Федеральное агентство по образованию

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ГОУВПО «АмГУ»

УТВЕРЖДАЮ

Зав.кафедрой БЖД

_____А.Б.Булгаков

«_____»_____2007г.

Программное обеспечение задач БЖД

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС

для специальности 280101 «Безопасность жизнедеятельности в техносфере»

Составитель: Дрюков А. А., ассистент кафедры БЖД

Благовещенск 2007 г.

Печатается по решению
редакционно-издательского совета
инженерно-физического факультета
Амурского государственного
университета

А. А. Дрюков

Учебно-методический комплекс по дисциплине «Программное обеспечение задач БЖД» для студентов очной и заочной сокращенной форм обучения специальности 280101 «Безопасность жизнедеятельности в техносфере». - Благовещенск: Амурский гос. ун-т, 2007. – 268 с.

Учебно-методические рекомендации ориентированы на оказание помощи студентам очной и заочной форм обучения по специальности 280101 «Безопасность жизнедеятельности в техносфере» для формирования представления о назначении и видах системного программного обеспечения работоспособности компьютеров, применения прикладного программного обеспечения для решения задач в области безопасности жизнедеятельности, ознакомление студентов с инструментарием технологии программирования.

СОДЕРЖАНИЕ

Федеральное агентство по образованию РФ

Амурский государственный университет

УТВЕРЖДАЮ
Проректор по УНР
_____Е.С. Астапова
подпись, И.О.Ф

«__» _____ 2006 г.

РАБОЧАЯ ПРОГРАММА
по дисциплине «Программное обеспечение задач БЖД»

(наименование дисциплины)

для специальности 280101, Безопасность жизнедеятельности в техносфере
(шифр и наименование специальности)

Курс 2 Семестр 3

Лекции 36 (час.) Экзамен нет
(семестр)

Лабораторные занятия 18 (час.) Зачет 3
(семестр)

Самостоятельная работа 22 (час.)

Всего часов 76

Составитель А. А. Дрюков, ассистент
(И.О.Ф., должность, ученое звание)

Факультет инженерно-физический

Кафедра БЖД

2006 г.

Рабочая программа авторская

Рабочая программа обсуждена на заседании кафедры БЖД

_____

«__» _____200___ г., протокол № _____

Заведующий кафедрой А.Б. Булгаков

Рабочая программа одобрена на заседании УМС 280101 (БЖД в техносфере)
(наименование специальности)

«__» _____200___ г., протокол № _____

Председатель _____ О.Т. Аксенова
(подпись, И.О.Ф.)

Рабочая программа переутверждена на заседании кафедры от _____ протокол № _____ .

Зав.кафедрой _____ _____
подпись Ф.И.О.

СОГЛАСОВАНО
Начальник УМУ
_____ Г.Н. Торопчина
(подпись, И.О.Ф.)

«__» _____200___ г.

СОГЛАСОВАНО
Председатель УМС факультета
_____
(подпись, И.О.Ф.)

«__» _____200___ г.

СОГЛАСОВАНО
Заведующий выпускающей кафедрой
_____
(подпись, И.О.Ф.)

«__» _____200___ г.

1. Цели и задачи дисциплины, ее место в учебном процессе

1.1. Цель преподавания дисциплины

Основной целью дисциплины «Программное обеспечение задач БЖД» является сформировать представление о назначении и видах системного программного обеспечения работоспособности компьютеров, применения прикладного программного обеспечения для решения задач в области безопасности жизнедеятельности, познакомить студентов с инструментарием технологии программирования.

1.2 Задачи изучения дисциплины

В результате изучения дисциплины студенты должны знать основные понятия и термины программного обеспечения, характерные особенности программного продукта, методы защиты программных продуктов, классификацию программных продуктов, характерные особенности системного программного обеспечения, инструментария технологии программирования и пакетов прикладных программ, уметь выбирать и комбинировать программно-аппаратные средства для наиболее эффективного решения поставленных задач по природопользованию, наглядно интерпретировать полученные результаты с использованием современных программных средств общего назначения, включая средства мультимедиа.

2.Содержание дисциплины

2.1. Наименование тем, их содержание, объем в часах лекционных занятий

Дисциплина «Программное обеспечение задач БЖД» является базовой дисциплиной для изучения остальных связанных с информационными системами, информационными технологиями и базами данных в частности дисциплины «Информационные технологии в управлении безопасностью жизнедеятельности» которая является федеральным компонентом. Дисциплина введена по решению совета ВУЗа, входит в блок ОПД.В и является дисциплиной по выбору.

Тема 1. **.** *Программные продукты и их основные характеристики (4 часа)*

Основные понятия программного обеспечения. Категории специалистов, занятых разработкой и эксплуатацией программ. Характеристика программного продукта. Защита программных продуктов: основные понятия о защите программных продуктов; программные системы защиты от несанкционированного копирования. Правовые методы защиты программных продуктов: патентная защита, лицензионные соглашения и др.

Тема 2 Классификация программных продуктов (2 часа)

Классы программных продуктов. Системное программное обеспечение: назначение и состав системного ПО. Операционные системы клиент-сервер. Интерфейсы и оболочки.

Тема 3. Основы теории баз данных (6 часов)

Основные понятия теории баз данных. Классификация баз данных. Структурные элементы базы данных. Модели данных. Системы управления базами данных. СУБД в многопользовательских системах. Технология использования СУБД. Обзор СУБД. Проектирование баз данных. Понятие предметной области.

Тема 4. Проектирование баз данных (4 часа)

Технология анализа предметной области. Логическое проектирование. СУБД MS Access. Связи между таблицами. Ключи, макросы модули и объекты. Запросы и выборки. Формы данных. Обработка данных с помощью отчетов.

Тема 5. Стандартные приложения служебного назначения операционной системы Windows (2 часа)

Обзор стандартных приложений служебного назначения. Очистка диска. Дефрагментация диска. Уплотнение диска. Проверка диска. Системный монитор.

Тема 6. Инструментарий технологии программировани (2 часа)

Состав и назначение инструментария технологии программирования. Средства для создания приложений: локальные средства разработки программ; интегрированные среды разработки программ. CASE- технологии создания информационных систем. Программные продукты для создания приложений.

Тема 7. Пакеты прикладных программ (4 часа)

Характеристика пакетов прикладных программ. Проблемно-ориентированные ППП. ППП автоматизированного проектирования. ППП общего назначения. Методо-ориентированные ППП. Офисные ППП. Настольные издательские системы. Программные средства мультимедиа.

Тема 8. Текстовый процессор – прикладной программный продукт (4 часа)

Работа с текстом: минимальный набор типовых операций – операции, производимые над абзацами документа, операции, производимые с фрагментами текста. Расширенный набор типовых операций: контекстный поиск и замена, операции сохранения, проверка правописания, использование шаблонов, автотекст, слияние документов. Работа издательских систем: особенности издательских систем, основы создания документа в издательских системах.

Тема 9. Универсальные пакеты прикладных программ для обработки данных (2 часа)

Электронные таблицы (Excel) – основные понятия, функциональные возможности, технология работы. Пакеты для инженерных и научных расчетов (Mathcad) – основные понятия, технология работы. Примеры экологических расчетов и задач охраны безопасности труда.

Тема 10. Сервисное программное обеспечение (2 часа)

Программы архиваторы: понятие процесса архивации файлов, основные виды программ архиваторов. Помещение файлов в архив. Извлечение файлов из архива. Удаление

файлов из архива. Многофункциональные интегрированные архиваторы RAR и ZIP.

Тема 11. Антивирусные программные средства (2часа)

Сущность и проявление компьютерных вирусов. Жизненный цикл вируса. Объекты воздействия и деструктивные функции вирусов. Классификация компьютерных вирусов. Основные виды вирусов. Программы обнаружения и защиты от вирусов. Проблемы защиты от макровирусов. Сетевые вирусы. Основные меры по защите от вирусов.

Тема 12. Права доступа в сети (2час)

Основные положения. Мое сетевое окружение. Как найти в сети другие компьютеры. Как найти файлы и папки на других компьютерах. Как получить доступ. Как предоставить другим пользователям доступ к вашим файлам. Выдача разрешений на доступ к сети.

2.2. Лабораторные занятия, их наименование и объем в часах

1. Создание презентации – 2 часа.

Разработка плана презентации. Знакомство с меню. Использование мастера автосодержания. Применение шаблонов. Выбор структуры слайда. Ввод тегста и его форматирование. размещение объектов на слайде: рисунок, диаграммы, таблицы. Привязка гиперссылки к объекту. Сохранение результатов работы. Оформление по шаблону. Внесение изменений в оформление слайдов. Выбор способа перехода от слайда к слайду. Выбор встроенной анимации.

2. Изучение табличного процессора MS Excel – 2 часа.

Задание и определение типов данных. Автозаполнение. Применение и копирование формул. Создание диаграмм. Работа с базами данных.

3. Изучение СУБД MS Access – 4 часа.

Создание таблиц. Ключи. Связи между таблицами. Макросы модули и объекты. Запросы, формы, отчеты.

4. Основы создания Web-страницы – 2 часа.

Ознакомление с языком HTML -основой WWW. Создание Web- страницы с использованием основных тегов ввода и форматирования текста и др. информации.

5. Основы работы с Mathcad– 4 часа.

Определения и переменные. Ввод текста. Повторяющиеся вычисления. Области и меню. Графики. Определение переменных и функций. Глобальные определения. Вычисление выражений. Копирование числовых результатов. Управление вычислениями. Комплексные числа. Создание вектора или матрицы. Вычисления с массивами. Векторные и матричные операторы. Векторные и мат-

ричные функции. Создание графика и его форматирование. Полярные графики. Графики поверхностей. Трехмерные гистограммы. Анимация.

6. Вычисления в Mathcad – 2 часа.
Вставка встроенных функций. Усечение и функции округления. Численное решение нелинейного уравнения. Решение систем уравнений. Решение матричных уравнений. Символьное решение уравнений. Символьные операции.

7. Переменные и символьные вычисления в Mathcad – 2 часа.
Операторы вычисления сумм и произведений. Производные. Интегралы.
Имена. Предопределенные переменные. Числа. Вычисления с единицами измерений. Стиль представления результатов вычислений. Задание операторов пользователя.

2.3. Самостоятельная работа студентов

1. Средства управления печатью в Windws.
2. Средства связи в Windows.
3. Основные характеристик операционной системы XP.
4. Программы для обработки и создания изображений (Photoshop и CorelDraw).
5. Демонстрация презентации: контроль времени выступления, заметки к слайду, проведение презентации на чужом компьютере, скрытые слайды. Вывод презентации на печать.
6. Мультимедийные Web-страницы.
7. Интерактивные Web.

2.4. Вопросы к зачету

1. Программные продукты и их основные характеристики.
2. Защита программных продуктов. Правовые методы защиты программных продуктов.
3. Классификация программных продуктов.
4. Базовое программное обеспечение.
5. Основные понятия теории баз данных. Классификация баз данных.
6. Структурные элементы базы данных.
7. Модели данных.
8. Системы управления базами данных.
9. СУБД в многопользовательских системах.
10. Технология использования СУБД.
11. Обзор СУБД.
12. Проектирование баз данных.
13. Понятие предметной области.
14. Технология анализа предметной области.
15. Логическое проектирование.
16. СУБД MS Access. Связи между таблицами. Ключи, макросы модули и объекты.
17. Инструментарий технологии программирования.
18. Проблемно-ориентированные ППП.
19. ППП автоматизированного проектирования.

20. ППП общего назначения.
21. Методо-ориентированные ППП.
22. Офисные ППП.
23. Настольные издательские системы.
24. Работа с текстом: минимальный набор типовых операций.
25. Расширенный набор типовых операций.
26. Основы создания документа в издательских системах.
27. Электронные таблицы (Excel).
28. Пакеты для инженерных и научных расчетов (Mathcad).
29. Программы архиваторы.
30. Сущность и проявление компьютерных вирусов.
31. Жизненный цикл вируса.
32. Классификация компьютерных вирусов.
33. Программы обнаружения и защиты от вирусов.
34. Основные меры по защите от вирусов.
35. Права доступа в сети.
36. Программа PowerPoint.
37. Создание Web-сайта.

3. Учебно-методические материалы по дисциплине

3.1. Список рекомендуемой литературы

Основная

1. Джон Крейнак, Джой Хебрейкен. Энциклопедия. Интернет. – Санкт-Петербург, 1999. – 865с.
2. Энди Шафран. Создание Web-страниц: Самоучитель (+CD). – СПб.: Питер, 2000. – 485с.
3. Ратбон Э. Windows 2000 Professional. – М.:, Вильямс. – 2004. – 374с.
4. Сагман С.В. . Microsoft Windows XP. – М.: ЭКОМ, – 2002. – 416с.

Дополнительная

1. Охара Ш. Microsoft Windows XP. – М.:, Эксмо-Пресс. – 2005. – 396с.
2. Виллет Э. Officce XP. Библия пользователя. – М.: Вильямс, – 2000. – 346с.

4. Учебно-методическая (технологическая) карта дисциплины.

Номер темы	Темы лекционных занятий	Количество часов лекционных занятий	Темы лабораторных работ	Количество часов лабораторных занятий	Самостоятельная работа студентов		Формы контроля
					содержание	часы	
1	Программные продукты и их основные характеристики	4	Создание презентации	2	Изучить порядок выполнения лабораторной работы	1	Текущий контроль
2	Классификация программных продуктов	2	Изучение табличного процессора MS Excel	2	Изучить порядок выполнения лабораторной работы	1	Текущий контроль
3	Основы теории баз данных	6	Изучение СУБД MS Access	2	Изучить порядок выполнения лабораторной работы	1	Текущий контроль
4	Проектирование баз данных	4	Основы создания Web-страницы	2	Изучить порядок выполнения лабораторной работы	1	Текущий контроль
5	Стандартные приложения служебного назначения операционной системы Windows	2	Стилевое оформление текста на Web-странице	2	Изучить порядок выполнения лабораторной работы	1	Текущий контроль
6	Инструментарий технологии программирования	2	Списки и таблицы	2	Изучить порядок выполнения лабораторной работы	1	Текущий контроль
7	Пакеты прикладных программ	4	Использование графики в Web-страницах	2	Изучить порядок выполнения лабораторной работы	1	Текущий контроль
8	Текстовый процессор – прикладной программный продукт	4	Ссылки на Web-страницах	2	Изучить порядок выполнения лабораторной работы	1	Текущий контроль
9	Универсальные пакеты прикладных программ для обработки данных	2	Создание фреймов	2	Изучить порядок выполнения лабораторной работы	1	Текущий контроль
10	Сервисное программное обеспечение	2	-	-	-		-
11	Антивирусные программные средства	2	-	-	-	-	-

Номер темы	Темы лекционных занятий	Количество часов лекционных занятий	Темы лабораторных работ	Количество часов лабораторных занятий	Самостоятельная работа студентов		Формы контроля
					содержание	часы	
1	Программные продукты и их основные характеристики	4	Создание презентации	2	Изучить порядок выполнения лабораторной работы	1	Текущий контроль
12	Права доступа в сети	2	-	-	-	-	-

5. Основные критерии оценки знаний студентов по дисциплине “Программное обеспечение задач БЖД”

Студенты обязаны сдать зачет в строгом соответствии с учебным планом, а также утвержденным программам, едиными для всех форм обучения.

Зачет по дисциплине «Программное обеспечение задач БЖД» служит формой контроля усвоения дисциплины в целом.

К зачету допускаются студенты, выполнившие и защитившие в полном объеме лабораторные работы.

Сроки проведения зачета устанавливаются графиком учебного процесса, утвержденным проректором по учебной работе.

Знания, умения и навыки обучающегося определяются оценками «зачтено» и «не зачтено». Критерии приведены в таблице.

