенную ЭС. Так, пользователем может быть юрист, использующий её для квалификации конкретного случая; студент, которому ЭС помогает изучать информатику и т. д. Термин пользователь несколько неоднозначен. Обычно он обозначает конечного пользователя. Пользователем может быть:

- создатель инструмента, отлаживающий средство построения ЭС;
- инженер знаний, уточняющий существующие в ЭС знания;
- эксперт, добавляющий в систему новые знания;
- клерк, заносящий в систему текущую информацию.

Важно различать инструмент, который используется для построения ЭС, и саму ЭС. Инструмент построения ЭС включает как язык,

используемый для доступа к знаниям, содержащимся в системе, и их представления, так и поддерживающие средства программы, которые помогают пользователям взаимодействовать с компонентой экспертной системы, решающей проблему.

### ***Преимущества использования экспертных систем***

Возникает вопрос: "Зачем разрабатывать экспертные системы? И не лучше ли обратиться к человеческому опыту, как это было в прошлом?". Отметим лишь основные преимущества, которые даёт использование ЭС. Преимуществами и положительными качествами искусственной компетенции являются:

1) Её постоянство. Человеческая компетенция ослабевает со временем. Перерыв в деятельности человека-эксперта может серьёзно отразиться на его профессиональных качествах.

2) Лёгкость передачи или воспроизведения. Передача знаний от одного человека другому - долгий и дорогой процесс. Передача искусственной информации - это простой процесс копирования программы или файла данных.

3) Устойчивость и воспроизводимость результатов. Эксперт-человек может принимать в тождественных ситуациях разные решения из-за эмоциональных факторов. Результаты ЭС - стабильны.

3) Стоимость. Эксперты, особенно высококвалифицированные обходятся очень

4) дорого. ЭС, наоборот, сравнительно недороги. Их разработка дорога, но они дёшевы в эксплуатации.

Вместе с тем разработка ЭС не позволяет полностью отказаться от эксперта-человека. Хотя ЭС хорошо справляется со своей работой, тем не менее в определённых областях человеческая компетенция явно превосходит искусственную. Однако и в этих случаях ЭС может позволить отказаться от услуг высококвалифицированного эксперта, оставив эксперта средней квалификации, используя при этом ЭС для усиления и расширения его профессиональных возможностей.

### ***7.4 Особенности построения и организации экспертных систем***

Основой любой ЭС является совокупность знаний, структурированная в целях упрощения процесса принятия решения. Для специалистов в области искусственного интеллекта термин знания означает информацию, которая необходима программе, чтобы она вела себя "интеллектуально". Эта информация принимает форму фактов и правил. Факты и правила в ЭС не всегда либо истинны, либо ложные. Иногда существует некоторая степень неуверенности в достоверности факта или точности правила. Если это сомнение выражено явно, то оно называется "коэффициентом доверия".

Коэффициент доверия - это число, которое означает вероятность или степень уверенности, с которой можно считать данный факт или правило достоверным или справедливым.

Многие правила ЭС являются эвристиками, то есть эмпирическими правилами или упрощениями, которые эффективно ограничивают поиск решения. ЭС используют эвристики, так как задачи, которые она решает, трудны, не до конца понятны, не поддаются строгому математическому анализу или алгоритмическому решению. Алгоритмический метод гарантирует корректное или оптимальное решение задачи, тогда как эвристический метод даёт приемлемое решение в большинстве случаев.

Знания в ЭС организованы так, чтобы знания о предметной области отделить от других типов знаний системы, таких как общие знания о том, как решать задачи или знание о том, как взаимодействовать с пользователем. Выделенные знания о предметной области называются базой знаний, тогда как общие знания о нахождении решений задач называются механизмом вывода. Программные средства, которые работают со знаниями, организованными таким образом, называются системами, основанными на знаниях.

БЗ содержит факты (данные) и правила (или другие представления знаний), использующие эти факты как основу для принятия решений. Механизм вывода содержит:

- интерпретатор, определяющий как применять правила для вывода новых знаний на основе информации, хранящейся в БЗ;
- диспетчер, устанавливающий порядок применения этих правил.

Такие ЭС получили название статических ЭС, они не учитывают изменения окружающего мира за время решения задачи.

Однако существует более высокий класс приложений, где требуется учитывать динамику изменения окружающего мира за время исполнения приложения. Такие экспертные системы получили название динамических ЭС.

## **7.5. Основные режимы работы экспертных систем**

В работе ЭС можно выделить два основных режима: *режим приобретения знаний* и *режим решения задачи* (режим консультации или режим использования). В режиме приобретения знаний общение с ЭС осуществляется экспертом (при помощи инженера знаний).

Используя компонент приобретения знаний, эксперт описывает проблемную область в виде совокупности фактов и правил. Другими словами, "наполняет" ЭС знаниями, которые позволяют ей самостоятельно решать задачи из проблемной области.

Отметим, что этому режиму при традиционном подходе к программированию соответствуют этапы: алгоритмизации, программирования и отладки, выполняемые программистом. Таким образом,

в отличие от традиционного подхода в случае ЭС разработку программ осуществляет не программист, а эксперт, не владеющий программированием.

В режиме консультаций общение с ЭС осуществляется конечный пользователь, которого интересует результат и (или) способ его получения. Необходимо отметить, что в зависимости от назначения ЭС пользователь может:

- не быть специалистом в данной предметной области, и в этом случае он обращается к ЭС за результатом, который не умеет получить сам;
- быть специалистом, и в этом случае он обращается к ЭС с целью ускорения получения результата, возлагая на ЭС рутинную работу.

Следует отметить, что в отличие от традиционных программ ЭС при решении задачи не только исполняют предписанную алгоритмом последовательность операций, но и сама предварительно формирует её.

Хорошо построенная ЭС имеет возможность самообучаться на решаемых задачах, пополняя автоматически свою БЗ результатами полученных выводов и решений.

### *Отличие экспертивных систем от традиционных программ*

Особенности ЭС, отличающие их от обычных программ, заключаются в том, что они должны обладать:

1. Компетентностью, а именно:
  - Достигать экспертивного уровня решений (т.е. в конкретной предметной области иметь тот же уровень профессионализма, что и эксперты-люди).
  - Быть умелой (т.е. применять знания эффективно и быстро, избегая, как и люди, ненужных вычислений).
  - Иметь адекватную робастность (т.е. способность лишь постепенно снижать качество работы по мере приближения к границам диапазона компетентности или допустимой надёжности данных).
2. Возможностью к символическим рассуждениям, а именно:
  - Представлять знания в символическом виде
  - Переформулировать символические знания. На жаргоне искусственного интеллекта символ это строка знаков, соответствующая содержанию некоторого понятия. Символы объединяют, чтобы выразить отношения между ними. Когда отношения представлены в ЭС они называются символыми структурами.
3. Глубиной, а именно:
  - Работать в предметной области, содержащей трудные задачи
  - Использовать сложные правила (т.е. использовать либо сложные конструкции правил, либо большое их количество)
4. Самосознанием, а именно:
  - Исследовать свои рассуждения (т.е. проверять их правильность)
  - Объяснять свои действия

Существует ещё одно важное отличие ЭС. Если обычные программы разрабатываются так, чтобы каждый раз порождать правильный результат, то ЭС разработаны с тем, чтобы вести себя как эксперты. Они, как правило, дают правильные ответы, но иногда, как и люди, способны ошибаться.

Традиционные программы для решения сложных задач, тоже могут делать ошибки. Но их очень трудно исправить, поскольку алгоритмы, лежащие в их основе, явно в них не сформулированы. Следовательно, ошибки нелегко найти и исправить. ЭС, подобно людям, имеют потенциальную возможность учиться на своих ошибках.

## Лекция 8 (2 часа)

- Компьютерная телекоммуникация*
- Модемная компьютерная телекоммуникация*
- Программное обеспечение модемной телекоммуникации*
- Локальные и глобальные информационно-вычислительные сети*
- Принципы организации информационно-вычислительных сетей*
- Архитектура ИВС*
- Глобальная сеть Internet. Общая характеристика глобальной сети Internet*
- Протоколы сети Internet*
- Адрес компьютера*

## **8 КОМПЬЮТЕРНАЯ ТЕЛЕКОММУНИКАЦИЯ.**

Одной из основных *составляющих* и *определяющих* прогресса человеческого общества является уровень его *коммуникабельности*. Современная коммуникабельность общества все более принимает компьютерный вид, основу которого составляет **компьютерная телекоммуникация**.

Компьютерная телекоммуникация позволяет в рамках указанных технических средств оперативно (вплоть до режима реального времени) обмениваться информацией (практически любого типа) абонентам, имеющим ЭВМ. При этом ЭВМ должны быть оборудованы специальными техническими средствами и снабжены соответствующим ПО и подключены к линиям связи того или иного типа. В рамках компьютерной телекоммуникации пользователь ЭВМ получает возможность: организовывать электронную почту, получать доступ к удаленным базам данных и знаний (БД/БЗ), разделять вычислительные ресурсы в рамках локальных (ЛВС) и/или глобальных (ГВС) вычислительных сетей, участвовать в теледискуссиях, производить через свой ПК банковские операции и многое другое.

Компьютерная телекоммуникация имеет три основных уровня: модемная, ЛВС и ГВС.