Основные критерии оценки знаний студентов

Оценка	Полнота, системность, прочность знаний	Обобщенность знаний

«зачтено»	Изложение полученных знаний в устной, письменной или графической форме, полное, в системе, в соответствии с требованиями учебной программы; допускаются единичные несущественные ошибки, самостоятельно исправляемые студентами	Выделение существенных признаков изученного с помощью операций анализа и синтеза; выявление причинно-следственных связей; формулировка выводов и обобщений; свободное оперирование известными фактами и сведениями с использованием сведений из других предметов
	Изложение полученных знаний в устной, письменной и графической форме, полное, в системе, в соответствии с требованиями учебной программы; допускаются отдельные несущественные ошибки, исправляемые студентами после указания преподавателя на них	Выделение существенных признаков изученного с помощью операций анализа и синтеза; выявлений причинно-следственных связей; формулировка выводов и обобщений, в которых могут быть отдельные несущественные ошибки; подтверждение изученного известными фактами и сведениями
	Изложение полученных знаний неполное, однако это не препятствует усвоению последующего программного материала; допускаются отдельные существенные ошибки, исправленные с помощью преподавателя	Затруднения при выполнении существенных признаков изученного, при выявлении причинно-следственных связей и формулировке выводов
«не зачтено»	Изложение учебного материала неполное, бессистемное, что препятствует усвоению последующей учебной информации; существенные ошибки, неисправляемые даже с помощью преподавателя	Бессистемное выделение случайных признаков изученного; неумение производить простейшие операции анализа и синтеза; делать обобщения, выводы

2. Перечень вопросов для проведения зачета со студентами заочной сокращенной формы обучения

1. Программные продукты и их основные характеристики.
2. Защита программных продуктов. Правовые методы защиты программных продуктов.
3. Классификация программных продуктов.
4. Базовое программное обеспечение.
5. Инструментарий технологии программирования.
6. Проблемно-ориентированные ППП.
7. ППП автоматизированного проектирования.
8. ППП общего назначения.
9. Методо-ориентированные ППП.
10. Офисные ППП.
11. Электронные таблицы (Excel).

12. Пакеты для инженерных и научных расчетов (Mathcad).
13. Программы архиваторы.
14. Сущность и проявление компьютерных вирусов.
15. Жизненный цикл вируса.
16. Классификация компьютерных вирусов.
17. Программы обнаружения и защиты от вирусов.
18. Основные меры по защите от вирусов.
19. Права доступа в сети.
20. Программа PowerPoint.
21. Создание Web-сайта.

3. Требования и порядок сдачи зачета по дисциплине

Зачет сдается в период зачетной недели экзаменационной сессии. Зачет проводится в объеме программы учебной дисциплины. Форма сдачи зачета – устная. При устной форме зачета преподавателю предоставляется право задавать студенту по программе курса дополнительные вопросы, а также помимо теоретических вопросов, давать практические задания по программе данного курса.

Преподаватель на зачете учитывает не только ответы на вопросы, но наличие всех выполненных лабораторных работ и посещаемость студента в семестре в случае пропуска лекций выполняется проверочная работа по темам пропущенных лекций. Отвечая на зачете, студент должен дать развернутый ответ. При этом показать знание теории и продемонстрировать свободную ориентацию в указанном материале, знание понятий и терминологии, ответить на уточняющие вопросы.

4. Методические рекомендации для проведения самостоятельной работы студентов

по дисциплине «Программное обеспечение задач БЖД**»**

Самостоятельная работа студентов при изучении «Программное обеспечение задач БЖД» включает выполнение контрольной работы по следующим темам и вопросам

Тема: Операционная система Windows

1Средства управления печатью в Windws.
2Средства связи в Windows.
3Основные характеристик операционной системы XP.

Литература

1. Ратбон Э. Windows 2000 Professional. – М.:, Вильямс. – 2004. – 374с.
2. Сагман С.В. . Microsoft Windows XP. – М.: ЭКОМ, – 2002. – 416с.

Тема: Средства мультимедиа

Вопросы

1. Программы для обработки и создания изображений (Photoshop и Corel-Draw).
2. Демонстрация презентации: контроль времени выступления, заметки к слайду, проведение презентации на чужом компьютере, скрытые слайды. Вывод презентации на печать.
3. Мультимедийные Web-страницы.
4. Интерактивные Web.

Литература

1. Джон Крейнак, Джой Хебрейкен. Энциклопедия. Интернет. – Санкт-Петербург, 1999. – 865с.
2. Энди Шафран. Создание Web-страниц: Самоучитель (+CD). – СПб.: Питер, 2000. – 485с.

5. Содержание курса лекций по дисциплине «Программное обеспечение задач БЖД»

Тема 1. Программные продукты и их основные характеристики (4 часа)

Лекция №1

План:
1) Основные понятия программного обеспечения.
2) Категории специалистов, занятых разработкой и эксплуатацией программ.
3) Характеристика программного продукта

1) Понятие информационной технологии

Технология в переводе с греческого означает искусство, мастерство, умение. Под процессом понимается определенная совокупность действий, направленных на достижение поставленной цели. Процесс должен определяться выбранной человеком стратегией и реализовываться с помощью набора различных средств и методов.

Информационные технологии – процесс, использующий совокупность средств и методов сбора, обработки и передачи данных для получения информации нового качества о состоянии объекта, процесса или явления.

Информационная система – взаимосвязанная совокупность средств, методов и персонала, используемая для хранения, обработки и выдачи информации, необходимой в процессе принятия решений задач из любой области.

Цель информационной технологии – производство информации для ее анализа человеком м принятия на его основе решения по выполнению какого либо действия.

Основными составляющими информационных технологий являются:

- сбор данных или первичной информации;
- обработка данных и получение результатной информации;
- передача результатной информации пользователю для принятия на ее основе решений.

Системное обеспечение состоит из операционных систем и средств контроля и диагностики.

Операционная система (ОС) составляет основу программного обеспечения ПК. Операционная система – это совокупность программ, обеспечивающих управление процессом обработки информации и взаимодействие между аппаратными средствами и пользователем.

В состав ОС включаются также системные программы: программы-оболочки и утилиты.

Программы-оболочки – программы, обеспечивающие более удобный и наглядный способ общения пользователя с ПК, чем штатные средства ОС.

2) Категории специалистов занятых разработкой и эксплуатацией программ

Основное назначение автоматизированных мест работников – обеспечить условия для комфортной, высокопроизводительной и качественной работы. Создание АРМ предполагает, что основные операции по накоплению, хранению и переработке информации выполняет вычислительная техника. Пользователь же контролирует ее действия, меняет значение отдельных параметров, вводит исходные данные для решения задач и функций управления.

Автоматизированное рабочее место (АРМ) – это совокупность аппаратных, программных, методических и языковых средств, обеспечивающих автоматизацию функций пользователя в некоторой предметной области и позволяющих оперативно удовлетворять его информационные и вычислительные потребности.

Функционирование любого АРМ предполагает наличие следующих видлв обеспечения:

- технического;
- информационного;
- математического;
- программного.

В самом общем случае пользователи различаются по служебному положению, специальностям, уровню освоения и частоте работы с вычислительной техникой, виду потребляемых данных и т. д.

По служебному положению пользователи в сфере организационного управления могут быть разделены на 3 категории:

- руководители;
- персонал руководителей;
- обслуживающий персонал.

Как специалисты пользователи могут быть разбиты на группы, например, бухгалтеров, экономистов, статистиков, кадровиков и др.

По степени подготовленности к работе с вычислительной техникой выделяются пользователи, имеющие навыки в программировании, имеющие подготовку в используемого программном обеспечении АРМ.

Пользователи АРМ могут быть разделены на 2 группы в зависимости от периода получения данных:

- данные нужны в процессе их обработки и оформления;
- нужны законченные сведения о состоянии объектов.

Для первой группы пользователей АРМ необходим интерактивный режим работы, для пользователей второй группы он необязателен.

В зависимости от категории пользователей АРМ должно обеспечивать не только обработку данных, но и выдачу комментариев, подсказок. Для разных категорий пользователей также необходимы различные виды представления данных. Например, обслуживающий персонал решает повторяющиеся задачи,

использует внутренние данные конкретной организации, которые имеют короткий активный период существования.

Руководителям нужны как внутренние, так и внешние данные. Для такого типа пользователей необходимы АРМ, реализующие какую-либо цель управления или принятия решения.

3) Характеристики программного продукта

Современный рынок программных средств предлагает большое количество продуктов, для оценивания качества которых разрабатываются системы подходящих характеристик. Сложность решения задач оценки качества и выбора программных средств объясняется рядом объективных и субъективных причин, подробно рассмотренных в [1, 2].

Рассмотрим следующие характеристики качества программных средств: модифицируемость, изучаемость и завершенность.

Модифицируемость – это характеристика программных средств, которая упрощает внесение в него необходимых изменений и доработок и включает в себя характеристики расширяемости, структурированности и модульности.

Изучаемость - это характеристика, которая позволяет минимизировать усилия по изучению и пониманию программ и документации программных средств и включает в себя характеристики информативности, понятности, структурированности и удобочитаемости.

Завершенность характеризует степень обладания программных средств всеми необходимыми частями и чертами, требующимися для выполнения своих явных и неявных функций.

Эргономические требования, предъявляемые к ПСУН делятся на следующие группы:

- требования соответствия характеристик программного средства методи-
- ческому руководству и сопутствующей документации;
-
- требования к временным режимам работы программного средства;
-
- требования к общей визуальной среде на экране монитора;
-
- требования к цветовым характеристикам;

- требования к пространственному размещению информации на экране монитора;
- требования к организации диалога;
- требования к буквенно-цифровой символике и знакам;
- требования к звуковому сопровождению.

Требования соответствия характеристик программного средства методическому руководству и сопутствующей документации:

- соответствие программного средства методическому материалу;
- соответствие последовательности действий, необходимых для установки программного средства, инструкции;
- легкость запуска;
- соответствие основных технических характеристик программного средства параметрам, приведенным в документации;
- устойчивость работы.

Лекция №2

План:

1) Защита программных продуктов: основные понятия о защите программных продуктов; программные системы защиты от несанкционированного копирования.

2) Правовые методы защиты программных продуктов: патентная защита, лицензионные соглашения и др.

1) Защита программных продуктов: основные понятия о защите программных продуктов; программные системы защиты от несанкционированного копирования.

В соответствии с принятой классификацией выделяют шесть направлений деятельности по защите информации [7]:

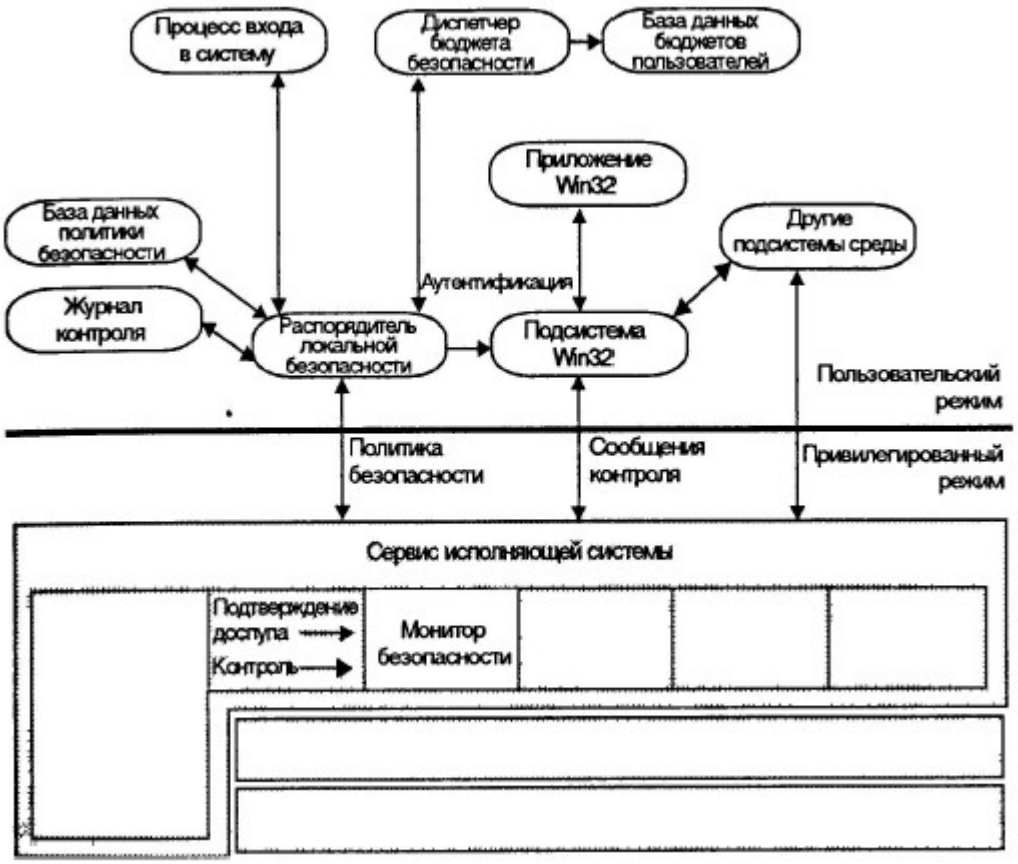
1. Защита от несанкционированного доступа в автоматизированных информационных системах, имеющая как программную, так и аппаратную реализацию.
2. Защита информации при передаче по каналам связи и в средствах их коммутации (в том числе в локальных (ЛВС) и распределённых вычислительных сетях).
3. Защита юридической значимости так называемых "электронных" документов (в системе электронной почты (e-mail) и др.)
4. Защита от утечки информации в виде побочных электромагнитных излучений и наводок (ПЭМИН).
5. Защита от программных средств скрытого информационного воздействия (частным случаем которых являются компьютерные вирусы).
6. Защита от несанкционированного копирования и распространения программ и баз данных. В настоящее время в связи со сложным характером взаимоотношений на рынке программных продуктов проблема защиты от несанкционированного копирования является одной из наиболее острых в области разработки программных средств.

Безопасность и отказоустойчивость

Безопасность для Windows NT является частью требований к ОС. Модуль безопасности включает компоненты для управления доступом к объектам (например, к файлам или разделяемым принтерам), для определения того, кем осуществляются действия над объектом, и для назначения списка протоколируемых событий.

Модель безопасности Windows NT

На рисунке представлена модель безопасности Windows NT, которая включает следующие компоненты:



1. *Процессы входа в систему* (Logon Processes), которые принимают запросы пользователей на вход в систему. Они включают начальный интерактивный вход в систему.

2. *Распорядитель локальной безопасности* (Local Security Authority), который гарантирует, что пользователь имеет разрешение на обращение к системе. Этот компонент является основой подсистемы безопасности Windows NT. Он генерирует маркеры доступа (описаны ниже), управляет политикой локальной безопасности и обеспечивает интерактивный сервис аутентификации пользователя. Распорядитель локальной безопасности также управляет политикой контроля и регистрирует контрольные сообщения, сгенерированные монитором безопасности.

3. Диспетчер бюджета безопасности (SAM - Security Account Manager), который поддерживает базу данных бюджетов пользователя. Эта база данных содержит информацию по бюджетам всех пользователей и групп. SAM обеспечивает аутентификацию пользователя, которая используется распорядителем локальной безопасности.

4. *Монитор безопасности* (Security Reference Monitor), который проверяет, имеет ли пользователь права доступа к объекту, и отслеживает любое предпринимаемое пользователем действие. Обеспечивает сервис и привилегированному и пользовательскому режимам, который гарантирует осуществление доступа к объектам только пользователям и процессам, имеющим необходимые разрешения.

Вместе эти компоненты известны как подсистема безопасности, которая является интегральной подсистемой, так как воздействует на операционную систему Windows NT в целом.

Модель безопасности Windows NT разработана в соответствии с уровнем С2, определенным Министерством обороны США.

Наиболее важные требования r уровн. безопасности перечислены ниже:

- владелец ресурса (например, файла) должен иметь возможность управлять доступом к ресурсу;
- операционная система должна защищать объекты от несанкционированного использования другими процессами. Например, система должна защищать память так, чтобы ее содержимое не могло читаться после освобождения процессом, и после удаления файла не допускать обращения к данным файла;
- перед получением доступа к системе каждый пользователь должен идентифицировать себя, вводя уникальное имя входа в систему и **пароль**
- администратор системы должен иметь возможность контроля связанных с безопасностью событий (audit security-related events).

Доступ к этим контрольным данным должен быть ограничен администратором.

- система должна защищать себя от внешнего вмешательства типа модификации выполняющейся системы или хранимых на диске системных файлов.

Структура системы защиты от несанкционированного копирования.

В общем случае система защиты от несанкционированного копирования представляет собой комплекс средств, предназначенный для затруднения (в идеале - предотвращения) нелегального копирования (исполнения) защищаемого программного модуля, с которым она ассоциирована.

Подсистема внедрения управляющих механизмов представляет собой комплекс программных средств, предназначенный для подключения внедряемого защитного кода к защищаемому программному модулю. Внедряемый защитный код - это программный модуль, задача которого состоит в противодействии попыткам запуска (исполнения) нелегальной копии защищаемой программы.

Подсистема реализации защитных функций представляет собой программную секцию, решающую задачу распознавания легальности запуска защищаемой программы.

Системы защиты от несанкционированного копирования можно классифицировать по способу внедрения защитного механизма:

- встроенная (внедряется при создании программного продукта);
- пристыковочная (подключается к уже готовому программному продукту).

2) Правовые методы защиты программных продуктов: патентная защита, лицензионные соглашения и др.