### **8.1 Модемная компьютерная телекоммуникация**

Характерной чертой современного использования ПК является организация на их основе информационного обмена по каналам связи. Этому способствуют не только быстрый рост парка различного типа ПК, появление доступных технических средств (модемы, факсы )

**Организация связи.** Уровни развития и возможности ПС данной группы различны и нами выделяются два основных вида связи: двух

*локальных ПК и модемная связь* ПК с *удаленным* абонентом, в качестве которого может выступать, в частности, другой ПК или факс. К *первому* уровню можно отнести ПС, поддерживающие простую связь двух ПК, соединенных через *последовательные* порты соединительным кабелем.

Наиболее распространенным и доступным видом связи является *телефонная*, позволяющая осуществлять связь по *коммутуируемым* или *выделенным* каналам. Так как в телефонной линии связи используется передача *аналоговой* информации, а ЭВМ работает с *дискретной* (цифровой), то для обеспечения *интерфейса* обоих типов информации используются специальные *устройства - модемы*, осуществляющие *модуляцию дискретного* сигнала в *аналоговый* и обратную операцию *демодуляции*. *Модем* (*факс-модем*) представляет собой *устройство* в составе аппаратуры передачи данных, осуществляющее функции *преобразования* потока битов в *аналоговые* сигналы, пригодные для передачи по некоторому *аналоговому* каналу связи (телефонному, телеграфному, кабельному, радио, спутниковому, световоду и т.д.), и наоборот. Большинство *модемов* разрабатывается в соответствии со спецификой национальных и международных стандартов, обеспечивая совместимость устройств и каналов связи. *Модемы* относительно компоновки с ЭВМ могут быть как *встроенные*, так и *внешние*, но в любом случае они являются *посредниками* между ЭВМ и телефонной линией связи.

В настоящее время именно *модемы* являются наиболее широко используемыми аппаратными средствами для обеспечения соединения ПК по каналам связи с удаленными абонентами (ЛВС, ГВС, другая ЭВМ, факс и т.д.). При выборе типа модема для своего ПК следует учитывать их распространенность, совместимость с другими типами модемов, соответствие используемых линий связи и возможности сопутствующего ему ПО.

При передаче информации через *модем* используются *два* способа: *асинхронный* и *синхронный*, позволяющие выделять для *декодирования* составляющие ее символы и производить контроль корректности передачи. Так как каждый символ имеет *байт-битовое* представление, то передача символов производится побито один за другим.

При *асинхронной* передаче каждый передаваемый символ представляется бинарным *пакетом* длиной 11 бит: первый бит пакета является *начальным* (вб), за ним следуют 8 бит бинарного кода передаваемого символа, затем *бит контроля* (бк) на четность/нечетность и завершает пакет *конечный бит* (кб). В случае схемы проверки на *нечетность* значения битов *нб*, бк и кб выбираются такими, чтобы общее число единичных битов в пакете было *нечетным*. При *синхронной* передаче символы передаются *блоками*, каждый из которых открывается *двумя начальными синхросимволами* (еe), за ними следуют 8-битные коды некоторого количества передаваемых символов, завершающихся *контрольными битами* (КоБ) и

*двумя конечными синхросимволами.* Так как реализация *асинхронной* схемы передачи/приема информации достаточно проста и дешева, то она используется весьма широко. Вместе с тем она относительно *медленная*, ибо каждый передаваемый символ *нагружается* тремя сопутствующими битами (**нб**, **бк**, **кб**), т.е. избыточность составляет 37.5 %, что существенно влияет на скорость обмена информацией. Поэтому в последнее время все более широкое распространение находит *синхронный* принцип передачи информации для ПК, что обеспечивается соответствующими технико-программными средствами (например широко используемая плата **IRMA** фирмы *OSA* для IBM-совместимых ПК и др.).

Наряду с типом сигнала (*аналоговый* или *цифровой*) и способом передачи информации (*асинхронный* или *синхронный*), поддерживаемых *модемами*, они характеризуются *режимами направленности* передачи, определяющими степень *нагрузки* линии связи. В этом плане модемы допускают три режима передачи: *симплексный*, *полудуплексный* и *дуплексный*.

**Симплексный** режим характеризуется *однонаправленностью* передачи информации и используется, как правило, в *системах* сбора и регистрации информации, поступающей *только в одном* направлении (к ЭВМ) от внешних устройств (датчиков и т.д.) *аналогового* типа. Данный режим используется и в некоторого типа локальных информационно-вычислительных сетях.

**Полудуплексный** режим характеризуется *двусторонним* потоком информации, но в каждый момент времени передача производится только в одном направлении. Наиболее широко *полудуплексный* режим используется для обмена информацией между головной ЭВМ и *удаленными* терминалами. Данный режим из-за возникающих временных задержек - относительно *медленный*.

**Дуплексный** режим характеризуется *одновременной* передачей информации в *обоих направлениях*, избавлен от последствий задержек предыдущего, но значительно дороже двух предыдущих. Однако он используется при необходимости обеспечить быстрый обмен информацией между ЭВМ и ее *удаленной периферией*.

**Телефонная** линия является *средой* (или *каналом*), по которой происходит непосредственная передача информации и которая в настоящее время является *наиболее* массово используемой. Более дорогой средой является *коаксиальный* кабель (подобный используемому в *кабельном телевидении*), обеспечивающий высокую пропускную способность порядка  $10^8$  бит/с. *Коаксиальные* кабели наиболее часто используются для организации соединений между ПК в территориально ограниченных (как правило, в пределах одного здания) ЛВС. В ряде случаев они используются вместо *телефонных* линий связи. В отличие от упомянутых сред передачи информации для *микроволновой* служит атмосфера. Данная среда интенсивно используется для передачи больших объемов информации на длинные

расстояния. Но так как распространение **микроволн** используют **микроволновую** среду. Однако в целом ряде случаев данный тип среды имеет существенные недостатки, которые позволяет успешно устранять использование **спутниковой** связи. Стоимость **спутниковых** систем связи достаточно велика и действуют они подобно системе ретрансляторов для случая микроволновой среды. Более того, **спутниковая** связь имеет ряд существенных недостатков: подверженность атмосферным и солнечным влияниям, сложность обеспечения защиты от несанкционированного доступа к передаваемой информации и др. Поэтому, несмотря на доминирующую роль спутниковой связи на длинные расстояния, **революционизирующую** роль в системах связи призвана сыграть **оптоволоконная** технология из-за низкой стоимости, высокой пропускной способности, низком уровне помех и т.д. Данная система связи особенно бурно развивается в США и Японии. Рассмотренные средства связи в значительной степени используются для организации как **модемной** телекоммуникации, так и в **локальных** и **глобальных** информационно-вычислительных сетях.

## 8.2 Программное обеспечение модемной телекоммуникации

**Программное обеспечение модемной телекоммуникации**, в первую очередь, для класса ПК в настоящее время достаточно многочисленно, разнообразно и позволяет организовывать обмен информацией на различных уровнях.

Особую группу составляют ПС, специально ориентированные на поддержку различного типа **модемных телекоммуникаций** на основе ПК, обеспечивающие развитые встроенные средства.

В частности, пакет **Quick Link II Fax (QL2Fax)** фирмы *Smith Micro Software* поставляется совместно с **факс-модемом** и является хорошим примером комплексного решения проблемы организации надежной и удобной факсмодемной связи на основе IBM-совместимых ПК с использованием современных информационных технологий. Пакет имеет развитый и достаточно простой интерфейс с пользователем типа меню, обеспечивая его в любой момент нужной функцией; поддерживаются все возможности модемов, обеспечивая скорость передачи данных в диапазоне 300-15 200 бод (*единица* скорости передачи данных по линиям связи - соответствует одному биту/с).

**Факс-модемный** режим позволяет осуществлять обмен на основе приемо-передающих факс-модемов, обеспечивающих скорость передачи до 9600 бод. Пакет позволяет легко реализовывать логическую связь ПК с абонентом посредством поддержки большого типа **терминальных** эмуляторов; поддерживает широкий набор протоколов передачи и

исправления ошибок, включая протоколы *Kermit* и *SuperKermit*, популярные при организации обмена с ЭВМ общего назначения. Р

Наряду с возможностью посредством *модемной* телекоммуникации осуществлять связь с любым абонентом, имеющим соответствующие средства (ПК+модем+ПС), пользователь получает доступ к ряду известных *абонентских сервисных систем* (АСС), среди которых можно отметить такие популярные, как *E-mail*, *CompuServe*, *DowJones*, *Mead-Data Central*, *Prodigy* и др. Наиболее часто используемой является *E-mail*, представляющая собой *электронную почту*, пользователи которой указывают ее в качестве обычных почтовых реквизитов. Большая

### **8.3 Локальные и глобальные информационно-вычислительные сети**

**Сетевая** телеборотка является дальнейшим развитием *распределенной* обработки информации (являясь в определенной мере альтернативной *мультипроцессорной* локальной обработке), позволяющей обеспечивать доступ к информационно-вычислительным ресурсам широкому кругу удаленных от этих ресурсов пользователей из различных предметных областей. В этом плане *сетевая телеборотка* определяется как *совокупность* взаимосвязанных *аппаратно-программных* средств, *протоколов связи* и *интерфейсов*, обеспечивающая территориальное распределение по сети *управляющих*, *вычислительных* и *информационных* ресурсов. Под *информационно-вычислительной сетью* (ИВС) понимается комплекс территориально распределенных ЭВМ и терминального оборудования, соединенных между собой каналами передачи информации. Проблематика *сетевой* телекоммуникации требует определенного предварительного обсуждения ряда присущих ей понятий и вопросов.

### **8.4. Принципы организации информационно-вычислительных сетей**

ИВС можно рассматривать как *систему* обработки информации с распределенными по территории *аппаратными*, *программными* и *информационными* ресурсами, позволяющую: реализовывать распределенные БД/БЗ и обработку информации, повышать надежность и эффективность функционирования системы в целом, объединять разнотипные вычислительные средства и их периферию, проводить 28-3995 специализацию среди элементов системы, решать качественно *новые* задачи в различных прикладных областях (информационное обслуживание, организация массовых мероприятий, новые виды сервиса и др.) и т.д. Основу каждой ИВС составляет ее *архитектура*, определяемая реализацией структуры сети связи с учетом дисциплины *соединений* узлов и их *топологии*. Все более становясь основой современной компьютерной информатизации, ИВС получают весьма широкое распространение в

различных ***прикладных*** областях. В связи с этим весьма ***актуальным*** становится вопрос разработки эффективной архитектуры, включающей все компоненты и уровни сетевой обработки информации. В рамках архитектуры сети в явном виде рассматриваются вопросы: кодирования информации и ее передачи, контроля ошибок и управления потоками сообщений, адресации абонентов сети, анализа работы сети в аварийных и особых ситуациях и др.

До середины 70-х годов каждая создаваемая сеть передачи информации устанавливала свои специфические правила передачи (*протоколы*) и основные функции. Такой подход сильно затруднял освоение сетей и их совместное использование. Поэтому для проектирования эффективных ИВС была осознана необходимость в разработке на основе системного анализа единых протоколов, распределения функций между компонентами сети и стандартизации ряда других вопросов. Первую попытку в этом направлении сделала фирма IBM введением сетевого SNA-стандарта; за ней последовал целый ряд других известных фирм, разработав собственные стандарты сетевых архитектур (*DNA, BNA, DCA, DINA, DSA, DONA, DCNA* и др.).

## 8.5 Архитектура ИВС

Архитектура ИВС представляет собой систему *протоколов* и *логические структуры*, необходимые для создания действующей сети. Физическими ее элементами являются ЭВМ различных типов и классов, каналы связи, терминальное и другое оборудование.

***Логическая*** структура с точки зрения передачи информации моделирует структурные элементы сети и определяет их архитектуру, распределение функций и интерфейсы в сети. ***Логическая структура*** сети позволяет по единой технологии управлять каналами связи и другими элементами сети с учетом их специфики. В качестве элементов ***логической*** структуры выступают ***узлы*** (моделирующие ЭВМ, терминальные устройства, устройства коммутации и т.д.), ***соединительные элементы*** (моделирующие каналы связи) и ***процессы*** (моделирующие программное функционирование отдельных узлов сети).

Существенную роль в архитектуре ИВС играет ее топология, в основе которой лежат три основных типа: *звездообразная, кольцевая и древовидная*.

***Звездообразная*** топология характеризуется наличием многих узлов, соединенных с центральным (*концентратором*), обеспечивающим управление потоками информации и сообщений. Концентрация управляющей информации в одном узле позволяет обрабатывать большие потоки информации и упрощает управление БД/БЗ. К недостаткам такой организации следует отнести сильную зависимость надежности ИВС от надежности ее центрального узла. Как правило, данная топология присуща

*локальным* ИВС, где вопросы надежности центрального узла решать существенно проще.

**Кольцевая** топология характеризуется совокупностью узлов, соединенных в замкнутую систему с управлением, распределенным по всем ее узлам. В отличие от предыдущей при данной топологии потоки информации пересыпаются в одном направлении, что снимает вопрос их *маршрутизации*. Однако при больших потоках информации такая топология чрезвычайно неэффективна, ибо каналы связи используются всеми узлами сети совместно. В некоторых кольцевых ИВС организованы два кольца соединений для выполнения передач информации в обоих направлениях. Для установления очередности выхода узлов в кольцевую сеть используются методы временного мультиплексирования, передачи маркера и расширения кольца.

**Древовидная** топология представляет собой развитие звездообразной и характеризуется иерархической организацией соединений узлов сети. Децентрализация обработки информации в центральных узлах позволяет в определенной мере *устранять* основной недостаток звездообразных сетей - тесную зависимость надежности всей сети от надежности ее центрального узла. Однако при этом возникает **необходимость** в определении числа уровней иерархии, соответствующих объемам потоков информации, числа узлов, подчиненных узлу верхнего уровня я, и маршрутизации потоков.

В отличие от *кольцевой магистральной* топология характеризуется наличием для всех узлов сети *единой* линии соединения, называемой *магистралью* или *шиной*. В этом случае каждый узел сети использует магистраль для связи с другим узлом; такой тип организации связей наиболее типичен для класса мини-ЭВМ, рассмотренного выше.

**Многосвязная** топология является результатом развития предыдущих, при этом несколько *кольцевых* структур образуют подсеть, в которой обеспечивается управление каждым узлом. В таких сетях можно управлять потоками информации большего объема, чем в кольцевых ИВС, однако при этом необходимо предусматривать специальные узлы *ретрансляции*.

**Комбинированная** топология *объединяет* все указанные выше типы ИВС, реализуя смешанное управление (как *централизованное*, так и *децентрализованное*). В таких ИВС может быть обеспечена высокая надежность при передаче больших объемов информации, но при этом существенно усложняются проектирование и управление БД/БЗ. Для выбора топологии ИВС определенное значение имеют: требуемая дисциплина обслуживания абонентов, технические средства и тип самой сети. В любом случае топология сети оказывает существенное влияние на ее пропускную способность, устойчивость сети к отказам ее узлов, логические возможности сети и стоимость.

Наряду с *топологической* ИВС допускают классификации и по другим существенным признакам. По функциональному назначению различают

сети **информационные**, предоставляющие абоненту в основном информационные услуги; **вычислительные**, выполняющие главным образом решение задач в узловых ЭВМ с обменом программами и данными; и **информационно-вычислительные** (ИВС), обеспечивающие вышеуказанные виды обслуживания в целом. В дальнейшем мы будем говорить о сетях наиболее общего типа - ИВС, т.к. в принципе, любая **вычислительная сеть** может обеспечивать и **информационные** услуги.

По **типу** используемых ЭВМ выделяются **однородные и неоднородные** сети; последние содержат программно несовместимые ЭВМ и чаще всего встречаются на практике.

По территориальному признаку сети делятся на **локальные, региональные и глобальные**. **Локальные** сети охватывают узлы, расположенные друг от друга не более чем на несколько километров (большинство из них охватывают одно большое здание и прилегающую к нему территорию); **региональные** сети охватывают город, район, область, небольшую республику и т.д.; **глобальные** сети охватывают всю страну, несколько стран и целые континенты.

По методу передачи информации различаются сети с коммутацией: **каналов, сообщений, пакетов** и со **смешанной** коммутацией. Современные ИВС характеризуются в основном использованием режима **коммутации пакетов**. Рассмотрим этот вопрос несколько детальнее. Как уже говорилось, под **каналом связи** понимается совокупность: *физическая среда* (телефонная линия, коаксиальный кабель, оптоволокно, атмосфера и др.) и *технические средства* (модемы, адаптеры, мультиплексоры и т.д.) обеспечения передачи информации между узлами сети. Типы каналов по направленности передачи информации (**симплексные, полудуплексные и дуплексные**), а также вопросы передачи дискретной информации по **аналоговым** (наиболее распространенным) каналам связи рассматривались ранее.

## **8.6 Глобальная сеть Internet. Общая характеристика глобальной сети Internet**

В структуре глобальной сети можно выделить три уровня (рисунок 8.1).

Первый – внутренний уровень составляет сеть передачи данных. Она состоит из узлов связи. Каждый узел связи представляет собой совокупность средств передачи данных и состоит из коммутационной ЭВМ и аппаратуры передачи данных.

Во второй уровень входят разнообразные серверы, называемые **хост-ЭВМ** [host computer], которые выполняют в сети задачи по хранению и обработке данных. Такими серверами могут быть, например, серверы различных локальных сетей.

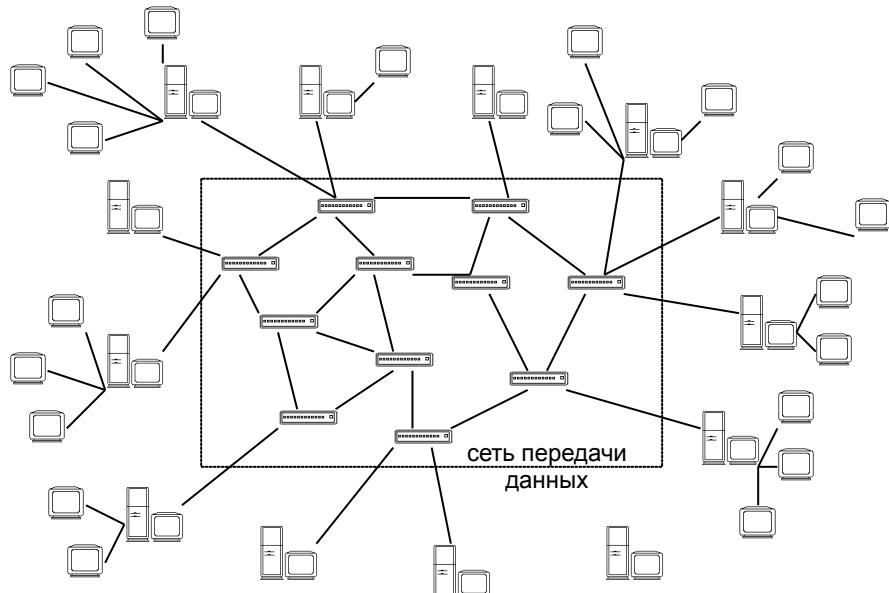


Рисунок 8.1 - Структура глобальной сети

Третий уровень – терминальный – состоит из обычных клиентских рабочих станций, которые пользуются услугами глобальной сети.