Правовое регулирование на информационном рынке. Развитие рыночных отношений в информационной деятельности поставило вопрос о защите информации как объекта интеллектуальной собственности и имущественных прав на нее. Ряд указов, постановлений, законов в Российской Федерации о защите программных продуктов:

- "Об информации, информатизации и защите информации".
- "Об авторском праве и смежных правах".
- "О правовой охране программ для ЭВМ и баз данных".
- "О правовой охране типологий интегральных схем".

Базовым юридическим документом является закон "Об информации, информатизации и защите информации". С его помощью удалось частично решить вопросы правового регулирования на информационном рынке; защиты прав и свобод личности от угроз и ущерба, связанных с искажением, порчей, уничтожением "персональной" информации.

В законе определены цели и основные направления государственной политики в сфере информатизации. Информатизация определяется как важное новое стратегическое направление деятельности государства. Указано, что государство должно заниматься формированием и реализацией единой государственной научно-технической и промышленной политики в сфере информатизации.

Закон создает условия для включения России в международный информационный обмен, предотвращает бесхозяйственное отношение к информационным ресурсам и информатизации, обеспечивает информационную безопасность и права юридических и физических лиц на информацию. В законе определяются комплексные проблемы организации информационных ресурсов, правовое положение по их использованию и предлагается рассматривать информационные ресурсы в двух аспектах:

- как материальный продукт, который можно покупать и продавать;
- как интеллектуальный продукт, на который распространяется право интеллектуальной собственности, авторское право.

Закон закладывает юридические основы гарантий прав граждан на информацию. Он направлен на урегулирование важнейшего вопроса экономической реформы - формы, права и механизма реализации собственности на накопление информационных ресурсов и технологические достижения. Обеспечена защита собственности в сфере информационных систем и технологий, что способствует формированию цивилизованного рынка информационных ресурсов, услуг, систем, технологий, средств их обеспечения.

Тема 2. Классификация программных продуктов (2 часа)

Лекция №3

План:

1) Системное программное обеспечение: назначение и состав системного программного обеспечения.

2) Операционные системы клиент-сервер. Интерфейсы и оболочки.

Системное программное обеспечение: назначение и состав системного программного обеспечения.

НАЗНАЧЕНИЕ И СОСТАВ СИСТЕМНОГО ПО

Системное программное обеспечение - это все программные средства, необходимые для реализации прикладной системы (автоматизированной информационной системы - АИС, или отдельных прикладных программ), удовлетворяющей требованиям конечных пользователей.

Под системным программированием понимается работа, выполняемая системными программистами, т.е. создание системного программного обеспечения. *Граница между системным программированием и прикладным программированием зависит от обстоятельств.* Например, с точки зрения программиста, который занимается ядром операционной системы, человек, создающий компилятор, является пользователем системы (т. е. прикладным программистом), хотя этот же человек будет, вероятно, рассматриваться как системный программист тем лицом, которое занимается написанием программ для отыскания минимума функции. Вместе с тем, человек, который будет использовать программу минимизации, *может считать всех трех вышеупомянутых программистов системными программистами.*

В данном курсе под **системным программистом** будет пониматься человек, который специализируется по системному программированию и программному обеспечению нижнего уровня, т.е. операционным системам, компиляторам, системам связи и системам управления базами данных и др.

Развитие системного программного обеспечения происходило параллельно с развитием аппаратных средств вычислительной техники и связи.

В середине 50-х гг. в основном разрабатывались прикладные программы, реализующие функции, для которых были необходимы возможности компьютеров. Производители технических средств изготавливали компьютеры, а пользователи разрабатывали программы с помощью операционных систем и ассемблеров, находящихся в самом начале своего развития. Режим выполнения про-

грамм зависел не от примитивных операционных систем (ОС), а от набора команд и архитектуры ЭВМ. В конце 50-х - начале 60-х гг. начали появляться управляющие системы, ориентированные на определенные типы ЭВМ. Эти программные средства, вначале состоявшие из программ ввода-вывода, являлись ресурсом коллективного пользования. Они управляли вводом-выводом на перфокарты, печатающие устройства, магнитные ленты, а к концу этого периода - на устройства памяти прямого доступа (магнитные барабаны и диски).

Дополнительно появились языки высокого уровня, такие, как Фортран и Кобол, которые существенно упростили разработку программ. В середине - конце 60-х гг. резко возросла значимость операционных систем. Это произошло в связи с появлением новых возможностей, таких, как большая оперативная память, и увеличением сложности оборудования. В результате появился класс программных систем, включающих языки высокого уровня и ряд других средств. В 70-х гг. стали возможными интерактивный доступ к вычислительным ресурсам с использованием терминалов, а также средства межмашинной коммуникации. Как следствие, операционные системы были пополнены соответствующими функциями. Это, в свою очередь, позволило решить проблемы создания распределенных систем реального времени для коммерческих приложений. Ведущими технологиями этого, второго, этапа были обработка баз данных и передача информации. Эти средства совместно со средствами разработки интерактивных систем предоставили возможности управления компьютерными системами в режиме ON-LINE (неавтономный режим работы пользователя в сетях ЭВМ).

Были созданы надлежащие условия для эффективной разработки прикладных систем. Имеются в виду средства управления данными, представленными в реляционной модели, генераторы прикладных программ, средства управления дисплеями и пр. В результате прикладные программы были отделены от операционных систем и оборудования. При использовании этой новой, *ориентированной на приложения технологии основные силы тратятся на решение*

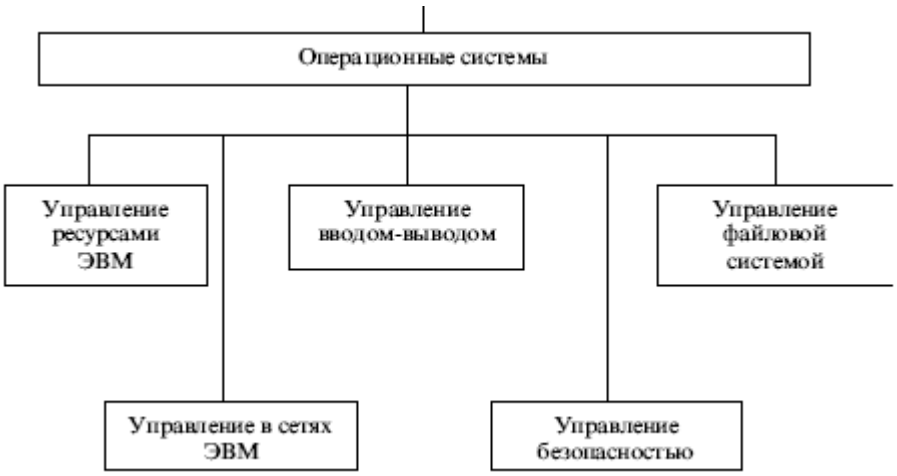
проблемы, а не на реализацию в среде конкретной ЭВМ или операционной системы.

Описываемые изменения затронули большинство семейств ЭВМ. В средине 70-х гг. каждое семейство компьютеров с соответствующими операционной и прикладными системами было ориентировано на конкретные приложения с учетом таких критериев, как производительность и среда применения (имеются в виду пакетный и интерактивный режимы или режим обработки транзакций). Каждый продукт разрабатывался так, чтобы оказаться лучшим среди других подобных продуктов. Такая стратегия оказалась успешной применительно к условиям рыночной конкуренции.

В 80-х гг. сильно возросла мощность всего ряда ЭВМ (от ПЭВМ до самых больших ЭВМ). Чтобы устоять в условиях сильной конкуренции, были добавлены новые возможности. Сегодня во всех системах мощность оборудования и программные средства достигли в своем развитии такой точки, когда прикладные системы должны обеспечивать только функциональный уровень и практически не зависят от аппаратных средств, хотя остаются частично зависимыми от архитектуры операционной системы, под управлением которой они выполняются.

В последующий период времени большое влияние на развитие системного ПО оказали средства организации сетей (Systems Network Architecture - SNA), которые позволили достаточно эффективно управлять большими вычислительными сетями. Интерактивные средства управления вычислениями, объединенные с возможностями построения вычислительных сетей, обеспечили мощные средства управления автоматизированными системами в режиме реального времени.

В целом, структуру системного ПО, сложившуюся к настоящему времени, можно изобразить схемой, представленной на рис. 1.



Структура современного ПО

Эта модель программного обеспечения может рассматриваться как организационное средство производства прикладных систем. С помощью модели не только проще классифицировать программные продукты, разработанные в среде разных операционных систем, но и обеспечить базис для определения общих для всех ОС интерфейсов, которые помогут унифицировать разработку приложений.

Операционные системы

Введение в операционные системы

Операционные системы осуществляют управление системными ресурсами с целью эффективного, безопасного и надежного выполнения прикладных программ. В основе любой ОС лежит ядро операционной системы, управляющей процессами операционной системы и распределяющей для них системные ресурсы. Ядро ОС располагается постоянно в оперативной памяти ЭВМ.

При включении питания ЭВМ системный загрузчик (компонента управляющей программы) осуществляет начальную загрузку ядра операционной системы и его инициализацию, после чего все управление системными процессами осуществляет ОС. Взаимодействие между ОС и всеми остальными программно-техническими компонентами АИС осуществляется через систему прерываний - комплекс технических и программных средств, обеспечивающих возможность прерывать выполнение программы при поступлении внешних или внутренних сигналов прерывания и продолжить выполнение пре-

рванной программы от точки прерывания после обработки сигналов прерывания.

ОС состоит из набора системных программ, входящих в состав операционной системы и выполняющих управляющие и обслуживающие функции. Системные программы размещаются в системной библиотеке, содержащей определенные компоненты операционной системы. Каждый ресурс имеет уникальное имя, идентифицирующее его в данной операционной системе. Все события, происходящие в управляемой ОС системе, регистрируются в системном журнале (специальном наборе данных).

Известны следующие виды ОС:

- локальные операционные системы, осуществляющие управление ресурсами в пределах конкретного вычислительного комплекса, без использования каналов связи;

- сетевые операционные системы, осуществляющие управление ресурсами в локальных и глобальных сетях передачи данных.

Кроме того, могут использоваться специальные ОС, например операционные системы автоматизированного проектирования и операционные системы реального времени. Система реального времени - это система, осуществляющая информационный обмен с другими системами, периферийными устройствами или датчиками при таких временных характеристиках, которые позволяют немедленно обрабатывать всю поступающую информацию и информацию для вывода. Эти системы обычно используются в управлении техническими объектами (АСУТП).

Значительная доля современных автоматизированных систем имеет распределенный характер, то есть их компоненты распределены по различным вычислительным комплексам, которые связаны между собой каналами передачи данных - средствами двухстороннего обмена данными, представляющими собой совокупность аппаратуры окончания канала данных и линии передачи данных. Используется также термин *передающая среда* - совокупность линии передачи данных и, возможно, коммутаторов данных, концентраторов данных, повтори-

телей и другого оборудования, не относящегося к станции данных, организация структуры и функционирование которой обеспечивают физическую передачу данных между станциями данных. Распределенные системы базируются на сетях ЭВМ - совокупности сети передачи данных, взаимосвязанных сетью ЭВМ и необходимых для реализации этой связи программного обеспечения и технических средств, которая предназначена для организации распределенной обработки информации.

Логическая структура и принципы работы информационной сети называются архитектурой сети, для описания которой используются следующие основные понятия:

передача данных - пересылка данных при помощи средств связи из одного места для приема их в другом;

функциональный блок сети - система или устройство, выполняющие определенную, логически связанную группу функций;

источник данных - функциональный блок, порождающий данные для передачи;

приемник данных - функциональный блок, принимающий переданные данные;

протокол - набор семантических и синтаксических правил, определяющий поведение функциональных блоков при передаче данных;

обмен данными - передача данных в соответствии с установленным протоколом;

соединение - связь, устанавливаемая между функциональными блоками для передачи информации;

ресурсы сети - программное, техническое, информационное и организационное обеспечение информационной сети, предназначенное для решения прикладных задач.

В настоящее время сетевые программные системы создаются на базе концепции открытых систем. Открытая система - это совокупность систем, взаимодействие которых подчиняется требованиям стандартов Международной организации по стандартизации. Совокупность технических и (или) программных

средств, входящая в состав открытой системы, реализующая согласованные по горизонтали и вертикали функции некоторого уровня и всех расположенных ниже уровней базовой эталонной модели и выполняющая функции поставщика сервиса для компонентов сети, называется *службой взаимосвязи открытых систем.*

Операционные системы клиент-сервер

Разновидностью открытых систем являются системы, реализованные на базе сетей типа "intranet" - корпоративных сетей, созданных на базе технологий Интернет.

В принципе все продукты и решения, разработанные для Интернет, могут использоваться и для интрасетей. Но, тем не менее, в их реализации есть некоторые особенности, которые объясняют выделение технологии создания интрасетей в некоторое самостоятельное направление и, соответственно, появление для них некоторых специфических программных решений. Интернет представляет собой сеть из компьютеров-узлов, взаимодействие между которыми выполняется с помощью протоколов TCP/IP. На физическом уровне узлы обычно связаны в некоторые внутренние сети, поэтому Интернет фактически является иерархической структурой:" Интернет-сети-узлы". Но, с логической точки зрения, каждый узел является самостоятельной единицей, имеющей свой уникальный IP (Internet Protocol) адрес, что обеспечивает возможность общения любых двух узлов Интернет. В основе организации Интернет лежат две основные идеи.

Во-первых, взаимодействие узлов не должно зависеть от типов компьютеров, их архитектур, операционных систем и пр., а также физической реализации связи между ними. Для осуществления этого используются простые и надежные протоколы установления связи и передачи данных. Причем вопрос о том, насколько узел понимает смысл принимаемых или передаваемых данных, является его личной проблемой.

Во-вторых, Интернет должен надежно функционировать независимо от изменения своей топологии, в частности, от подключения или отключения его

узлов. Причина обеспечения такой надежности заключается в том, что этот процесс никак не контролируется. Единственное, за чем здесь осуществляется контроль - это за обеспечением уникальности IP адресов узлов и соблюдением определенного порядка в их нумерации для решения проблем маршрутизации.

Логика развития корпоративных компьютерных сетей привела к объединению локальных сетей, реализованных, условно говоря, на уровне подразделения (здания), в глобальные. Однако создание интрасетей на уровне прямого взаимодействия компьютеров через Интернет является просто невозможным. Во-первых, это может быть просто нецелесообразно для относительно небольших сетей, в которых объединение компьютеров выполняется за счет чисто внутренних линий (например, на территории завода). Во-вторых (и это самое главное), из-за необходимости обеспечения информационной защиты при взаимодействии с внешним миром - ведь открытые протоколы Интернет делают прозрачными все включенные в него узлы. Специфика программных продуктов и технологий для интрасетей заключается в том, что число узлов в них является ограниченным и изменение топологии сети и содержимого информации носит некоторый предсказуемый характер. А в этом случае организация таких процедур, как, например, маршрутизация и поиск информации может быть оптимизирована совершенно другими методами, чем в хаотичном Интернет.

При создании информационных систем на базе интрасетей необходимо решить следующие задачи:

- полностью интегрировать внутренние, локальные сети с Интернет для расширения возможностей коммуникации предприятий с клиентами и партнерами;
- реализовать новые методы навигации, применяемые в Интернет, во всех продуктах, с тем, чтобы облегчить операции поиска и анализа информации, а также взаимодействия с партнерами;
- упростить разработку, внедрение и администрирование прикладных программ с целью оптимизации и сокращения циклов разработки;

- интегрировать новые продукты и Интернет-технологии с существующими инфраструктурами, чтобы пользователи могли на основе уже имеющихся решений постепенно развивать свои информационные системы.

В основе решения всех этих задач лежат сетевые операционные системы. Напомним, что **операционная система** - это совокупность программ, предназначенная для обеспечения определенного уровня эффективности вычислительной системы за счет автоматизированного управления ее работой и предоставляемого пользователям определенного набора услуг.

Наибольший объем современных сетевых ОС представлен классами клиентских и серверных операционных систем, что определяется быстрым развитием сетей ЭВМ.

Клиентская ОС начала 2000-х гг. - это надежная, устойчивая и мощная операционная система для персональных компьютеров, которая обладает высоким уровнем защиты данных, работает с различными файловыми системами, на различных аппаратных платформах и поддерживает многопроцессорные системы. Как правило, эта ОС локализована, т.е. имеет полностью русифицированный интерфейс -все позиции меню, система справки, системные утилиты и приложения, входящие в состав, переведены на русский язык. Локализованная версия корректно поддерживает кириллицу на системном уровне и позволяет использовать русскоязычные приложения.

Сетевые возможности клиентской ОС позволяют использовать ее в качестве клиента сетей ЭВМ, а иногда, как невыделенный сервер, в одноранговых сетях, т.е. управлять правами доступа к отдельным каталогам и файлам. Пользователь может контролировать исполнение приложений, системных служб, просматривать динамический график использования памяти и процессора.

Важное значение придается безопасности и надежности. Приложения изолируются друг от друга и от аппаратуры. Тем самым исключено влияние некорректно работающего приложения на другие и работоспособность системы в целом. Многозадачный режим дает пользователям возможность работать быстро и выполнять одновременно несколько приложений без потери производительности.

Например, пользователь может работать с офисными приложениями в то время, когда в фоновом режиме идет загрузка данных из Интернет и печать документа.

Все клиентские ОС включают в свой состав средства для работы с Интернет:

- средство для навигации в Интернете;
- инструменты создания персональных Web-страниц и распространения информации внутри корпоративных сетей. Клиентские ОС позволяют выбирать при загрузке определенную конфигурацию, с которой будет работать конкретный пользователь. Профиль, который выбирается при загрузке, включает настройки разрешения экрана, информацию о запускаемых сервисах и устройствах. Администратор рабочей станции может создавать профили пользователей, в которых содержатся настройки рабочего стола и окружения для каждого пользователя, ограничения на доступ к системным ресурсам и приложениям. Профили могут быть индивидуальными или общими. В последнем случае администратор может создать специальный профиль и присвоить его всем пользователям рабочей станции.

В таких системах обязательно присутствуют развитые средства коммуникации в сетях ЭВМ - программы просмотра и создания факсов, системы электронной почты и службы удаленного доступа и др.

Клиентские ОС обязательно содержат средства, позволяющие им поддерживать совместную работу с подобными системами других производителей. Например, глобальная служба каталогов Novell, встроенная в Windows NT Workstation, реализует интеграцию с сетями на платформе Novell NetWare, обеспечивая выполнение сценариев входа в систему NetWare и функции доступа к файлам и каталогам.

Современные серверные ОС обеспечивают следующие, нижеперечисленные режимы работы в сетях ЭВМ:

- режим файл-сервера обеспечивает коллективное использование файлов;
- все файловые ресурсы, независимо от того, на каком диске они расположены, могут быть предоставлены для совместного доступа;

- в качестве рабочих станций могут выступать компьютеры, на которых установлены клиентские операционные системы.

Режим сервера печати позволяет подключать и предоставлять в совместное пользование необходимое число принтеров. Они могут быть подключены локально или по сети с помощью протоколов TCP/IP. Чтобы пользователь, который работает на автоматизированном рабочем месте (АРМе) с клиентской ОС, мог подключиться к удаленному принтеру, предоставляемому в режиме сервера печати, ему достаточно выбрать этот принтер из списка доступных. При этом используется драйвер, установленный на сервере.

В режиме сервера приложений обеспечивается выполнение с АРМа клиента приложений под управлением сервера, таких, как:

- системы управления базами данных;
- системы информационного обмена;
- системы управления производством, документооборотом, финансами и другими приложениями.

Сервер резервирования данных предоставляет возможность резервного копирования файлов. Администратор системы определяет пользователя, ответственного за эту операцию, и только он регулярно выполняет копирование данных на выделенный для этого носитель. Эта операция, как правило, автоматизирована.

В режиме удаленного доступа взаимодействуют две ОС -серверная, установленная на ЭВМ-сервере, и клиентская, установленная на АРМах. Пользователь АРМа, связанного с сетью через сервер удаленного доступа, чувствует себя работающим непосредственно в сети - он может осуществлять доступ к файлам и данным, печатать документы, обмениваться с коллегами сообщениями по электронной почте. Такой прозрачный доступ удобен администраторам системы и для связи территориально удаленных филиалов предприятий. Специальный набор протоколов (PPP) позволяет осуществлять удаленный доступ в условиях разнородной сети. Поддержка PPP гарантирует возможность удаленного доступа через любой стандартный PPP-сервер удаленного доступа. Современная сер-

верная ОС должна обеспечивать доступ к сети для пользователей, применяющих средства удаленного доступа других производителей.

Режим сервера связи обычно подразумевает возможность соединения между собой различных сегментов разнородных сетей. Для прозрачного подключения к разнородным сетям создаются продукты, обеспечивающие клиентскую и серверную части не зависящим от платформы протоколом доступа к файлам в сети.

Иногда в серверной ОС выделяется режим обеспечения услуг Интернет, таких, как сервер Web, FTP и Gofer, хотя часто эти услуги реализуются самостоятельными программами, обеспечивающими распространение в Интернет потоков аудио- и видеоинформации.

Все основные серверные ОС имеют:

- пользовательский графический интерфейс;
- диспетчер заданий;
- службу администрирования;
- службы мониторинга сети и осуществления диагностики;
- службы каталогов и регистрации сетевых пользователей. Администрирование сети - добавление новых пользователей,

авторизация доступа к ресурсам сети, отслеживания административных, кадровых, структурных, функциональных и технических изменений -производится централизованно с АРМа системного администратора.

Кроме клиентских и серверных ОС широко используются современные **операционные системы автоматизированного проектирования** (ОС САПР) - часть программного обеспечения автоматизированного проектирования, предназначенная для управления проектированием. В настоящее время подавляющее большинство проектных и конструкторских работ выполняется с помощью соответствующих САПР -от архитектурного и строительного проектирования до автоматизированных систем создания программных средств самих САПРов.

Операционные системы реального времени - системы, обеспечивающие режим работы ЭВМ в реальном времени, используются для управления техноло-

гическими процессами или автономными техническими системами, например космическими объектами.

В целом можно сказать, что ОС различного назначения занимают самое нижнее, базовое место в иерархии системных программных средств.

Интерфейсы и оболочки

По мере роста мощности ЭВМ увеличиваются затраты на диалоговую компоненту программного обеспечения. Термин «эффективность» постепенно изменил свое значение. Если раньше он отражал такие характеристики, как процессорное время и объем занимаемой памяти, то теперь под ним понимают простоту разработки, легкость сопровождения и удобство работы с программой. Поэтому затраты на исследование и разработку пользовательского интерфейса являются оправданными. **Интерфейс** - это совокупность средств и правил, обеспечивающих взаимодействие устройств цифровой вычислительной системы и (или) программ, а **интерфейс пользователя** - программные и аппаратные средства взаимодействия оператора или пользователя с программой или ЭВМ.

Большинство современных пользовательских интерфейсов основываются на аналогичных идеях:

- активное использование «мыши»;
- ориентированность на объекты;
- графика и имитация процессов и явлений;
- возможность использования алгоритмов, знакомых каждому человеку из его обыденной жизни.

Разнообразие аппаратных и системных платформ, на которых должно будет работать это программное обеспечение, требует его переносимости на уровне исходного кода. Вышеизложенные требования логически приводят к идее переносимого унифицированного программного инструмента для создания пользовательских интерфейсов, точнее, программной системы (модуля, блока), которая обеспечивает интерфейсом с пользователем.

Выделяют три объекта, для каждого из которых ставятся различные цели при разработке СПИ (система построения интерфейсов).

1. Интерфейс с пользователем:

- согласованность;

- поддержка пользователя разного уровня;

- обеспечение обработки ошибок и восстановления.

2. Разработчик программного обеспечения:

- предоставление абстрактного языка для конструирования интерфейса пользователя;

- предоставление согласованных интерфейсов для связанных прикладных задач;

- обеспечение простоты изменения интерфейса на стадии его проектирования (быстрое создание прототипа);

- упрощение разработки повторным использованием программных компонент;

- обеспечение простоты изучения и использования прикладных программ.

3. Конечный пользователь:

- согласованность интерфейса по прикладным программам;

- многоуровневая поддержка сопровождения или функций помощи;

- поддержка процесса обучения;

- поддержка расширяемости прикладных программ.

Эти цели определяют следующие функциональные характеристики СПИ:

- работа с входными устройствами;

- проверка допустимости ввода;

- обработка ошибок пользователя;

- реализация обратной связи;

- поддержка обновления/изменения данных прикладной задачи;

- поддержка задач развития интерфейса;

- синтаксическая поддержка.

Тема 3. Основы теории баз данных (6 часов)

Лекция №4

План:

1) Основные понятия теории баз данных.

2) Классификация баз данных.

3) Структурные элементы базы данных.

4) Модели данных.

1. Основные понятия теории баз данных

Современные информационные системы, основанные на концепции интеграции данных, характеризуются огромными объемами хранимых данных, сложной организацией, необходимостью удовлетворять разнообразные требования многочисленных пользователей.

Информационная система - система, реализующая автоматизированный сбор, обработку и манипулирование данными и включающая технические средства обработки данных, программное обеспечение и соответствующий персонал.

Цель любой информационной системы - обработка данных об объектах реального мира. Основой информационной системы является база данных. В широком смысле слова база данных - это совокупность сведений о конкретных объектах реального мира в какой-либо предметной области.

Под **предметной областью** принято понимать часть реального мира, подлежащего изучению для организации управления и в конечном счете автоматизации, например, предприятие, вуз и т.д.

Создавая базу данных, пользователь стремится упорядочить информацию по различным признакам и быстро производить выборку с произвольным сочетанием признаков. Большое значение при этом приобретает структурирование данных.

Структурирование данных - это введение соглашений о способах представления данных.

Неструктурированными называют данные, записанные, например, в текстовом файле.

Ниже приведен пример неструктурированных и структурированных данных, содержащих сведения о студентах (номер личного дела, фамилию, имя, отчество и год рождения).

Неструктурированные данные:

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

Легко убедиться, что сложно организовать поиск необходимых данных, хранящихся в неструктурированном виде.

Структурированные данные:

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

Чтобы автоматизировать поиск и систематизировать эти данные, необходимо выработать определенные соглашения о способах представления данных, т.е. дату рождения нужно записывать одинаково для каждого студента, она должна иметь одинаковую длину и определенное место среди остальной информации. Эти же замечания справедливы и для остальных данных (номер личного дела, фамилия, имя, отчество).

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

В современной технологии баз данных предполагается, что создание базы данных, ее поддержка и обеспечение доступа пользователей к ней осуществляются централизованно с помощью специального программного инструментария - системы управления базами данных.

База данных (БД) - это поименованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области.

Объектом называется элемент предметной области, информацию о котором мы сохраняем.

Объект может быть реальным (например, человек, изделие или населенный пункт) и абстрактным (например, событие, счет покупателя или изучаемый студентами курс). Так, в области продажи автомобилей примерами объектов могут служить МОДЕЛЬ АВТОМОБИЛЯ, КЛИЕНТ и СЧЕТ. На товарном складе - это ПОСТАВЩИК, ТОВАР, ОТПРАВЛЕНИЕ и т. д.

Система управления базами данных (СУБД) - это комплекс программных и языковых средств, предназначенных для создания, ведения и совместного применения баз данных многими пользователями.

Централизованный характер управления данными в базе данных предполагает необходимость существования некоторого лица (группы лиц), на которое возлагаются функции администрирования данными, хранимыми в базе.

2 Классификация баз данных

По *технологии обработки данных* базы данных подразделяются на централизованные и распределенные.

Централизованная база данных хранится в памяти одной вычислительной системы. Если эта вычислительная система является компонентом
сети ЭВМ, возможен распределенный доступ к такой базе. Такой способ использования баз данных часто применяют в локальных сетях ПК.

Распределенная база данных состоит из нескольких, возможно пересекающихся или даже дублирующих друг друга частей, хранимых в различных ЭВМ вычислительной сети. Работа с такой базой осуществляется с помощью системы управления распределенной базой данных (СУРБД).

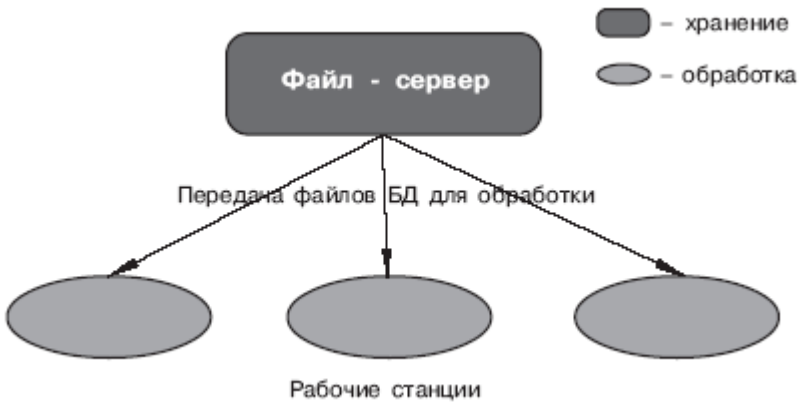
По *способу доступа к данным* базы данных разделяются на базы данных с локальным доступом и базы данных с удаленным (сетевым) доступом.

Системы централизованных баз данных с сетевым доступом предполагают различные **архитектуры** подобных систем:

- файл-сервер;

- клиент-сервер.

Файл-сервер. Архитектура систем БД с сетевым доступом предполагает выделение одной из машин сети в качестве центральной (сервер файлов). На такой машине хранится совместно используемая централизованная БД. Все другие машины сети выполняют функции рабочих станций, с помощью которых поддерживается доступ пользовательской системы к централизованной базе данных. Файлы базы данных в соответствии с пользовательскими запросами передаются на рабочие станции, где в основном и производится обработка. При большой интенсивности доступа к одним и тем же данным производительность информационной системы падает. Пользователи могут создавать также на рабочих станциях локальные БД, которые используются ими монопольно. Концепция файл-сервер условно отображена на рис. 1.1.



Рабочие станции Рис. 1.1. Схема обработки информации в БД по принципу файл-сервер

Клиент-сервер. В этой концепции подразумевается, что помимо хранения централизованной базы данных центральная машина (сервер базы данных) должна обеспечивать выполнение основного объема обработки данных. Запрос на данные, выдаваемый клиентом (рабочей станцией), порождает поиск и извлечение данных на сервере. Извлеченные данные (но не файлы) транспортируются по сети от сервера к клиенту. Спецификой архитектуры клиент-сервер является использование языка запросов SQL. Концепция клиент-сервер условно изображена на рис. 1.2.

Рисунок 2

3 Структурные элементы базы данных

Понятие базы данных тесно связано с такими понятиями структурных элементов, как поле, запись, файл (таблица) (рис. 1.3).

Рис. 1.3. Основные структурные элементы БД

Поле - элементарная единица логической организации данных, которая соответствует неделимой единице информации - реквизиту. Для описания поля используются следующие характеристики:

имя, например, Фамилия, Имя, Отчество, Дата рождения;

тип, например, символьный, числовой, календарный;

длина, например, 15 байт, причем будет определяться максимально возможным количеством символов;

точность для числовых данных, например два десятичных знака для отображения дробной части числа.

Запись - совокупность логически связанных полей.

Экземпляр записи - отдельная реализация записи, содержащая конкретные значения ее полей.

Файл (таблица) - совокупность экземпляров записей одной структуры.

Описание логической структуры записи файла содержит последовательность расположения полей записи и их основные характеристики, как это показано на рис. 1.4 и 1.5.

Имя файла					
Поле		Признак ключа	Формат поля		
Имя (обозначение)	Полное наименование		Тип	Длина	Точность (для чисел)
имя 1					
имя п					

Рис. 1.4. Описание логической структуры записи файла

В структуре записи файла указываются поля, значения которых являются ключами: первичными (ПК) и вторичными (ВК).

Первичный ключ (ПК) - это одно или несколько полей, однозначно идентифицирующих запись. Если первичный ключ состоит из одного поля, он называется простым, если из нескольких полей - составным ключом.

Вторичный ключ (ВК) - это одно или несколько полей, которые выполняют роль поисковых или группировочных признаков. В отличие от первичного, значение вторичного ключа может повторяться в нескольких записях файла, то есть он не является уникальным. Если по значению первичного ключа может быть найден один единственный экземпляр записи, то по вторичному -несколько.

Имя файла: СТУДЕНТ					
Поле		Признак ключа	Формат поля		
Обозначение	Наименование		Тип	Длина	Точность
Номер	№ личного дела	*	Симв	5	
Фамилия	Фамилия студента		Симв	15	
Имя	Имя студента		Симв	10	
Отчество	Отчество студента		Симв	15	
Дата	Дата рождения		Дата	8	

Рис. 1.5. Описание логической структуры записи файла СТУДЕНТ

4 Модели данных

Ядром любой базы данных является модель данных.

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

Модель данных - совокупность структур данных и операций их обработки.