Каждая локальная сеть называется *сайтом* [site], а юридическое лицо, обеспечивающее работу сайта – *провайдером*. Сайт состоит из группы серверов, которая выполняет определённые задачи.

Основными характеристиками сети являются: время доставки сообщений, производительность и стоимость обработки данных.

Время доставки сообщений определяется как статистическое среднее время от момента передачи сообщения в сеть до момента получения сообщения адресатом.

Производительность сети представляет собой суммарную производительность серверов.

Стоимость обработки данных определяется как стоимостью средств, используемых для обработки, так и временем доставки и производительностью сети.

Тип сети и все её характеристики в основном определяются строением и принципами работы сети передачи данных, которые описываются протоколом. *Протокол* [protocol] – это система правил, определяющих формат и процедуры передачи данных по сети. Можно сказать, что протокол представляет собой язык, на котором «разговаривают» ЭВМ в сети. Протокол, в частности, определяет, как будут идентифицироваться в сети хост-ЭВМ и как можно найти их в сети, то есть определяются адресация и порядок маршрутизации.

Свойства глобальной сети Internet определяются так называемым IP-протоколом.

## 8.7 Протоколы сети Internet

Основное, что отличает Internet от других сетей - это ее протоколы - TCP/IP. Вообще, термин TCP/IP обычно означает все, что связано с протоколами взаимодействия между компьютерами в Internet. Он охватывает целое семейство протоколов, прикладные программы, и даже саму сеть. TCP/IP - это технология межсетевого взаимодействия, технология internet.

Свое название протокол TCP/IP получил от двух коммуникационных протоколов (или протоколов связи). Это Transmission Control Protocol (TCP) и Internet Protocol (IP). Несмотря на то, что в сети Internet используется большое число других протоколов, сеть Internet часто называют TCP/IP-сетью, так как эти два протокола, безусловно, являются важнейшими.

Как и во всякой другой сети в Internet существует 7 уровней взаимодействия между компьютерами: физический, логический, сетевой, транспортный, уровень сеансов связи, представительский (представительный, уровень представления) и прикладной уровень. Соответственно каждому уровню взаимодействия соответствует набор протоколов (т.е. правил взаимодействия).

Протоколы физического уровня определяют вид и характеристики линий связи между компьютерами. В Internet используются практически все известные в настоящее время способы связи от простого провода (витая пара) до волоконно-оптических линий связи (ВОЛС).

Для каждого типа линий связи разработан соответствующий протокол логического уровня, занимающийся управлением передачей информации по каналу. К протоколам логического уровня для телефонных линий относятся протоколы SLIP (Serial Line Interface Protocol) и PPP (Point to Point Protocol). Для связи по кабелю локальной сети - это пакетные драйверы плат ЛВС.

Протоколы сетевого уровня отвечают за передачу данных между устройствами в разных сетях, то есть занимаются маршрутизацией пакетов в сети. К протоколам сетевого уровня принадлежат IP (Internet Protocol) и ARP (Address Resolution Protocol).

Протоколы транспортного уровня управляют передачей данных из одной программы в другую. К протоколам транспортного уровня принадлежат TCP (Transmission Control Protocol) и UDP (User Datagram Protocol).

Протоколы уровня сеансов связи отвечают за установку, поддержание и уничтожение соответствующих каналов. В Internet этим занимаются уже упомянутые TCP и UDP протоколы, а также протокол UUCP (Unix to Unix Copy Protocol).

Протоколы представительского уровня занимаются обслуживанием прикладных программ. К программам представительского уровня принадлежат программы, запускаемые, к примеру, на Unix-сервере, для предоставления различных услуг абонентам. К таким программам относятся:

telnet-сервер, FTP-сервер, Gopher-сервер, NFS-сервер, NNTP (Net News Transfer Protocol), SMTP (Simple Mail Transfer Protocol), POP2 и POP3 (Post Office Protocol) и т.д.

К протоколам прикладного уровня относятся сетевые услуги и программы их предоставления. Для того чтобы услуги были предоставлены пользователю каждый компьютер в глобальной сети имеет уникальный адрес.

## 8.8 Адрес компьютера

Каждый компьютер, подключенный к сети Интернет имеет два равноценных уникальных адреса: цифровой IP-адрес и символический доменный адрес. Присваивание адресов происходит по следующей схеме: международная организация Сетевой информационный центр выдает группы адресов владельцам локальных сетей, а последние распределяют конкретные адреса по своему усмотрению.

*IP-адрес* компьютера имеет длину 4 байта. Обычно первый и второй байты определяют адрес сети, третий байт определяет адрес подсети, а четвертый - адрес компьютера в подсети. Для удобства IP-адрес записывают в виде четырех чисел со значениями от 0 до 255, разделенных точками, например: 145.37.5.150. Адрес сети - 145.37; адрес подсети - 5; адрес компьютера в подсети - 150.

*Доменный адрес* (англ. *domain* - область), в отличие от цифрового, является символическим и легче запоминается человеком. Пример доменного адреса: *barsuk.les.nora.ru*. Здесь домен *barsuk* - имя реального компьютера, обладающего IP-адресом, домен *les* - имя группы, присвоившей имя этому компьютеру, домен *nora* - имя более крупной группы, присвоившей имя домену *les*, и т.д. В процессе передачи данных доменный адрес преобразуются в IP-адрес.

### Услуги предоставляемые сетью

Все услуги предоставляемые сетью Internet можно условно поделить на две категории: обмен информацией между абонентами сети и использование баз данных сети.

К числу услуг связи между абонентами принадлежат.

**Telnet** - удаленный доступ. Дает возможность абоненту работать на любой ЭВМ сети Internet как на своей собственной. То есть запускать программы, менять режим работы и т.д.

**FTP** (File Transfer Protocol) - протокол передачи файлов. Дает возможность абоненту обмениваться двоичными и текстовыми файлами с любым компьютером сети. Установив связь с удаленным компьютером, пользователь может скопировать файл с удаленного компьютера на свой или скопировать файл со своего компьютера на удаленный.

**NFS** (Network File System) - распределенная файловая система. Дает возможность абоненту пользоваться файловой системой удаленного компьютера, как своей собственной.

**Электронная почта** - обмен почтовыми сообщениями с любым абонентом сети Internet. Существует возможность отправки как текстовых, так и двоичных файлов. На размер почтового сообщения в сети Internet накладывается следующее ограничение - размер почтового сообщения не должен превышать 64 килобайт.

**Новости** - получение сетевых новостей и электронных досок объявлений сети и возможность помещения информации на доски объявлений сети. Пользователь может по своему выбору подписаться на любые группы новостей.

**Rsh** (Remote Shell) - удаленный доступ. Аналог Telnet, но работает только в том случае, если на удаленном компьютере стоит ОС UNIX.

**Rexec** (Remote Execution) - выполнение одной команды на удаленной UNIX-машине.

**Ping** - проверка доступности удаленной ЭВМ по сети.

**Talk** - дает возможность открытия "разговора" с пользователем удаленной ЭВМ. При этом на экране одновременно виден вводимый текст и ответ удаленного пользователя.

**Iptunnel** - дает возможность доступа к серверу ЛВС NetWare с которым нет непосредственной связи по ЛВС, а имеется лишь связь по сети Internet.

**Whois** - адресная книга сети Internet. По запросу абонент может получить информацию о принадлежности удаленного компьютера, о пользователях.

**Finger** - получение информации о пользователях удаленного компьютера.

**Шлюзы** - дают возможность абоненту отправлять сообщения в сети, не работающие с протоколами TCP\IP (FidoNet, Goldnet, AT50).

## Лекция9 (2 часа)

*Системы информационного поиска сети Интернет*

*Электронная почта и система телеконференций*

*Работа с программой Internet Explorer 5.0*

*Электронная почта*

*Работа с программой Outlook Express*

### **9. Системы информационного поиска сети Интернет**

В Интернет представлена информация на любые темы, которые только можно себе представить. Но найти в ней нужную информацию не так-то легко из-за того, что сеть по своей природе не имеет чёткой структуры. Поэтому для ориентировки в Интернет и быстрого получения свежей справочной информации разработаны системы поиска информации.

Все системы поиска информации Интернет располагаются на специально выделенных компьютерах с мощными каналами связи. Ежеминутно они бесплатно обслуживают огромное количество клиентов.

Поисковые системы можно разбить на два типа:

- *предметные каталоги*, формируемые людьми-редакторами;
- *автоматические индексы*, формируемые специальными компьютерными программами, без участия людей.

Системы, основанные на предметных каталогах. Используют базы данных, формируемые специалистами-редакторами, которые отбирают информацию, устанавливают связи для баз данных, организуют и снабжают данные в разных поисковых категориях перекрёстными ссылками. Кампании, владеющие предметными каталогами, *непрерывно исследуют, описывают и каталогизируют содержимое WWW-серверов и других сетевых ресурсов, разбросанных по всему миру*. В результате этой работы клиенты Интернет имеют постоянно обновляющиеся иерархические (древовидные) каталоги, на верхнем уровне которых собраны самые общие категории, такие как “*бизнес*”, “*наука*”, “*искусство*” и т.п., а элементы самого нижнего уровня представляют собой ссылки на отдельные *WWW*-страницы и серверы вместе с кратким описанием их содержимого.

Каталоги, составленные людьми, *более осмыслены*, чем автоматические индексы. Их очень мало, так как *их создание и поддержка требуют огромных затрат*.

Автоматические индексы. Переоценить их трудно. Поиск по ключевым словам в одной базе данных, занимающий в худшем случае несколько секунд, принесёт те же результаты, что и обшаривание всех *WWW*-страниц во всей сети Интернет.

Автоматический индекс состоит из трёх частей:

- *программы-робота*;
- *базы данных*, собираемой этим роботом;

- *интерфейса* для поиска в этой базе, с которым и работает пользователь.

Все эти компоненты функционируют *без вмешательства человека*.

К автоматическим индексам следует прибегать только тогда, когда ключевые слова точно известны, например, фамилия человека или несколько специфических терминов из соответствующей области. Индексы получают информацию из каждого отдельного узла, регистрируют и индексируют её и добавляют к своим базам данных.

В Интернет один и тот же узел сети может одновременно работать по нескольким протоколам. Поэтому крупные узлы сети сейчас обладают полным набором серверов, и к ним можно обращаться почти по любому из существующих протоколов.

К системам автоматизированного поиска информации в сети Internet принадлежат следующие системы.

**Gopher** - наиболее широко распространенное средство поиска информации в сети Internet, позволяющее находить информацию по ключевым словам и фразам. Gopher позволяет получить информацию без указания имен и адресов авторов, благодаря чему пользователь не тратит много времени и нервов. Он просто сообщает системе Gopher, что именно ему нужно, и система находит соответствующие данные. Gopher-серверов свыше двух тысяч, поэтому с их помощью не всегда просто найти требуемую информацию. В случае возникших затруднений можно воспользоваться службой VERONICA. VERONICA осуществляет поиск более чем в 500 системах Gopher, освобождая пользователя от необходимости просматривать их вручную.

**WAIS** - еще более мощное средство получения информации, чем Gopher, поскольку оно осуществляет поиск ключевых слов во всех текстах документов. Запросы посылаются в WAIS на упрощенном английском языке. Это значительно легче, чем формулировать их на языке алгебры логики, и это делает WAIS более привлекательной для пользователей-непрофессионалов.

**WWW** - система для работы с гипертекстом. Потенциально она является наиболее мощным средством поиска. Гипертекст соединяет различные документы на основе заранее заданного набора слов. Например, когда в тексте встречается новое слово или понятие, система, работающая с гипертекстом, дает возможность перейти к другому документу, в котором это слово или понятие рассматривается более подробно.

WWW часто используется в качестве интерфейса к базам данных WAIS, но отсутствие гипертекстовых связей ограничивает возможности WWW до простого просмотра.

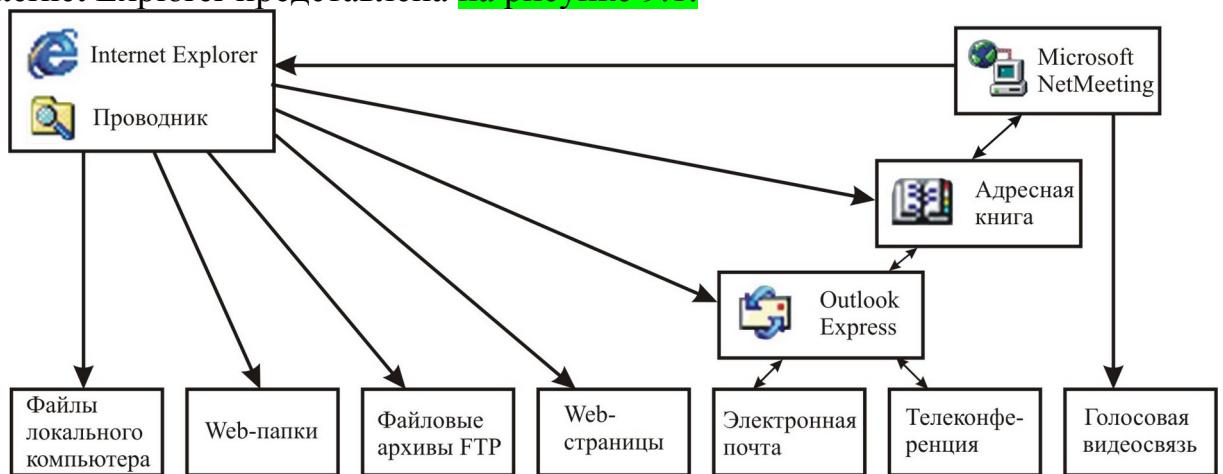
Практически все услуги сети построены на принципе клиент-сервер. Сервером в сети Internet называется компьютер способный предоставлять клиентам (по мере прихода от них запросов) некоторые сетевые услуги.

Взаимодействие клиент-сервер строится обычно следующим образом. По приходу запросов от клиентов сервер запускает различные программы предоставления сетевых услуг. По мере выполнения запущенных программ сервер отвечает на запросы клиентов.

Все программное обеспечение сети также можно поделить на клиентское и серверное. При этом программное обеспечение сервера занимается предоставлением сетевых услуг, а клиентское программное обеспечение обеспечивает передачу запросов серверу и получение ответов от него.

## 9.1 Работа с программой Internet Explorer 5.0

Примером броузера, предназначенного для просмотра Web-документов, может служить Internet Explorer 5.0. Программа предоставляет единый метод доступа к локальным документам компьютера, ресурсам корпоративной сети Intranet и к информации, доступной в Интернете. Она обеспечивает работу с World Wide Web, предоставляет идентичные средства работы с локальными папками компьютера и файловыми архивами *FTP*, дает доступ к средствам связи через Интернет. Соответствующие программы (Outlook Express и Microsoft NetMeeting) автономны, но рассматриваются как часть пакета Internet Explorer 5.0. Схема использования Интернета через Internet Explorer представлена на [рисунке 9.1](#).



*Рисунок 9.1 - Организация доступа к ресурсам Интернета*

Для запуска броузера Internet Explorer можно использовать значок Internet Explorer на Рабочем столе или на Панели быстрого запуска, а также Главное меню (*Пуск → Программы → Internet Explorer*). Кроме того, программа запускается автоматически при попытке открыть документ Интернета или локальный документ в формате *HTML*. Для этой цели можно использовать ярлыки Web-страниц, папку Избранное (*Пуск → Избранное*) или пункт меню «*Избранное*» в строке меню окна папки или программы.

«Проводник», панель инструментов «Рабочего стола» Адрес или поле ввода в диалоговом окне Запуск программы (*Пуск → Выполнить*).

Если соединение с Интернетом отсутствует, то после запуска программы на экране появится диалоговое окно для управления установкой соединения. При невозможности установить соединение сохраняется возможность просмотра в *автономном режиме* ранее загруженных Web-документов. При наличии соединения после запуска программы на экране появится так называемая «домашняя», или *основная*, страница, выбранная при настройке программы.

## 9.2 Электронная почта

*Электронная почта* (*Electronic mail*, англ. *mail* - почта, сокр. *E-mail*, читается “и-мэйл”). Служит для передачи текстовых сообщений в пределах Интернет, а также между другими сетями электронной почты. К тексту письма современные почтовые программы позволяют прикреплять звуковые и графические файлы, а также двоичные файлы - программы.

При использовании электронной почты каждому абоненту присваивается уникальный почтовый адрес, формат которого имеет вид: <имя пользователя> @ < имя почтового сервера>. Например: *earth@space.com*, где *earth* - имя пользователя, *space.com* - имя компьютера, @ - разделительный символ “эт коммерческое”.

Сообщения, поступающие по *E-Mail*, хранятся в специальном «почтовом» компьютере в выделенной для получателя области дисковой памяти (его “*почтовом ящике*”), откуда он может их выгрузить и прочитать с помощью специальной программы-клиента.

Для отсылки сообщения нужно знать *электронный адрес абонента*. При качественной связи электронное письмо доходит в любую точку мира в течение нескольких минут.

Пользователи электронной почты стремятся придерживаться правил сетевого этикета (*нэтикета*), а для выражения эмоций используют схематические изображения человеческого лица, так называемые смайлики (англ. *smiley*, “улыбочка”), некоторые из которых приведены ниже.

*Смайлики (рассматривайте, склонив голову влево)*

:-)	улыбка	:-Q	курит	:-)~	пускает	слюнки
:-)))	хохот	:-@	кричит	8:-)	маленькая	девочка
:-~)	насморк	:-()	грусть	:-*	съел	горькое
:-*)	пьяница	':-(	плачет	':-)	плачет	от счастья
:-{)	усатый	;:-)	хитрец	:-&	поклялся	молчать
:-[	вампир	>:-()	злится	O-)		аквалангист
-:-) панк		: 0 зевает		=8-) носит	очки	

## Система телеконференций

*Система телеконференций Usenet* (от *Users Network*). Это основанная на использовании компьютерной техники система, позволяющая пользователям, несмотря на их взаимную удаленность в пространстве, а иногда, и во времени, участвовать в совместных мероприятиях, таких, как организация и управление сложными проектами. Она организует *коллективные обсуждения* по различным направлениям, называемые *телеконференциями*.

Пользователи обеспечиваются терминалами (обычно это дисплеи и клавиатуры), подсоединенными к компьютеру, которые позволяют им связываться с другими членами группы. Для передачи информации между участниками совещания используются *линии связи*.