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

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

Иерархическая модель данных

Иерархическая модель организует данные в виде древовидной структуры.

К основным понятиям иерархической структуры относятся: уровень, элемент (узел), связь.

Дерево представляет собой иерархию элементов, называемых узлами.

Узел - это совокупность атрибутов данных, описывающих некоторый объект.

На самом верхнем уровне иерархии имеется один и только один узел - корень. Каждый узел, кроме корня, связан с одним узлом на более высоком уровне, называемым исходным для данного узла. Ни один элемент не имеет более одного исходного. Каждый элемент может быть связан с одним или несколькими элементами на более низком уровне. Они называются порожденными (рис 1.6).

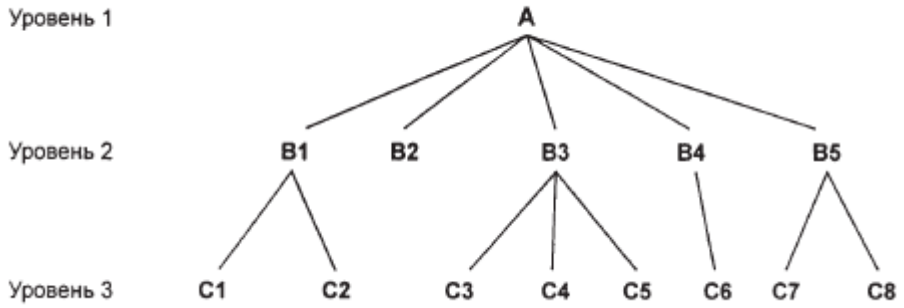


Рис. 1.6. Графическое изображение иерархической структуры БД

К каждой записи базы данных существует только один (иерархический) путь от корневой записи. Например, как видно из рис. 1.6, для записи С4 путь проходит через записи А и В3.

Пример, представленный на рис. 1.7, иллюстрирует использование иерархической модели базы данных. Для рассматриваемого примера иерархическая структура правомерна, так как каждый студент учится в определенной (только одной) группе, которая относится к определенному (только одному) институту.

4 Сетевая модель данных

Сетевая модель организует данные в виде сетевой структуры. Структура называется сетевой, если в отношениях между данными порожденный элемент имеет более одного исходного.

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

На рис. 1.8 изображена сетевая структура базы данных в виде графа.

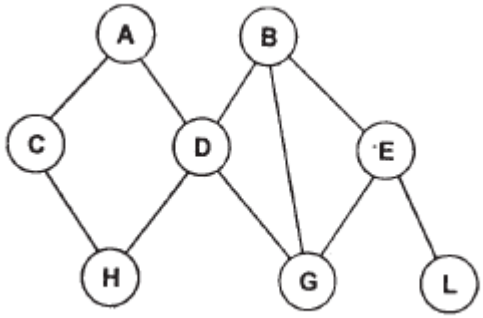


Рис. 1.8. Графическое изображение сетевой структуры

Рис. 1.8. Графическое изображение сетевой структуры

Примером сложной сетевой структуры может служить структура базы данных, содержащей сведения о студентах, участвующих в научно-исследовательских работах (НИРС). Возможно участие одного студента в нескольких НИРС, а также участие нескольких студентов в разработке одной НИРС. Графическое изображение описанной в примере сетевой структуры, состоящей только из двух типов записей, показано на рис. 1.9. Единствен отношение представляет собой сложную связь между записями в обоих направлениях.

Рис. 1.9. Пример сетевой структуры БД

Рис. 1.9. Пример сетевой структуры БД

1.4.3. Реляционная модель данных

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

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

Реляционная модель ориентирована на организацию данных в виде двумерных таблиц. Например, реляционной таблицей можно представить информацию о студентах, обучающихся в вузе (рис. 1.10).

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

Рис. 1.10. Пример реляционной таблицы

Реляционная модель данных является совокупностью взаимосвязанных двумерных таблиц - объектов модели.

Связи между двумя логически связанными таблицами в реляционной модели устанавливаются по равенству значений одинаковых атрибутов этих таблиц.

Каждая реляционная таблица представляет собой двумерный массив и обладает следующими *свойствами:*

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

При описании реляционной модели часто используют следующие термины: отношение, кортеж, домен.

Отношения представлены в виде таблиц, строки которых соответствуют записям (кортежам), а столбцы - полям, атрибутам отношений (доменам).

Поле, каждое значение которого однозначно определяет соответствующую запись, называется простым ключом (ключевым полем).

Если записи однозначно определяются значениями нескольких полей, то такая таблица базы данных имеет составной ключ.

В примере, показанном на рис. 1.10, ключевым полем таблицы является "№ личного дела".

Между двумя реляционными таблицами могут быть сформированы связи. Различные таблицы, могут быть связаны между собой через общее поле данных.

На рис. 1.11 показан пример реляционной модели, построенной на основе отношений: СТУДЕНТ, СЕССИЯ, СТИПЕНДИЯ.

СТИПЕНДИЯ (Результат)

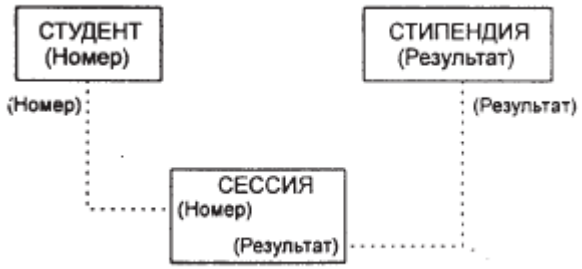


Рис 1.11. Пример реляционной модели

Рис 1.11. Пример реляционной модели

Таблица СТУДЕНТ имеет поля: Номер, Фамилия, Имя, Отчество, Дата рождения, Группа;

СЕССИЯ - Номер, Оценка 1, Оценка 2, Оценка 3, Оценка 4, Результат;

СТИПЕНДИЯ - Результат, Процент.

Таблицы СТУДЕНТ И СЕССИЯ имеют совпадающие ключи (Номер), что дает возможность легко организовать связь между ними. Таблица СЕССИЯ имеет первичный ключ Номер и содержит внешний ключ Результат, который обеспечивает ее связь с таблицей СТИПЕНДИЯ.

Благодаря имеющимся связям достигаются следующие преимущества:

1. Удается избежать дублирования информации. Все необходимые данные можно хранить только в одной таблице. Так, например, нет необходимости в таблице СЕССИЯ хранить номер группы каждого студента, сдающего экзамены, достаточно задать связь с таблицей СТУДЕНТ.

2. В реляционных базах данных легко производить изменения. Если в таблице СЕССИЯ изменить какие-нибудь значения, то правильная информация автоматически будет связана с другими таблицами, ссылающимися на первую (например, таблица СТИПЕНДИЯ).

3. В нереляционных базах данных сложно передать все имеющиеся зависимости, т.е. связать друг с другом данные из различных таблиц. Реляционная база данных выполняет все эти действия автоматически.

4. В реляционных базах данных удается легко избежать установления ошибочных связей между различными таблицами данных, а необходимый объем памяти сокращен до минимума.

Лекция №5-6

План:

1) Системы управления базами данных.

2) СУБД в многопользовательских системах.

3) Технология использования СУБД.

4) Обзор СУБД.

5) Проектирование баз данных. Понятие предметной области.

1) Системы управления базами данных

Система управления базами данных представляет собой пакет программ, посредством которого реализуется централизованное управление базой данных и обеспечивается доступ к данным. СУБД выступает в качестве интерфейса между пользователями и БД.

СУБД обеспечивает программные средства для создания, загрузки, запроса и обновления данных, контролирует действия, связанные с вводом-выводом данных, решает вопросы совместного их использования и защиты.

СУБД служит для поддержания базы данных в актуальном состоянии и обеспечивает эффективный доступ пользователей к содержащимся в ней данным в рамках предоставленных пользователям полномочий.

По степени универсальности различают два класса СУБД:

1) системы общего назначения;

2) специализированные системы.

СУБД общего назначения не ориентированы на какую-либо предметную область или на информационные потребности какой-либо группы пользователей. Каждая система такого рода реализуется как программный продукт, способный функционировать на некоторой модели ЭВМ в определенной операционной системе, и поставляется многим пользователям как коммерческое изделие. Такие СУБД обладают средствами настройки на работу с конкретной базой данных. Использование СУБД общего назначения в качестве инструментального средства для создания автоматизированных информационных систем, основанных на технологии баз данных, позволяет существенно сокращать сроки разработки, экономить трудовые ресурсы. Этим СУБД присущи развитые функциональные возможности и даже определенная функциональная избыточность.

Специализированные СУБД создаются в редких случаях при невозможности или нецелесообразности использования СУБД общего назначения.

СУБД общего назначения - это сложные программные комплексы, предназначенные для выполнения всей совокупности функций, связанных с созданием и эксплуатацией базы данных информационной системы.

- Они позволяют определять структуру создаваемой базы, инициализировать ее и производить начальную загрузку данных.

- Они управляют полномочиями пользователей на доступ к БД, организуют параллельный доступ к ней нескольких пользователей.

- Они обеспечивают защиту логической и физической целостности данных - защиту от разрушений.

- *СУБД поддерживают один из возможных типов моделей данных -сетевую, иерархическую или реляционную,* которые являются одним из важнейших признаков классификации СУБД.

- СУБД обеспечивают многоцелевой характер использования базы данных, защиту и восстановление данных. Наличие развитых диалоговых средств и языка

запросов высокого уровня делает СУБД удобным средством для конечного пользователя.

Основными средствами СУБД являются:

1) средства задания (описания) структуры базы данных;

2) средства конструирования экранных форм, предназначенных для ввода данных, просмотра и их обработки в диалоговом режиме;

3) средства создания запросов для выборки данных при заданных условиях, а также выполнения операций по их обработке;

4) средства создания отчетов из базы данных для вывода на печать результатов обработки в удобном для пользователя виде;

5) языковые средства - макросы, встроенный алгоритмический язык (Dbase, Visual Basic или другой), язык запросов (QBE - Query By Example, SQL) и т.п., которые используются для реализации нестандартных алгоритмов обработки данных, а также процедур обработки событий в задачах пользователя;

6) средства создания приложений пользователя (генераторы приложений, средства создания меню и панелей управления приложениями), позволяющие объединить различные операции работы с базой данных в единый технологический процесс.

2) СУБД в многопользовательских системах

База данных, как правило, содержит данные, необходимые многим пользователям. Получение одновременного доступа нескольких пользователей к общей базе данных возможно при установке СУБД в локальной сети персональных компьютеров и создании многопользовательской базы данных (рис. 1.12).

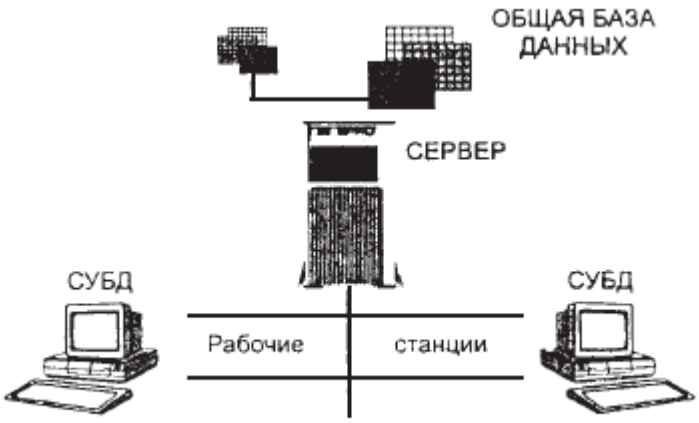


Рис. 1.12. СУБД в многопользовательской системе

В сети СУБД следит за разграничением доступа разных пользователей к общей базе данных и обеспечивает защиту данных при одновременной работе пользователей с общими данными. Автоматически обеспечивается защита данных от одновременной их корректировки несколькими пользователями-клиентами.

В сети с файловым сервером база данных может размещаться на сервере. При этом СУБД загружается и осуществляет обработку данных базы на рабочих станциях пользователей. Концепция файлового сервера в локальной сети обеспечивается рядом сетевых операционных систем. Наиболее популярными являются Microsoft Windows NT и NetWare Novell.

В сети, поддерживающей концепцию «клиент-сервер», используется сервер баз данных, который располагается на мощной машине, выполняет обработку данных, размещенных на сервере, и отвечает за их целостность и сохранность. Для управления базой данных на сервере используется язык структурированных запросов SQL (Structured Queries Language). На рабочих

станциях-клиентах работает СУБД-клиент. Пользователи могут взаимодействовать не только со своими локальными базами, но и с данными, расположенными на сервере. СУБД-клиент, в которой поддерживается SQL, в полном объеме может посылать на сервер запросы SQL, получать необходимые данные, а также посылать обновленные данные.

При этом с общей базой данных могут работать СУБД разного типа, установленные на рабочих станциях, если в них поддерживается SQL.

Подключение из СУБД к серверам баз данных SQL может быть осуществлено с помощью драйверов ODBC. ODBC (Open Database Connectivity -открытый стандарт доступа к базам данных) поддерживает стандартный протокол для серверов баз данных SQL.

Свойства СУБД и базы данных

К основным свойствам СУБД и базы данных можно отнести:

- отсутствие дублирования данных в различных объектах модели, обеспечивающее однократный ввод данных и простоту их корректировки;
- непротиворечивость данных;
- целостность БД;
- возможность многоаспектного доступа;
- всевозможные выборки данных и их использование различными задачами и приложениями пользователя;
- защиту и восстановление данных при аварийных ситуациях, аппаратных и программных сбоях, ошибках пользователя;
- защиту данных от несанкционированного доступа средствами разграничения доступа для различных пользователей;
- возможность модификации структуры базы данных без повторной загрузки данных;
- обеспечение независимости программ от данных, позволяющее сохранить программы при модификации структуры базы данных;
- реорганизацию размещения данных базы на машинном носителе для улучшения объемно-временных характеристик БД;
- наличие языка запросов высокого уровня, ориентированного на конечного пользователя, который обеспечивает вывод информации из базы данных по любому запросу и предоставление ее в виде соответствующих отчетных форм, удобных для пользователя.

3) Технология использования СУБД

СУБД является основой создания практических приложений пользователя для различных предметных областей.

Критерии выбора СУБД пользователем. Выбор СУБД для практических приложений пользователем определяется многими факторами, к которым относятся:

1) имеющееся техническое и базовое программное обеспечение, их конфигурация, оперативная и дисковая память;

2) потребности разрабатываемых приложений пользователя;

3) тип поддерживаемой модели данных, специфика предметной области, топология информационно-логической модели;

4) требования к производительности при обработке данных;

5) наличие в СУБД необходимых функциональных средств;

6) наличие русифицированной версии СУБД;

7) уровень квалификации пользователей и наличие в СУБД диалоговых средств разработки и взаимодействия с БД.

Этапы внедрения и эксплуатации БД:

Установка СУБД. СУБД является программным продуктом, поставляемым в виде пакета прикладных программ, который должен быть установлен (инсталлирован) на компьютер с учетом его конфигурации, ресурсов и операционной системы, а также требований к набору функций.

Процесс поэтапного внедрения. После установки СУБД можно осуществлять создание БД, в том числе задавать структуру БД, производить ввод данных, а также выполнять любые действия, предусмотренные функциональными возможностями СУБД. Следует заметить, что современные СУБД для ПК обладают достаточной гибкостью. Это позволяет на самых ранних этапах разработки приложений пользователя приступать к созданию отдельных частей БД. Такая БД по мере углубления разработки может легко расширяться и модифицироваться. Таким образом, облегчается ускоренное освоение персоналом технологии работы с БД, изучение возможностей СУБД и поэтапное внедрение.

Разработка структуры базы данных. Разработка приложений на основе СУБД предполагает подготовку решений по структуре БД. Эти решения непосредственно связаны с внемашинной сферой - с описанием внемашинной информационной базы, ее документов, содержащих необходимую информацию, а также с постановкой и алгоритмизацией задач по обработке этой информации.

На начальном этапе разработки структуры БД целесообразно построение информационно-логической модели, отражающей логическую структуру информации предметной области. Такая модель, отвечающая требованиям нормализации данных, является основой создания реляционных баз данных.

Создание базы данных средствами СУБД. В соответствии с разработанной структурой базы данных осуществляется ее создание средствами СУБД на машинном носителе и ввод в эксплуатацию. Для обеспечения процессов создания БД и ее эксплуатации необходимо знание возможностей инструментальных средств СУБД. При этом следует руководствоваться рекомендациями по технологии использования средств СУБД. Такая технология должна определять все необходимые процессы, включая первоначальный ввод, загрузку БД и контроль данных, выполнение операций по внесению изменений, реализацию запросов для получения нужных справок, восстановление БД и т.п. Одним из важнейших этапов этой технологии является подготовка экранных форм ввода-вывода для загрузки информации с документов внемашинной сферы в базу данных, корректировки данных и их просмотра.

Обработка данных средствами СУБД. Добавление, удаление, изменение и выборка данных производится при помощи языка запросов, встроенного алгоритмического языка и других средств СУБД. Реализация запросов обеспечивается диалоговой системой команд с меню или запросами по примеру QBE (Query By Example). В первом случае отдельный запрос выполняется одной или несколькими командами языка СУБД. Последовательность команд языка СУБД образует программу - командный файл (СУБД Dbase). Во втором - для выполнения запроса пользователь выбирает последовательно один или несколько пунктов меню или указывает в запросе пример (образец), по которо-

му составляется запрос, а также при необходимости указывает условия выбора и операции вычисления, которые необходимо выполнять с данными (СУБД Paradox, Access). Последовательность команд меню и запросов может быть запомнена в программе-макросе и в дальнейшем выполнена так же, как командный файл.

СУБД может иметь включающий или базовый язык программирования. В СУБД с включающим языком используется один из универсальных алгоритмических языков (C, Pascal и т. п.). Прикладная программа, написанная на включающем языке, может инициировать команды СУБД. В СУБД с базовым языком применяется собственный алгоритмический язык, позволяющий кроме операций манипулирования данными выполнять различные вычисления и обработку данных.

Стандартным реляционным языком запросов является язык структурированных запросов SQL (Structured Queries Language).

Язык запросов SQL реализован в целом ряде популярных СУБД для различных типов ЭВМ либо как базовый, либо как альтернативный. В силу своего широкого использования является международным стандартом языка запросов. Язык SQL предоставляет развитые возможности как конечным пользователям, так и специалистам в области обработки данных.

Совместимость с SQL-системами играет большую роль, когда предполагается проведение работы с корпоративными данными. Access и Paradox for Windows работают с источниками SQL-данных, совместимых с системой ODBC (Open Database Connectivity - открытое соединение баз данных). FoxPro (for DOS и for Windows) поставляются с дополнительными библиотеками, которые обеспечивают доступ к SQL-базам данных, способным работать совместно с системой ODBC.

Можно напрямую управлять базами данных Access с помощью языка SQL и передавать сквозные SQL-запросы совместимым со спецификацией ODBC SQL-базам данных, таким, как MS SQL Server и Oracle.

4) Обзор СУБД

Рынок программного обеспечения ПК располагает большим числом разнообразных по своим функциональным возможностям коммерческих систем управления базами данных общего назначения, а также средствами их окружения практически для всех массовых моделей машин и для различных операционных систем.

Используемые в настоящее время СУБД обладают средствами обеспечения целостности данных и безопасности, что дает возможность разработчикам гарантировать большую безопасность данных при меньших затратах сил на низкоуровневое программирование.

Продукты, функционирующие в среде WINDOWS, выгодно отличаются удобством пользовательского интерфейса и встроенными средствами повышения производительности.

Наибольшую популярность завоевали следующие СУБД:

dBASE, FoxPro, Paradox, Access, Oracle.

Производительность СУБД. Производительность СУБД оценивается:

- временем выполнения запросов;
- скоростью поиска информации;
- временем выполнения операций импортирования базы данных из других форматов;
- скоростью создания индексов и выполнения таких массовых операций, как обновление, вставка, удаление данных;
- максимальным числом параллельных обращений к данным в многопользовательском режиме;
- временем генерации отчета.

Достаточно быстрой СУБД является FoxPro 2.6, однако она не обладает средствами соблюдения целостности данных в отличие от более медленной СУБД Access 2.O.

Обеспечение целостности данных на уровне базы данных. Эта

характеристика подразумевает наличие средств, позволяющих удостовериться, что информация в базе данных всегда остается корректной и полной. Должны быть установлены правила целостности, и они должны храниться вместе с базой данных и соблюдаться на глобальном уровне.

Access и Paradox for Windows лучше других СУБД обеспечивают надежность сохранения целостности данных на уровне базы данных; правила хранятся вместе с базой данных и автоматически соблюдаются.

Обеспечение безопасности. Некоторые СУБД предусматривают средства обеспечения безопасности данных. Такие средства обеспечивают выполнение следующих операций:

Средства обеспечения безопасности информации

- шифрование прикладных программ;
- шифрование данных;
- защиту паролем;
- ограничение уровня доступа (к базе данных, к таблице, к словарю, для пользователя).

Самый высокий уровень безопасности данных реализован в СУБД dBASE IV. Администратор может назначить системе различные права доступа на уровне файла, поля, а также организовать автоматическое шифрование данных.

Хорошими характеристиками обеспечения безопасности отличается Access 2.О. Он предусматривает назначение паролей для индивидуальных пользователей или групп пользователей и присвоение различных прав доступа отдельно таблицам, запросам, отчетам, макрокомандам или новым объектам на уровне пользователя или группы.

Работа в многопользовательских средах. Практически все рассматриваемые СУБД предназначены для работы в многопользовательских средах, но обладают для этого различными возможностями.

Обработка данных в многопользовательских средах предполагает выполнение программным продуктом следующих функций:

1) блокировку базы данных, файла, записи, поля;

2) идентификацию станции, установившей блокировку;

3) обновление информации после модификации;

4) контроль за временем и повторение обращения;

5) обработку транзакций (транзакция - последовательность операций пользователя над базой данных, которая сохраняет ее логическую целостность);

6) работу с сетевыми системами (LAN Manager, NetWare, Unix). Хорошими возможностями для работы в многопользовательских средах
обладают СУБД Paradox for DOS 4.5, Access 2.0 и dBASE IV.

Импорт-экспорт. Эта характеристика отражает:

1) возможность обработки СУБД информации, подготовленной другими программными средствами;

2) возможность использования другими программами данных, сформированных средствами рассматриваемой СУБД.

Особый интерес представляют следующие форматы файлов: ASCII-файлы, .DBF, .WK*, .XLS.

Все рассматриваемые здесь СУБД обладают хорошими возможностями импорта-экспорта данных.

5) Проектирование баз данных. Понятие предметной области.

Каждая информационная система в зависимости от ее назначения имеет дело с частью реального мира, которую принято называть предметной областью (ПО) системы. ПО может относится к любому типу организаций: банк, университет, завод, магазин и т.д.

Предметная область информационной системы - это совокупность реальных объектов (сущностей), которые представляют интерес для пользователей.

Объект (сущность) - предмет, процесс или явление, о котором собирается информация, необходимая для решения задачи. Объектом может быть человек, предмет, событие.

Каждый объект характеризуется рядом основных свойств - атрибутов. **Атрибутом** называется поименованная характеристика объекта. Атрибут показывает, какая информация должна быть собрана об объекте.

Например, объект - клиент банка.

Атрибуты - номер счета, адрес, сумма вклада.

Технология анализа предметной области

Первым этапом проектирования БД любого типа является анализ предметной области, который заканчивается построением информационной структуры (концептуальной схемы). На данном этапе анализируются запросы пользователей, выбираются информационные объекты и их характеристики, которые предопределяют содержание проектируемой БД. На основе проведенного анализа структурируется предметная область. Анализ предметной области не зависит от программной и технической сред, в которых будет реализовываться БД.

Анализ предметной области целесообразно разбить на три фазы:

1) анализ концептуальных требований и информационных потребностей;

2) выявление информационных объектов и связей между ними;

3) построение концептуальной модели предметной области и проектирование концептуальной схемы БД.

Анализ концептуальных требований и информационных потребностей

Требования пользователей к разрабатываемой БД представляют собой список запросов с указанием их интенсивности и объемов данных. Эти сведения разработчики БД получают в диалоге с ее будущими пользователями. Здесь же выясняются требования к вводу, обновлению и корректировке информации. Требования пользователей уточняются и дополняются при анализе имеющихся и перспективных задач.

Рассмотрим примерный состав вопросника при анализе различных предметных областей.

Пример 1. Предлагается разработать БД для учета студентов вуза.

Анализ предметной области:

1. Сколько студентов учится в вузе?

2. Сколько факультетов и отделений в вузе?

3. Как распределены студенты по факультетам отделений и курсам?

4. Сколько дисциплин читается на каждом курсе по каждой специальности?

5. Как часто обновляется информация в БД?

6. Сколько преподавателей в вузе?

7. Сколько иногородних студентов живет в общежитии, на частных квартирах?

8. Сколько лекционных аудиторий и аудиторий для проведения практических занятий, лабораторий?

9. Какая преемственность существует между читаемыми курсами?

10. Как информация, представленная в п.п. 1-9, используется в настоящее время (расписание занятий, экзаменов, зачетов и т.д.) и как ее собираются использовать?

11. Сколько раз в день, сколько человек и кто пользуются БД?

Пример 2. Разработать требования к локальной БД "Аэропорт".

Вопрос 1. Для каких типов задач (приложений) проектируется БД?

Ответ. Для трех типов задач:

Задача 1. Информация об обслуживающем персонале.

Задача 2. Информация о полетных средствах.

Задача 3. Информация о графике движения самолетов.

Вопрос 2. Какими информационными объектами характеризуются эти задачи?

Ответ. Задача 1 характеризуется тремя информационными объектами: летный состав, диспетчеры, технический персонал.

Задача 2 характеризуется двумя информационными объектами: самолет, взлетное поле.

Задача 3 характеризуется одним информационным объектом - рейсы.

Вопрос 3. Каким текущим запросам должны удовлетворять данные информационные объекты? Ответ.

1. ФИО, звание, должность членов экипажа самолета.

2. Списочный состав диспетчеров.

3. Состав смены технического персонала.

4. Тип самолета, который может обслуживать тот или иной пилот.

5. Номер самолета, который обслуживает данный пилот, данная смена диспетчеров и технического персонала.

6. Номер личного дела сотрудника аэропорта.

7. Номер смены диспетчеров и технического персонала, обслуживающего аэропорт в заданном интервале времени.

8. Готовность самолета с таким-то номером к полету.

9. Количество часов налета такого-то самолета.

10. Готовность данной взлетной полосы в настоящее время.

11. Длина данной полосы.

12. Номер (номера) рейса до данного пункта назначения.

13. Какие промежуточные посадки совершает рейс №... ?

14. Время вылета и расчетное время прибытия рейса №... .

15. Время и место регистрации рейса №... .

16. Время посадки на рейс №... .

17. До какого времени задерживается рейс №... ?

18. Какие типы самолетов обслуживают рейс №... ?

19. Какой номер самолета обслуживает рейс №... ?

Вопрос 4. Каким перспективным запросам должны удовлетворять информационные объекты в БД "Аэропорт"?

1. С какого года используется самолет с №... в аэропорту, тип самолета?

2. Какое количество часов полета у члена экипажа, ФИО?

3. Расчетное время отпуска члена экипажа, диспетчера, технического работника.

Пример 3. Разработать БД "Видеомагнитофоны".

Вопрос 1. На кого рассчитана эта БД? Ответ. На покупателя видеосистем.

Вопрос 2. Что интересует покупателя?

Ответ. Покупателя интересуют технические характеристики системы, ее цена, фирма-изготовитель, технические характеристики и цена видеокассет, фирма-изготовитель кассет.

Далее проектировщик выбирает по технической документации параметры видеосистем, разрабатывает перечень запросов и уточняет его с будущим пользователем БД. Однако БД пока нет.

Есть только предложения проектировщика и одобрение будущего пользователя. Пользователю кажется, что все проблемы позади, а проектировщика еще ждет очень большая работа.

Выявленные запросы представлены следующим перечнем:

1. Выдать данные на определенную модель системы.
2. Какова цена той или иной системы?
3. Выдать системы определенной страны-изготовителя, цены которых не превышают заданную.
4. Выдать последние модели определенной фирмы.
5. Выдать модели определенной фирмы, габаритные размеры которых не превышают заданные.
6. Выдать № моделей определенной страны-изготовителя, цены которых не превышают заданные.
7. Какова цена последних моделей определенной фирмы?
8. Выдать № модели и фирму-изготовителя самого дешевого видеоплейера, выпущенного в определенном году.
9. Выдать характеристики видеомагнитофонов, выпущенных в определенном году.
10. Выдать наименование модели и фирму-изготовителя видеокассет с наибольшим временем проигрывания.
11. Выдать данные на самую дешевую видеокассету.
12. Выдать данные на самую дорогую видеокассету.
13. Сколько стоит видеокассета определенного наименования и определенной фирмы?
14. Выдать данные на видеокассету, имеющую определенное время проигрывания.

Тема 4. Проектирование баз данных (4 часа)

Лекция №7

План:

1) Технология анализа предметной области.

2) Логическое проектирование.

3) СУБД MS Access.

4) Связи между таблицами.

1) Технология анализа предметной области.

Выявление информационных объектов и связей между ними

Вторая фаза анализа предметной области состоит:

1 - в выборе информационных объектов,

2 - задании необходимых свойств для каждого объекта,

3 - выявлении связей между объектами,

4 - определении ограничений,

5 - накладываемых на информационные объекты,

6 - типы связей между ними,

7 - характеристики информационных объектов.

Проанализируем предметную область на примере БД "Видеомагнитофоны".

При выборе информационных объектов постараемся ответить на ряд вопросов:

На какие классы можно разбить данные, подлежащие хранению в БД?

Какое имя можно присвоить каждому классу данных?

Какие наиболее интересные характеристики (с точки зрения пользователя) каждого класса данных можно выделить?

Какие имена можно присвоить выбранным наборам характеристик?

Пример. Продолжим создание БД "Видеомагнитофоны", рассчитанной на пользователей, которые хотят приобрести данный вид техники.

После беседы с различными пользователями и просмотра каталогов было выяснено, что интерес представляют три информационных объекта: видеомагнито-

фон, видеоплейер, видеокассета. Рассмотрим наиболее существенные характеристики каждого информационного объекта.

Объект - ВИДЕОМАГНИТОФОН.

Атрибуты - страна-изготовитель, фирма-изготовитель, № модели, телевизионные системы, число кассетных гнезд, ресурс непрерывной работы, система автопоиска, напряжение в сети, наличие таймера, число программ, габаритные размеры, масса, цена в долларах, год выпуска.

Объект - ВИДЕОПЛЕЙЕР.

Атрибуты - страна-изготовитель, фирма-изготовитель, № модели, телевизионные системы, число воспроизводящих головок, ресурс непрерывной работы, напряжение в сети, наличие таймера, габаритные размеры, масса, цена в долларах, год выпуска.

Объект - ВИДЕОКАССЕТА.

Атрибуты - наименование, страна-изготовитель, фирма-изготовитель, тип кассеты, время проигрывания, цена в долларах.

Далее выделим связи между информационными объектами. В ходе этого процесса постараемся ответить на следующие вопросы:

Какие типы связей между информационными объектами?

Какое имя можно присвоить каждому типу связей?

Каковы возможные типы связей, которые могут быть использованы впоследствии?

Имеют ли смысл какие-нибудь комбинации типов связей?

Попытаемся задать ограничения на объекты и их характеристики.

Под ограничением целостности обычно понимают логические ограничения, накладываемые на данные. Ограничение целостности - это такое свойство, которое мы задаем для некоторого информационного объекта или его характеристики и которое должно сохраняться для каждого их состояния.

Введем следующие ограничения:

1. Значение атрибута "число кассетных гнезд" изменяется от 1 до 2.
2. Значение атрибута "ресурс непрерывной работы" изменяется от 4 до 24.

3. Значение атрибута "напряжение в сети" изменяется от 110 до 240 В.

4. Значение атрибута "число программ" изменяется от 1 до 20 и т.д.

Типы связей. Все информационные объекты предметной области связаны между собой.

Соответствия, отношения, возникающие между объектами предметной области называются **связями.** Различаются связи нескольких типов, для которых введены следующие обозначения:

а) один к одному (1:1);

б) один ко многим (1:М);

в) многие ко многим (М:М).

Рассмотрим эти типы связей на примере.

Пример. Дана совокупность информационных объектов, отражающих учебный процесс в вузе:

СТУДЕНТ (Номер, Фамилия, Имя, Отчество, Пол, Дата рождения. Группа) СЕССИЯ (Номер. Оценка 1, Оценка 2, Оценка 3, Оценка 4, Результат) СТИПЕНДИЯ (Результат. Процент) ПРЕПОДАВАТЕЛЬ (Код преподавателя. Фамилия, Имя, Отчество)

Связь один к одному (1:1) предполагает, что в каждый момент времени одному экземпляру информационного объекта А соответствует не более одного экземпляра информационного объекта В и наоборот.