Работа системы регулируется *координатором*, в функции которого входит организация работы участников совещания, обеспечение их присутствия на совещании и передача сообщаемой ими информации другим участникам совещания.

*Телеконференции* (или *группы, новостей*) представляют собой средства распространения сообщений, не предназначенных для конкретного адресата. Информация о наличии сообщения постепенно распространяется от одного *сервера новостей* к другому. Сообщение хранится на сервере в течение некоторого времени (от нескольких дней до нескольких недель) после чего сбрасывается. Пользователь имеет доступ ко всем сообщениям, имеющимся на данном сервере новостей.

В некоторых системах телеконференцсвязи участники имеют возможность «видеть» друг друга, что обеспечивается подсоединенными к системам телевизионными камерами и дисплеями

Авторы сообщений направляют сообщения в тематические телеконференции. В каждой телеконференции проводится ряд дискуссий по конкретным темам.

Сегодня Usenet имеет более десяти тысяч дискуссионных групп (*NewsGroups*) или телеконференций, каждая из которых посвящена определённой теме и является средством обмена мнениями. Телеконференции разбиты на несколько групп:

- *news* - вопросы, касающиеся системы телеконференций;
- *comp* - компьютеры и программное обеспечение;
- *rec* - развлечения, хобби и искусства;
- *sci* - научно-исследовательская деятельность и приложения;
- *soc* - социальные вопросы;
- *talk* - дебаты по различным спорным вопросам;
- *misc* - всё остальное.

Внутри этих категорий существует *иерархия*. Так, например, *rec.music.beatles* - это дискуссия о творчестве Битлз, входящая в подгруппу “музыка” группы дискуссий по искусству.

Имена телеконференций образуют иерархическую структуру, не имеющую единого корня. Элементы имени разделяются точками, старшие элементы располагаются слева, младшие – правее. Чем больше элементов в имени телеконференции, тем более узкой теме она посвящена.

При обращении к телеконференции сервер новостей передает на компьютер пользователя заголовки имеющихся в ней и не прочитанных пользователем сообщений. Текст сообщений передается позже в соответствии с указаниями пользователя и настройками программы чтения сообщений телеконференций. Можно также отправить в телеконференцию новое сообщение или отклик.

Существует большой выбор *программ чтения телеконференций*, которые формируют материал дискуссий в упорядоченном виде и предоставляют в распоряжение корреспондентов.

Аналог телеконференций в других сетях - “электронная доска объявлений” (Bulletin Board System, BBS).

Хотя электронная почта и служба новостей – разные службы, для пользователя они почти одинаковы, так как и в том и в другом случае речь идет об отправке и получении сообщений.

Для работы с электронной почтой и телеконференциями обычно используют единую программу, так как и в том и в другом случае речь идет об отправке и приеме сообщений. Часто оказывается удобным объединение средств работы с этими службами в рамках одной программы. Например, так сделано в программе Outlook Express, которая позволяет получать и отправлять сообщения электронной почты и телеконференций, используя аналогичные средства.

### **9.3 Работа с программой Outlook Express.3. Почтовая служба (E-mail)**

Электронная почта появилась одной из самых первых. До сих пор это одно из наиболее используемых, мощных и удобных средств сети. Эта служба позволяет обмениваться почтовыми сообщениями.

Основой для обмена такими сообщениями служат специальные почтовые сервера. На этих серверах *каждый абонент имеет почтовый ящик* - место, в котором хранится его почта.

Программа-клиент по команде пользователя обращается к этому почтовому ящику для выполнения операции с его письмами - просмотр, удаление, перемещение, отправку и пересылку.

Каждый ящик имеет хотя бы один почтовый адрес, состоящий из имени ящика (чаще всего это имя пользователя) и DNS-адреса сервера. Имя ящика и адрес разделяются символом @.

Пример такого почтового *Flamer@hotmail.com*

В принципе система позволяет одному ящику получать письма на несколько адресов, одному адресу - на несколько почтовых ящиков.

#### Отправка и получение сообщений



*Рисунок 9.2 – Программа Outlook Express*

Для работы с электронной почтой и телеконференциями обычно используют единую программу, так как и в том и в другом случае речь идет об отправке и приеме сообщений. Часто оказывается удобным объединение средств работы с этими службами в рамках одной программы. Например, так сделано в программе Outlook Express, которая позволяет получать и отправлять сообщения электронной почты и телеконференций, используя аналогичные средства (рисунок 9.2).

Возможность использования электронной почты сегодня перестала быть самостоятельной услугой и автоматически предоставляется тем, кто подключается к Интернету. Адрес электронной почты состоит из двух частей. Доменный адрес условно соответствует двум последним частям обозначения компьютера в адресе *URL* и фактически представляет собой адрес локальной сети, к которой принадлежит конкретный пользователь. Вторая часть адреса (которая в записи идет перед первой и отделяется от нее символом «@») указывает конкретного пользователя в этой локальной сети. Сообщения для данного адресата накапливаются на *почтовом сервере* а затем передаются на компьютер адресата по запросу.

Например, пользователь, подключающийся к Интернету через поставщика услуг *ABCDE*, может иметь адрес типа *tupname@abcde.ru*.

*Телеконференции* (или *группы, новостей*) представляют собой средства распространения сообщений, не предназначенных для конкретного адресата. Информация о наличии сообщения постепенно распространяется от одного *сервера новостей* к другому. Сообщение хранится на сервере в течение некоторого времени (от нескольких дней до нескольких недель) после чего сбрасывается. Пользователь имеет доступ ко всем сообщениям, имеющимся на данном сервере новостей.

Авторы сообщений направляют их в тематические телеконференции. Имена телеконференций образуют иерархическую структуру, не имеющую единого корня. Элементы имени разделяются точками, старшие элементы

располагаются слева, младшие – правее. Чем больше элементов в имени телеконференции, тем более узкой теме она посвящена.

Например, телеконференция *news.announces.newusers* содержит регулярно обновляемый набор сообщений (на английском языке), предназначенный для ознакомления начинающих с правилами использования телеконференций и сетевым этикетом. А скажем, с элементов *comp.hardware...* начинается целое семейство телеконференций, посвященных различным темам, связанным с аппаратным обеспечением компьютеров.

При обращении к телеконференции сервер новостей передает на компьютер пользователя заголовки имеющихся в ней и не прочитанных пользователем сообщений. Текст сообщений передается позже в соответствии с указаниями пользователя и настройками программы чтения сообщений телеконференций. Можно также отправить в телеконференцию новое сообщение или отклик.

Хотя электронная поста и служба новостей – разные службы, для пользователя они почти одинаковы, так как и в том и в другом случае речь идет об отправке и получении сообщений.

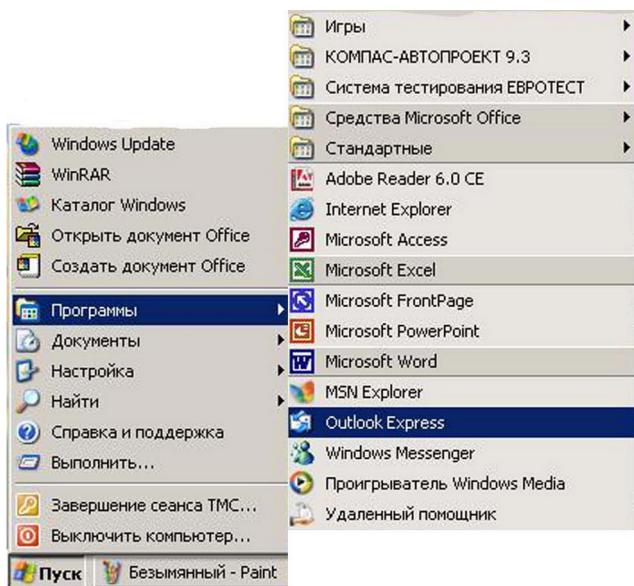
Сообщение, отправляемое в телеконференцию, носит общественный характер, а частную информацию следует пересылать по электронной почте. Однако ни одна из этих служб не годится для пересылки *конфиденциальной* информации, которая не должна быть доступна посторонним.

## **Работа с программой Outlook Express**

**Создание учетной записи.** Сообщения электронной почты и телеконференций накапливаются, соответственно, на *почтовом сервере* и *сервере новостей*. Для работы с этими службами предназначена программа *Microsoft Outlook Express*. Она является стандартной программой Windows и запускается путем нажатия команды «Пуск», выбора раздела «Программы» (рисунок 9.3).

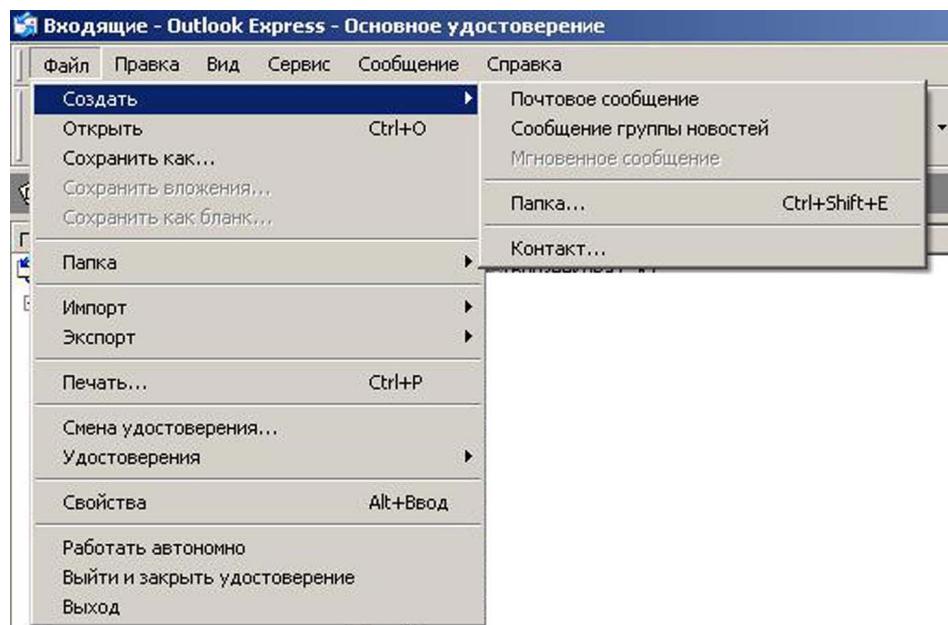
Так как сообщения поступают и отправляются через сервер, программе требуется указать информацию об используемом сервере. Эта информация хранится в виде *учетной записи*.

В программе *Outlook Express* учетную запись создают командой «Сервис» пунктом «Учетные записи». В диалоговом окне «Учетные записи» в Интернете надо щелкнуть на кнопке «Добавить» и выбрать в открывшемся меню службу, для которой создается учетная запись. Последующая информация вводится под управлением мастера и включает имя, указываемое как имя отправителя, адрес электронной почты, имя используемого сервера и, в случае необходимости, имя пользователя и пароль.



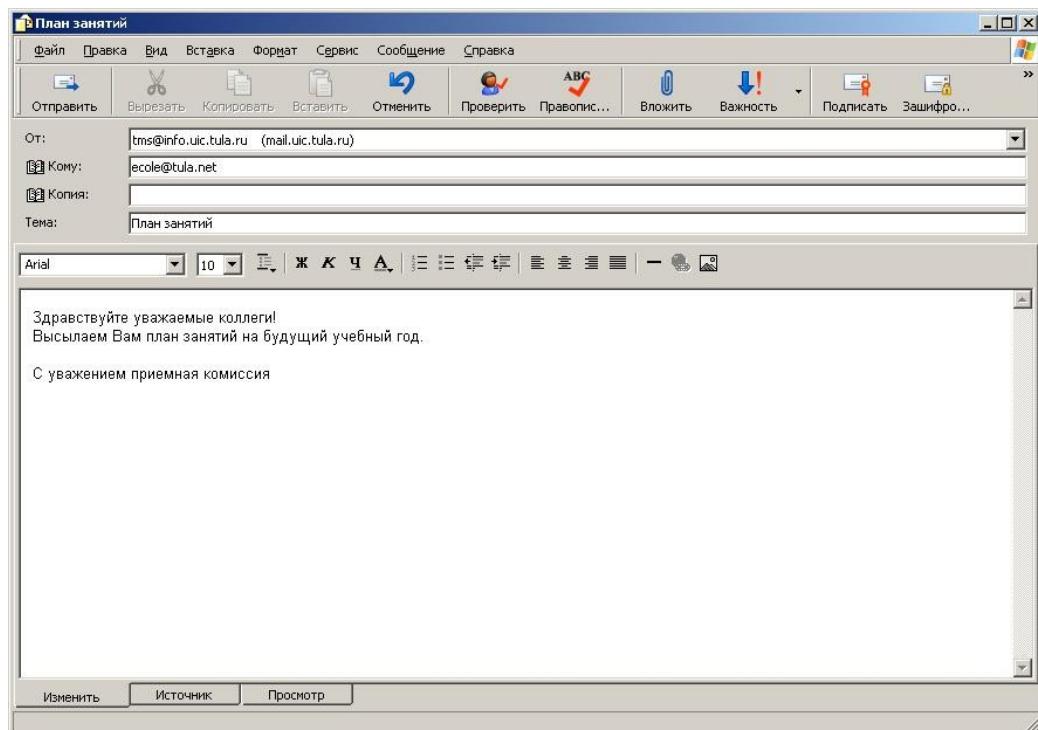
*Рисунок 9.3 – Вызов программы Microsoft Outlook Express*

**Создание сообщения электронной почты.** Чтобы отправить сообщение электронной почты, его надо создать. Для этого следует выбрать меню «Файл», подменю «Создать» (рисунок 9.4).



*Рисунок 9.4 – Меню «Файл»*

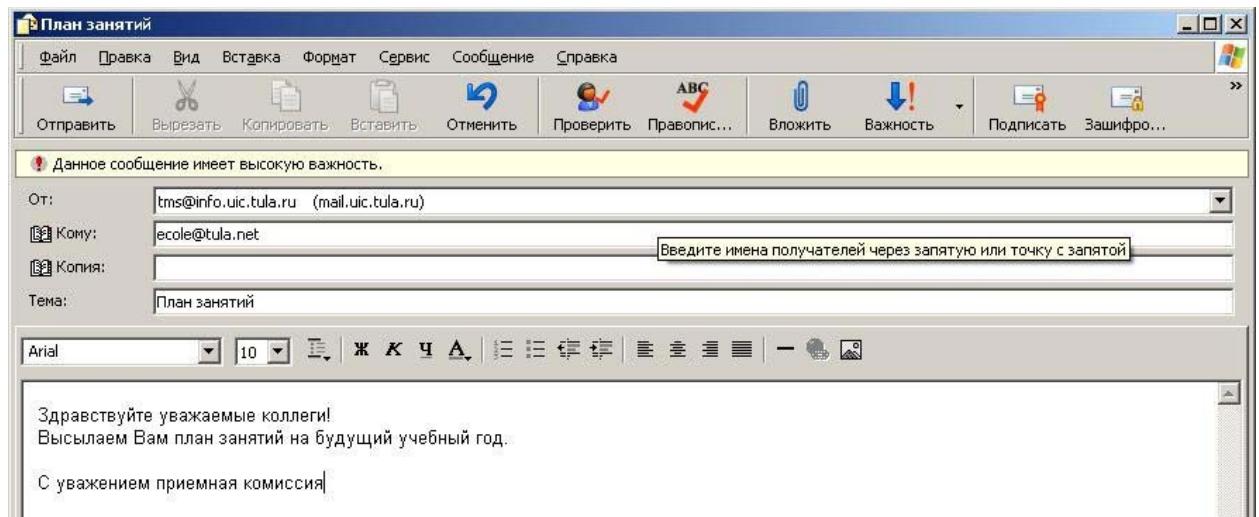
При этом открывается окно «Создать сообщение», рабочая область которого разбивается на две основные части. В верхней части располагаются поля для ввода служебной информации, а в нижней – собственно текст сообщения. В поле «Тема» вводится краткое описание вопроса, которому посвящено сообщение (рисунок 9.5).



*Рисунок 9.5 - Создание сообщения для отправки по электронной почте*

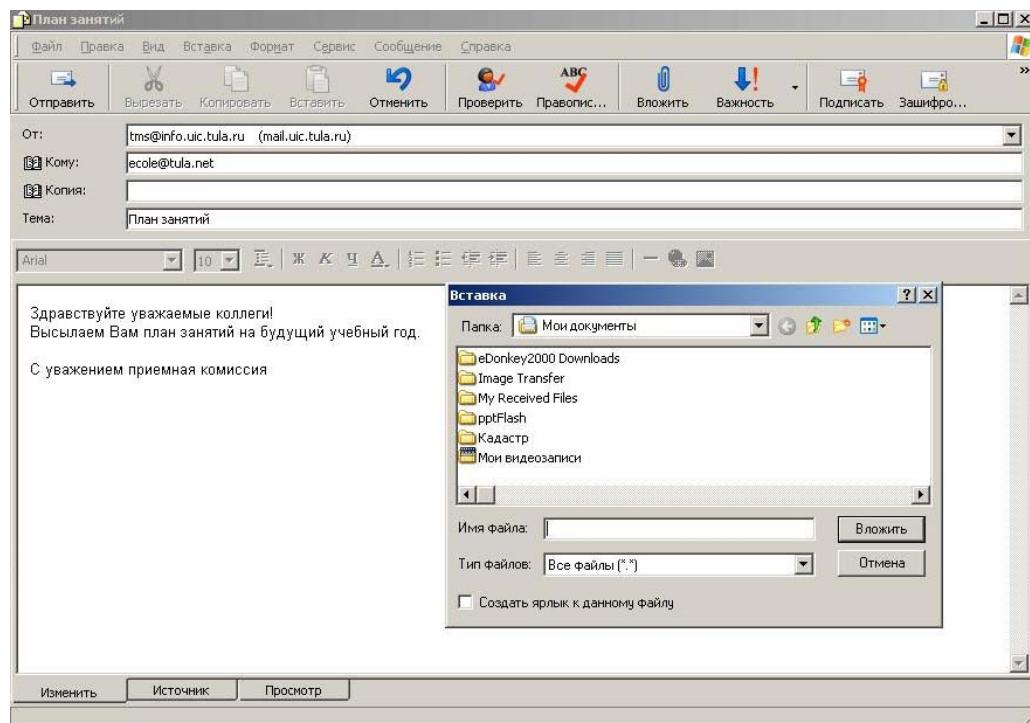
В поле «Кому» вводится адрес основного получателя письма, в поле «Копия» – адреса получателей копии.

Письму также можно указать степень важности, выбрав пункт «Важность» (рисунок 9.6).