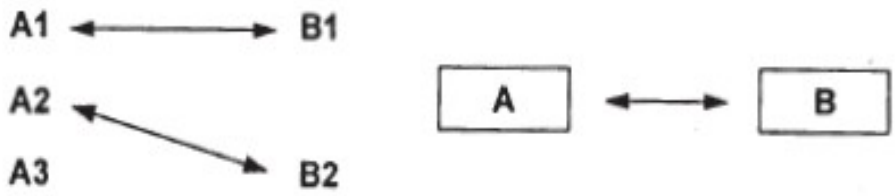


Рис. 2.1. Графическое изображение реального отношения 1:1

Примером связи 1:1 может служить связь между информационными объектами СТУДЕНТ и СЕССИЯ:

СТУДЕНТ <—> СЕССИЯ

Каждый студент имеет определенный набор экзаменационных оценок в сессию.

При связи **один ко многим** (1 :M) одному экземпляру информационного объекта A соответствует 0, 1 или более экземпляров объекта B, но каждый экземпляр объекта B связан не более чем с 1 экземпляром объекта A.

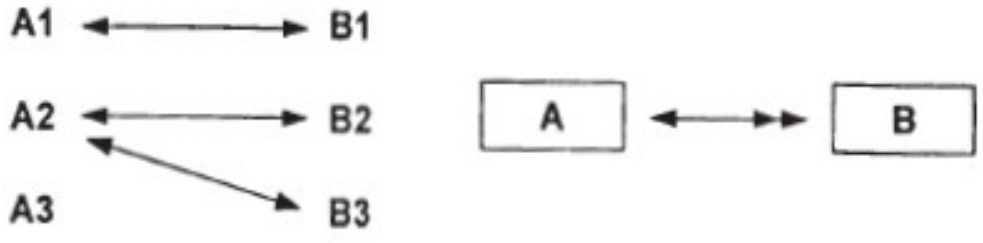


Рис. 2.2. Графическое изображение реального отношения 1:M

Примером связи 1:M служит связь между информационными объектами СТИПЕНДИЯ и СЕССИЯ:

СТИПЕНДИЯ <—» СЕССИЯ

Установленный размер стипендии по результатам сдачи сессии может повторяться многократно для различных студентов.

Связь **многие ко многим** (M:M) предполагает, что в каждый момент времени одному экземпляру информационного объекта A соответствует 0, 1 или более экземпляров объекта B и наоборот.

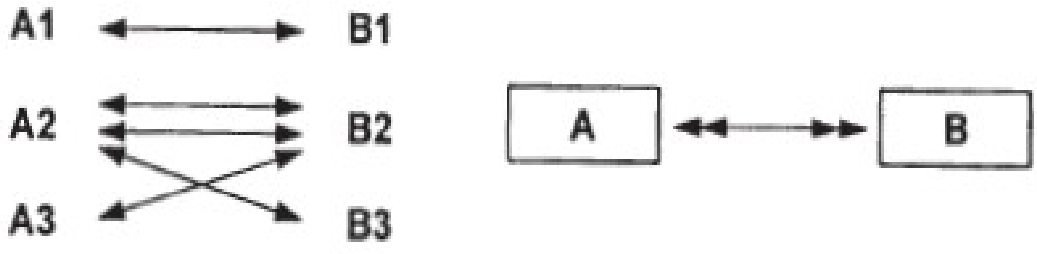


Рис. 2.3. Графическое изображение реального отношения M:M

Примером данного отношения служит связь между информационными объектами СТУДЕНТ и ПРЕПОДАВАТЕЛЬ:

СТУДЕНТ «— » ПРЕПОДАВАТЕЛЬ

Один студент обучается у многих преподавателей, один преподаватель обучает многих студентов.

Построение концептуальной модели предметной области

Заключительная фаза анализа предметной области состоит в проектировании ее информационной структуры или концептуальной модели.

Концептуальная модель включает описания объектов и их взаимосвязей, представляющих интерес в рассматриваемой предметной области (ПО) и выявляемых в результате анализа данных.

Концептуальная модель применяется:

1. для структурирования предметной области с учетом информационных интересов пользователей системы,

2. она дает возможность систематизировать информационное содержание предметной области, позволяет как бы "подняться вверх" над ПО и увидеть ее отдельные элементы. При этом, уровень детализации зависит от выбранной модели.

Концептуальная модель является представлением точки зрения пользователя на предметную область и не зависит ни от программного обеспечения СУБД, ни от технических решений.

Концептуальная модель должна быть стабильной. Могут меняться прикладные программы, обрабатывающие данные, может меняться организация их физического хранения, концептуальная модель остается неизменной или увеличивается с целью включения дополнительных данных.

Одной из распространенных моделей концептуальной схемы является модель «сущность - связь». Основными конструкциями данной модели являются сущности и связи.

Под сущностью понимают основное содержание объекта ПО, о котором собирают информацию. В качестве сущности могут выступать место, вещь, личность, явление.

Экземпляр **сущности** - конкретный объект.

Например:

сущность (объект) - служащий

экземпляр сущности - Иванов А. В.;

сущность (объект) - институт экземпляр сущности - СГУ.

Сущность принято определять атрибутами - поименованными характеристиками.

Например:

сущность - служащий

атрибуты: ФИО, год рождения, адрес, образование и т.д.

Чтобы задать атрибут в модели, ему надо присвоить имя и определить область допустимых значений. Одно из назначений атрибута - идентифицировать сущность.

Связь определяет отношения между сущностями. Типы связей: один к одному, один ко многим, многие ко многим.

При построении модели «сущность - связь» используют графические диаграммы. При этом обозначают:

сущности - прямоугольниками,

атрибуты - овалами,

связи - ромбами.

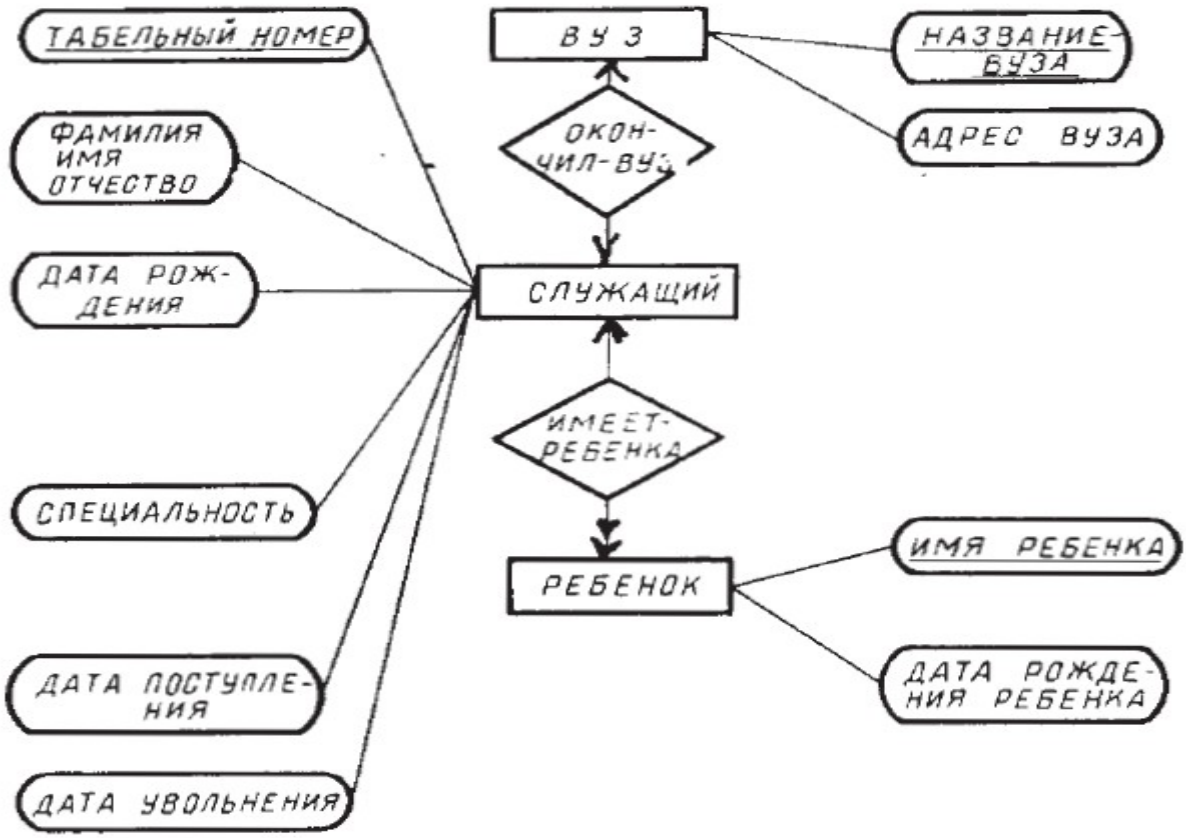


Рис. 2.4. Пример модели "сущность - связь"

На практике приходится строить несколько вариантов моделей, из которых выбирается одна, наиболее полно отображающая предметную область.

2) Логическое проектирование

Логическое проектирование представляет собой необходимый этап при создании БД.

Основной задачей логического проектирования является разработка логической схемы, ориентированной на выбранную систему управления базами данных (СУБД). Этап логического проектирования в отличие от концептуального проектирования полностью ориентирован на инструментальные средства компьютера.

Процесс логического проектирования состоит из следующих этапов:

1. Выбор конкретной СУБД.
2. Отображение концептуальной схемы на логическую схему.
3. Выбор ключей.
4. Описание языка запросов.

Одним из основных критериев выбора СУБД является оценка того, насколько эффективно внутренняя модель данных, поддерживаемая системой, способна описать концептуальную схему. Существующие СУБД делятся по типам моделей данных на реляционные, иерархические и сетевые. СУБД, ориентированные на персональные компьютеры, как правило, поддерживают реляционную модель данных. Подавляющее большинство современных СУБД - реляционные. Если выбрана реляционная система, то концептуальную схему БД предстоит отображать на реляционную.

При отображении концептуальной схемы на реляционную модель данных каждый прямоугольник схемы отображается в таблицу. При этом следует учитывать ограничения на размеры таблиц, которые накладывает выбранная СУБД.

Отобразим концептуальную схему, изображенную на рис. 2.5, на реляционную модель. Каждый прямоугольник (сущность) этой схемы будет представлен в

виде таблицы. Каждый столбец таблицы предназначен для записи одного атрибута и имеет свое уникальное имя.

Представим сущность ПРЕПОДАВАТЕЛЬ (ФИО, должность, звание, кафедра, стаж) в виде таблицы.

ПРЕПОДАВАТЕЛЬ

ФИО	Должность	Звание	Кафедра	Стаж

Определим структуру каждой таблицы, то есть зададим типы и размеры полей.

Признак ключа	Поле	Тип поля	Размер поля
Ключ	ФИО	символьный	21
	Должность	символьный	15
	Звание	символьный	10
Ключ	Кафедра	символьный	20
	Стаж	числовой	2

Можно произвести оценку требуемого объема памяти на хранение данной таблицы.

На хранение одной записи необходимо 21 + 15+10+20+2=68 байт. Всего предусмотрено 100 записей, то есть на хранение таблицы потребуется 6800 байт.

Аналогично поступают со всеми остальными сущностями (объектами) концептуальной схемы.

Выводы:

1) **к**онцептуальная модель представляет объекты предметной области и их взаимосвязи без указания способов их физического хранения,

2) концептуальная модель является, по существу, моделью предметной области,

3) При проектировании концептуальной модели все усилия разработчика должны быть направлены в основном на структуризацию данных и выявление взаимосвязей между ними без рассмотрения особенностей реализации и вопросов эффективности обработки.

4) Проектирование концептуальной модели основано на анализе решаемых на этом предприятии задач по обработке данных.

5) Концептуальная модель включает описания объектов и их взаимосвязей, представляющих интерес в рассматриваемой предметной области и выявляемых в результате анализа данных.

Концептуальная модель транспонируется затем в модель данных, совместимую с выбранной СУБД.

Версия концептуальной модели, которая может быть обеспечена конкретной СУБД, называется логической моделью.

Логическая модель отражает логические связи между элементами данных вне зависимости от их содержания и среды хранения.

Логическая модель данных может быть реляционной, иерархической или сетевой. Логическая модель отображается в физическую память, такую, как диск, лента или какой-либо другой носитель информации.

Физическая модель, определяющая размещение данных, методы доступа и технику индексирования, называется внутренней моделью системы.

В современных СУБД выполнение задач физического проектирования автоматизировано.

3) СУБД MICROSOFT ACCESS

Microsoft Access представляет собой реляционную базу данных.

Информация в базе данных Access представляется в виде отдельных таблиц. При этом каждый столбец таблицы соответствует полю данных, а каждая строка - записи данных.

При этом действует следующее правило (Записей данных состоящих из нескольких полей): запись данных представляет собой группу взаимосвязанных полей, рассматриваемых как единое целое.

Например, запись данных может содержать информацию о конкретном клиенте или продукте. Каждая запись данных в таблице содержит одинаковое число полей. Каждое поле содержит один и тот же тип информации. Например, в каждой записи данных некоторой таблицы поле ИМЯ КЛИЕНТА будет предназначено для имени клиента, т.е. будет содержать однотипную информацию.

Существуют различные типы данных. Тип данных для конкретного поля данных выбирается в зависимости от того, какая информация будет располагаться в этом поле. Размер поля данных определяется в зависимости от выбранного для него типа.

Вот некоторые примеры типов данных.

Текстовые поля (TEXT): могут содержать отдельные слова (например, имена), сочетания слов и чисел (например, почтовый адрес), или числа, которые не используются для математических расчетов (например, номера телефонов), а также специальные знаки (например «/» или «-»).

Числовые поля (NUMBER): содержат числа, используемые для различных расчетов (например, цены продуктов).

Поля даты и времени (DATE/TIME): содержат дату и/или время.

Поля денежных сумм (CURRENCY): содержат числовые величины в формате денежных сумм с обозначением денежных единиц - р. (рубль), $ (доллар), DM (немецкая марка) и т.п.

Логические поля (YES/NO): содержат логические данные со значениями ДА/НЕТ (YES/NO) или ИСТИНА/ЛОЖЬ (TRUE/FALSE).

Поля текстовых примечаний (MEMO): содержат большие участки текста (до 32 000 знаков).

4) Связи между таблицами

В базе данных Access между отдельными таблицами могут быть сформированы связи, соединяющие их в единую базу. Связи осуществляются через общие поля данных.

В качестве примера представим себе базу данных о клиентах, которая состоит из таблицы клиентов и таблицы заказов. Пусть в таблице клиентов

поля содержат разнообразную информацию о клиентах: фамилию, адрес, телефон и т.д., в том числе, и поле «Код клиента». В таблице заказов одним из прочих полей является «Код заказчика», имеющий такой же тип данных и размер, что и «Код клиента» в первой таблице. Если необходимо внести в таблицу за-

казов несколько конкретных заказов, поступивших от клиента Петрова, следует внести в записи об этих заказах только соответствующий «Код заказчика» и установить связь таблиц по этим полям. Access свяжет адресные данные клиента с заказом, и пользователь легко сможет выяснить, по какому адресу должен быть доставлен заказ, хотя описание заказа и адреса заказчиков хранятся в разных таблицах.

Лекция №8

План:

1) Ключи, макросы модули и объекты.

2) Запросы и выборки.

3) Формы данных.

4) Обработка данных с помощью отчетов.

1) Ключи, макросы модули и объекты.

Ключи

Часто необходимо осуществлять поиск данных, хранящихся в различных таблицах.

Предпосылкой и условием успешного поиска и установления связей служит правильное определение так называемых ключей. При помощи ключа Access может различать записи таблицы и упорядочивать их соответствующим образом. Каждая таблица должна содержать, как минимум, один ключ. Ключ представляет собой поле в соответствующей таблице. Он однозначно характеризует каждую запись, содержащуюся внутри таблицы.

Идеальным первичным ключом является такая информация, как номер клиента, номер счета и т.д. При помощи такого ключа легко определить соответствующую запись. Для первичного ключа не годятся поля, содержащие имя клиента. Всегда возможна ситуация, когда несколько человек имеют одно и то же имя.

Макросы, модули и объекты

При помощи макросов можно автоматизировать отдельные действия, не погружаясь в программирование. Запись макросов происходит по принципу магнитофона: отдельные действия, необходимые для определенной операции, записываются в нужной последовательности, а затем воспроизводятся любое количество раз простым нажатием кнопки. Например, для перехода к форме при открытии базы данных можно использовать специальный макрос. Тогда при открытии соответствующей базы данных на экране будет появляться форма.