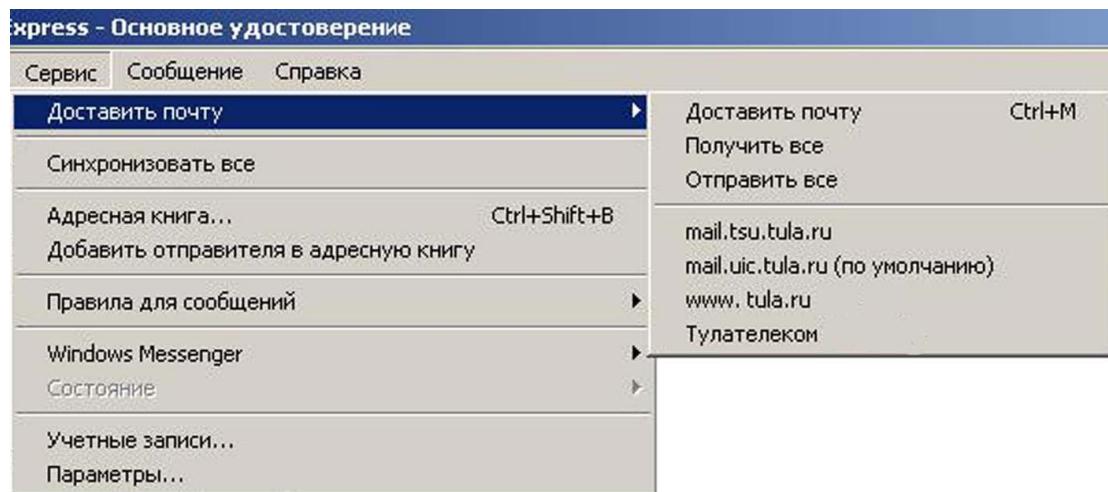
*Рисунок 9.6 – Вид письма с заданной важностью*

При необходимости к написанному письму можно добавить любой файл или несколько файлов. Для этого необходимо на панели выбрать пункт «Вложиться» (рисунок 9.7). В появившемся меню «Вставка» необходимо выбрать файлы для присоединения.



*Рисунок 9.7 – Меню вставки присоединяемых файлов*

В ходе создания и редактирования сообщения наличие связи с почтовым сервером не требуется. Такая связь нужна только в момент отправки (получения) сообщений. Программа *Outlook Express* устроена таким образом, что отправка и получение сообщений осуществляются одновременно. Так, получение и доставка почты осуществляются по щелчку на кнопке «*Отправить*» в окне создания сообщения или по щелчку на кнопке «*Доставить*» в основном окне программы *Outlook Express*. Также для отправки и получения почты можно использовать меню «*Сервис*» (рисунок 9.8).



*Рисунок 9.8 – Меню «Сервис»*

Сообщения электронной почты размещаются в системе «внутренних» папок программы *Outlook Express*. Поступившие сообщения заносятся в папку «*Входящие*». Открыв эту папку щелчком на ее значке на панели «Папки», можно увидеть в правой области списка поступивших сообщений. Если выбрать щелчком любое из сообщений, его содержание отобразится в области, расположенной ниже списка. Двойной щелчок позволяет открыть и прочитать сообщение в отдельном окне.



*Рисунок 9.9 – Информация об отсутствии входящих сообщений*

Если новых сообщений нет, то в нижней части окна появляется сообщение «*Новых сообщений нет*» (рисунок 9.9).

**Подготовка ответов на сообщения.** Как правило, использование любых средств коммуникации подразумевает диалог. В случае электронной почты речь идет об отправке ответов на полученные сообщения. Программа *Outlook Express* включает средства, упрощающие подготовку таких ответов. Открыв полученное сообщение в отдельном окне, можно использовать кнопки на панели инструментов.

- Кнопка «*Ответить*» отправителю служит для ответа автору письма. При этом в окне создания сообщения автоматически заполняются поля Кому и Тема, а в «тело» сообщения заносится текст исходного сообщения, что позволяет привязать комментарии непосредственно к отдельным фразам полученного письма.
- Кнопка «*Ответить*» всем служит для отправки ответа автору письма, а также всем, кто получил исходное сообщение. В окне создания сообщения автоматически заполняются поля Кому, Копия и Тема. Текст исходного сообщения копируется в тело сообщения.
- Кнопка «*Переслать*» позволяет отправить полученное сообщение другому корреспонденту. В данном случае автоматически заполняется только поле Тема, так как нового адресата необходимо указать дополнительно.

**Чтение сообщений телеконференций.** Механизм чтения сообщений телеконференций примерно тот же, что и при использовании электронной почты. После создания учетной записи для сервера новостей на панели «Папки» появляется значок, соответствующий выбранному серверу. После выбора этого значка автоматически открывается диалоговое окно Подписка на группу новостей, а программа получает список телеконференций, поддерживаемых данным сервером. Выбрав телеконференцию, следует щелкнуть на кнопке «*Подписаться*». Телеконференции с подпиской

отображаются непосредственно на панели «Папки», и для доступа к ним не требуется открывать диалоговое окно Подписка на группу новостей.

Работа с сообщениями телеконференций осуществляется примерно также, как с сообщениями электронной почты. При просмотре сообщения в отдельном окне можно «Ответить в группу» (отправить отклик в телеконференцию), «Ответить автору» (сообщение отправляется непосредственно автору по электронной почте) или «Переслать» сообщение по электронной почте другому корреспонденту.

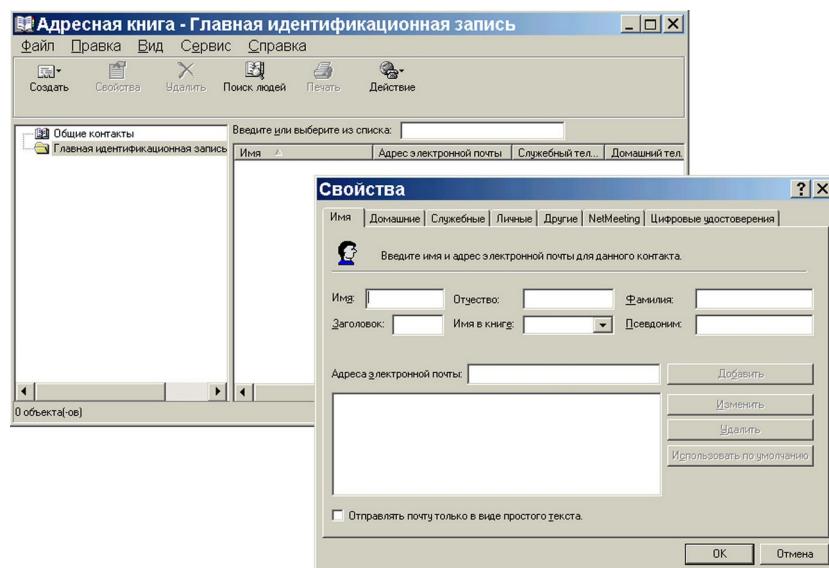
### **Работа с адресной книгой**

При активном использовании электронной почты общее число корреспондентов может достигать многих сотен. Помнить все электронные адреса просто немыслимо. Облегчить эту работу позволяет специальная программа «Адресная книга».

С ее помощью можно:

- запоминать адреса корреспондентов, от которых поступили сообщения;
- автоматизировать ввод адресов корреспондентов;
- организовать проверку правильности введенных адресов;
- упростить отправку сообщений группам адресатов.

Открывать Адресную книгу вручную (Пуск → Программы → Стандартные → Адресная книга) требуется только для ее редактирования. Чтобы добавить нового адресата, следует щелкнуть на кнопке «Создать» и выбрать в открывшемся меню пункт «Создать контакт». Откроется диалоговое окно «Свойства», содержащее многочисленные вкладки, предназначенные для ввода разнообразной информации об адресате. Имя и адрес электронной почты задаются на вкладке «Имя». Удобно использовать также поле «Псевдоним»: данные, введенные в это поле, можно указывать вместо адреса в ходе создания сообщения (рисунок 9.10).



*Рисунок 9.10 – Окна программы «Адресная книга»*

Если информация о корреспонденте поступила вместе с полученным от него сообщением, то занести эти данные в *Адресную книгу* можно непосредственно из программы *Outlook Express*. Для этого надо щелкнуть правой кнопкой мыши на имени , адресата в поле «*От*» в списке сообщений или в окне сообщения и выбрать в контекстном меню команды «*Добавить отправителя в адресную книгу*» или «*Добавить в адресную книгу*», соответственно.

Чтобы воспользоваться *Адресной книгой* для ввода адреса, надо в ходе создания сообщения щелкнуть на заголовке соответствующего поля («*Кому*», «*Копия*» или «*Скрытая*»). Адреса, помещаемые в каждое из этих полей, выбираются в диалоговом окне *Выбрать получателей*.

Адрес, взятый из Адресной книги, выделяется в соответствующем поле подчеркиванием. Если какие-то из адресов вводились вручную, но должны быть в Адресной книге, их можно проверить при помощи команды «*Сервис*» → «*Проверить имена*». Найденные адреса также будут подчеркнуты, ненайденные можно исправить, выбрав один из нескольких подходящих адресов, или занести в Адресную книгу.

Если необходимо регулярно отправлять сообщение одной и той же группе корреспондентов, Адресная книга позволяет создать и использовать *группу адресов*. Для этого используется команда «*Создать*» → «*Создать группу*». При добавлении участников в группу их адреса могут выбираться из Адресной книги или создаваться на месте. При указании в поле адреса имени группы сообщение отправляется всем выбранным корреспондентам.