Макросы могут не вполне удовлетворять потребностям пользователя. В этом случае можно воспользоваться языком программирования, имеющимся в Access. Он называется Access-Basic. При помощи этого языка автоматизируются сложные действия, например, сложные вычисления в отчете. Можно запрограммировать функцию вычислений и использовать ее, когда это необходимо. Процедуры, написанные на Access-Basic, запоминаются в программном модуле.

Все элементы, которые формируются и используются в Access называются объектами Access. К ним относятся таблицы, формы, запросы, отчеты, макросы, модули, графики, диаграммы, диалоговые окна, а также некоторые системные объекты, например, системный буфер обмена (Clipboard).

Конструкторы

Конструктор (Builder) - это инструмент Access, который облегчает выполнение конкретного задания. Крайне полезным является Конструктор выражений - он позволяет быстро сформировать сложное выражение. С помощью Конструктора запросов легко формируются запросы, которые используются для получения выборок данных для формы или отчета. Помимо перечисленных, в Access имеются и другие конструкторы. Это - макроконструктор, с помощью которого формируются различные макросы, а также конструкторы меню, полей, цветов, кодов и другие.

2) Запросы и выборки.

Запросы и выборки

Запросы используются для выбора из базы данных интересующей пользователя информации. Например, пользователя может заинтересовать, сколько клиентов его фирмы живет в Москве или каковы суммы выплаченной зарплаты сотрудникам его фирмы по подразделениям. Результат такого запроса называется выборкой. Под выборкой мы будем понимать динамическую таблицу с записями данных, которые удовлетворяют определенным условиям запроса.

Динамическая таблица запроса (Dynaset), в дальнейшем, выборка, -это таблица, формируемая всякий раз заново на основе реальных таблиц базы данных, содержимое которой удовлетворяет условиям запроса. С выборкой можно обращаться как с реальной таблицей - например, редактировать ее записи. Внесенные изменения будут отражены в записях реальных таблиц, стоящих за этой выборкой.

Основные типы запросов

Стандартные запросы. Чаще всего используются **стандартные запросы выбора,** используемые для того, чтобы отобрать и представить в виде удобной динамической таблицы интересующие пользователя данные из таблиц базы данных.

Запросы на выполнение действия. Наряду со стандартными запросами выбора, можно выделить также **запросы на выполнение действия,** которые используются для создания новых реальных таблиц данных, в дальнейшем существующих уже независимо от тех таблиц базы, которые были использованы для их построения. Эти запросы позволяют также изменять таблицы базы данных: обновлять их, дополнять новыми записями или удалять некоторые записи. С помощью запросов на выполнение действия можно перемещать или изменять табличные данные, обновлять, добавлять или удалять группы записей. Можно сформировать на основе выборки новую реальную таблицу данных, которая продолжит свое независимое существование и после закрытия запроса. Различают четыре вида запросов на выполнение действия.

Запрос на добавление. Можно добавлять отобранные записи из таблицы или запроса текущей базы данных в конец другой таблицы. Дополняемая таблица может находиться как в той же самой, так и в другой базе данных.

Запрос на удаление. С помощью запроса на удаление можно удалить группу записей данных, удовлетворяющих заданным условиям.

Запрос на обновление. Можно изменить группу записей данных, удовлетворяющих определенному условию.

Запрос на создание новой таблицы. Из выборки, являющейся динамической таблицей, сформированной при выполнении запроса и существующей только до окончания работы с результатами запроса, можно создать новую таблицу базы данных. Динамическая таблица является представлением данных других таблиц базы данных, и изменения, внесенные в выборку, будут отражены в записях таблиц, на основании которых эта выборка построена. Если же создать новую таблицу, она начнет свое независимое существование. Изменения в этой новой таблице не будут затрагивать использованные для ее построения исходные таблицы базы данных.

Параметрические запросы. Часто пользователю приходится иметь дело с запросами, устроенными в принципе одинаково, но имеющими некоторые различия в поставленных условиях отбора. В таком случае, чтобы каждый раз заново не создавать отдельный запрос, следует сформировать запрос с параметрами.

При выполнении **запроса с параметрами** Access запрашивает у пользователя значение параметров для определения условий выборки. Речь может идти, например, о выборе телефонов клиентов из определенного региона, причем интересующий пользователя регион изменяется от одного запроса к другому. В данном случае система запрашивает название региона.

Перекрестные запросы. Перекрестные запросы позволяют представить данные в весьма наглядной, компактной форме сводных кросс-таблиц, осуществляя при этом разнообразную группировку записей и групповые вычисления при обработке данных.

В качестве заголовков строк и столбцов используются значения заданных полей или выражения.

К примеру, нужно представить информацию о ежемесячных объемах продажи по всем наименованиям товаров. Если просто сгруппировать записи по месяцам и наименованиям товаров, то каждое наименование товара появится в выборке двенадцать раз - по разу для каждого месяца. Если же представить данные в виде сводной кросс-таблицы, в строках которой данные группируются по наименованиям товаров, а в столбцах - по месяцам, то результирующая выборка будет намного компактнее и нагляднее.

Два способа формирования запросов

По способу формирования запросы можно разделить на два вида:

Запросы по образцу или QBE-запросы, для определения которых пользователь должен указать параметры запроса в окне конструирования, задавая образцы для поиска информации.

Структурированные, или SQL-запросы, для определения которых пользователь должен описать запрос с помощью особого языка запросов, используя специальные команды и функции.

Практически любой запрос может быть сформулирован как в виде запроса по образцу, так и в виде SQL-запроса. Однако многие пользователи предпочитают работать с запросами по образцу, так как этот способ более наглядный, его легче освоить, не требуется учить команды и функции структурированного языка запросов.

При определении запроса по образцу Access автоматически формирует соответствующий SQL-запрос.

Основной тип запросов по образцу - это запросы на выбор. Пользователь может формировать запрос самостоятельно или обратиться к Мастерам запросов.

3) Формы данных.

Формы данных

В Access существует несколько способов ввода, редактирования и просмотра хранящейся в базе данных информации. Простейшим из них является следующий: открыть таблицу базы данных или запрос в табличном режиме - и можно вводить новые записи, удалять или редактировать старые, просматривать данные, относящиеся к этой таблице или запросу.

Однако есть и другой способ - подготовить формы данных, которые специально предназначены именно для наглядного представления информации из базы данных и могут существенно облегчить как ввод данных, так и восприятие хранящейся в базе данных информации.

Создание формы

Пользователь может создать простейшую форму на основе уже существующей таблицы Access. Может выбрать, какие поля и в какой последовательности должны быть представлены в форме, разбить их на логически связанные группы и задать их расположение на экране.

Access позволяет создать несколько форм для различного вида информации, например, формы счетов, заказов, адресов покупателей (клиентов) и другие. Данные одной таблицы могут быть представлены в нескольких различных формах, в то же время в одной форме может быть представлена информация из различных таблиц базы данных. Формы могут содержать иллюстрации, а также графическое представление информации из базы данных в виде диаграмм.

Как и при разработке таблиц и запросов, при разработке форм можно воспользоваться помощью соответствующего Мастера. При создании формы с помощью Мастера пользователю предстоит «дать интервью», ответив на его вопросы в серии диалоговых окон. Результатом такого диалога с Мастером форм будет служить готовая к употреблению форма, которую при желании можно будет подвергнуть дальнейшему усовершенствованию.

Существуют пять видов форм, которые можно разработать с помощью Мастера форм:

В один столбец (SINGLE COLUMN) Табличная форма (TABULAR) Диаграмма (GRAPH) Составная форма (MAIN/SUBFORM) Простая форма (AUTOFORM).

Форма в один столбец

Каждая запись данных представлена на отдельной странице формы. Для просмотра записей данных их необходимо пролистывать. Каждое поле данных находится в собственной строке. Поэтому имя и фамилия или индекс и место жительства представлены в отдельных строках, так как в данном случае речь идет об отдельных полях.

Подписи полей в форме соответствуют именам полей таблицы или их подписям, если они были заданы при определении свойств полей таблицы (в нижней части окна конструирования). Это, **Рис. 3.1. Форма в один столбец**

Табличная форма

Этот вид формы напоминает обыкновенное табличное представление запроса или таблицы базы данных. В табличной форме на одном листе одновременно представлено несколько записей данных. При этом каждая запись данных занимает отдельную строку. Каждое поле данных находится в отдельном столбце, а название поля располагается в шапке столбца. Внешний вид такой формы может быть очень похож на стандартное представление таблицы базы данных (рис. 3.2).

Форма-диаграмма

Форма-диаграмма - это форма со встроенной диаграммой, отображающей информацию из базы данных в графическом виде.

Пример. В базе данных AUTO, в таблице «Модели автомобилей», хранятся технические и экономические характеристики автомобилей - например, заводская цена. Эта информация может быть отображена в форме в виде диаграммы (рис.3.3).

Составная форма

Составная форма - это форма, содержащая информацию из двух различных таблиц базы данных, между которыми установлена связь. Одна из этих таблиц является главной, другая - подчиненной таблицей, с отношением связи между ними «один ко многим». При этом подчиненная таблица встроена в форму главной таблицы.

Пример. В верхней части формы представлены данные о моделях автомобилей (в виде формы в один столбец). В нижней части формы находится табличное представление всех заказов на данную модель. Главная таблица моделей связана с таблицей заказов отношением «один ко многим»

4) Обработка данных с помощью отчетов.

Обработка данных с помощью отчетов

В отчете пользователь может наглядно представить извлеченную из базы данных информацию, дополнив ее результатами анализа и вычислений. Пользователь свободен в выборе макета отчета и его оформления. В *MS* Access предусмотрены некоторые стандартные компоненты структуры отчета (шапки, примечания, колонтитулы, основная часть), однако каким именно будет макет отчета - решает пользователь.

Сконструировать макет отчета можно двумя способами: с помощью Мастера отчетов или самостоятельно. Начинающим пользователям Мастер отчетов помогает избежать ошибок. Во втором случае пользователь обладает более широкими возможностями. Однако возможен и промежуточный вариант: можно использовать помощь Мастера для создания «заготовки» отчета, а затем самостоятельно доработать этот отчет, внеся необходимые дополнения и изменения. Это наиболее рациональный способ.

Рассмотрим подробнее конструирование макета отчета с помощью Мастера.

Мастер отчетов помогает спроектировать отчет, ставя вопросы, касающиеся структуры, содержания и оформления отчета. Проведя с помощью серии диалоговых окон своеобразное «интервью», предлагая при этом на выбор возможные варианты ответов и демонстрируя возможные результаты выбора того или иного варианта ответа, Мастер формирует макет отчета.

Существует семь различных видов отчетов, которые можно создать с помощью мастера:

В один столбец (SINGLE COLUMN) Группировка данных и вычисление итогов Итоговый отчет (SUMMARY) Табличный отчет (TABULAR) Почтовые на-

клейки (MAILING LABEL) Слияние с MS Word (MS WORD MAIL MERGE) Простой отчет (AUTOREPORT).

Отчет в один столбец

В отчете этого вида каждое отображаемое поле записи расположено в отдельной строке. Слева в каждой строке находится подпись к полю. Записи данных расположены на странице строго друг под другом. Количество записей данных, помещающихся на одну страницу, определяется размером страницы.

У пользователя есть выбор между тремя различными стилями оформления отчета:

1. РАБОЧИЙ (EXECUTIVE) (предлагается использовать по умолчанию). В этом отчете используются двойные разделительные линии между записями. Подписи к полям выделяются полужирным шрифтом и отделены от полей несколькими пробелами.

2. ПРЕЗЕНТАЦИОННЫЙ (PRESENTATION). В нем записи отделены друг от друга ординарными линиями. Подписи к полям выделяются полужирным шрифтом и подчеркиванием. Расположение полей определяется графически, что обеспечивает строгое соблюдение выравнивания полей.

3. РАЗГРАФЛЕННЫЙ (LEDGER). В нем подписи к полям и сами поля располагаются в расчерченной таблице. Это облегчает сопоставление значений в отчете.

На рис 3.6 представлен отчет в один столбец (рабочий вариант).

Группировка данных и вычисление итогов

В этом отчете поля записи данных располагаются в строчку, в табличной форме. Подписи полей располагаются у верхнего края страницы. Access производит группировку записей и вычисляет промежуточные итоги по группам и общий итог для всех групп (рис. 3.7).

Итоговый отчет

В таком отчете производятся группировка данных и вычисление промежуточных и общих итогов. Выводимые поля располагаются в строчку -в табличной форме. Отличие от предыдущего отчета состоит в том, что итоговый отчет не

содержит детальных записей. В этом отчете выводятся только строки итогов, и результат выглядит более компактным (рис. 3.8).

Табличный отчет

В этом отчете поля записей данных располагаются, в строчку, в табличной форме. Подписи полей располагаются у верхнего края страницы. Каждая строка содержит новую запись данных. Вид этого отчета соответствует виду данных таблиц базы данных и запросов, представленных в табличной форме (рис. 3.9).

Слияние с MS Word

Слияние с MS Word - вид отчетов, используемый для подготовки серийных документов, что позволяет объединить данные таблицы или запроса MS Access с документом MS Word. Такое объединение документов и таблиц дает возможность быстро сформировать и отпечатать комплект схожих писем или других документов, подготовленных в MS Word и отличающихся друг от друга лишь некоторыми элементами, например, адресом или обращением к клиенту.

Эти элементы заполняются данными, взятыми из таблицы или запроса MS Access. Это означает следующее. Вместо конкретных данных об адресе клиента в документ MS Word вставляются специальные поля, соответствующие полям таблицы или запроса Access. При выводе документа на печать содержимое этих полей заполняется на основе конкретной записи данных из Access (рис. 3.11).

Наконец, последний вид отчетов, создаваемых с помощью Мастера, -это простой отчет. Если требуется быстро сформировать отчет в один столбец, в котором будут представлены все поля записи данных конкретной таблицы или запроса, следует выбрать этот вид отчета.

Создать простой отчет можно с помощью единственного щелчка мышью на кнопке инструмента ПРОСТОЙ ОТЧЕТ. Эта кнопка располагается на панели инструментов базы данных или открытой таблицы и запроса. При этом таблица или запрос не обязательно должны быть открыты. Достаточно выделить нуж-

ную таблицу или запрос на вкладке таблиц или запросов окна базы данных и нажать соответствующую кнопку.

Access автоматически сформирует простой отчет на основе открытой или выделенной таблицы или запроса, не задавая вопросов (рис. 3.12). Впоследствии можно, если нужно, отредактировать отчет. Для этого надо будет открыть отчет в режиме конструирования и внести изменения в его макет.

Тема 5. Стандартные приложения служебного назначения операционной системы Windows (2 часа)

Лекция №9

Стандартные приложения служебного назначения операционной системы Windows

План:

1) Обзор стандартных приложений служебного назначения. Очистка диска.

2) Дефрагментация диска.

3) Уплотнение диска.

4) Проверка диска.

5) Системный монитор.

1) Обзор стандартных приложений служебного назначения. Очистка диска.

Существует ряд служебных программ для работы с дисками. Они выполняют проверку на наличие ошибок, дефрагментацию, архивацию, а также очистку диска. Все эти программы можно запустить из главного меню Windows, однако удобнее это делать из папки Мой компьютер (My Computer).

Выберите команду **Мой компьютер** (My Computer) в главном меню Windows. Далее щелкните правой кнопкой мыши на значке, обозначающем диск, с которым будут работать служебные программы, чтобы открыть вспомогательное меню. Выберите команду **Свойства** (Options). Появится диалог изменения

свойств диска. Щелкните мышью на ярлычке **Сервис** (Tools), чтобы выбрать одноименную вкладку.

Это всевозможные временные файлы, скачанные из Интернет веб-страницы, а также не используемые файлы приложений. Чем больше программ вы устанавливаете и запускаете, чем чаще выходите в Интернет - тем больше шансов появления на жестком диске разного бесполезного мусора, который занимает драгоценное свободное место. Для поддержания порядка на своем ПК необходимо периодически устраивать уборку. Операционная система Windows XP предоставляет пользователю множество полезных утилит, среди которых есть программа под названием "Очистка диска".

Утилита "Очистка диска" располагается в стандартном наборе служебных программ Windows XP. Для ее запуска активируйте Пуск (Start), Программы (Programs), Стандартные (Accessories), Служебные (System Tools), Очистка диска (Disk Cleanup). Иначе программу можно запустить из командной строки, набрав ее название: cleanmgr. После запуска утилиты на экране появится рабочее окно программы.

Для продолжения работы необходимо выбрать логический диск, который будет подвергнут процедуре очистки.

Удаление программного обеспечения.

Никогда не удаляйте программное обеспечение путем помещения его папки в корзину. Следует использовать средство «Установка и удаление программ» (доступно с помощью команд «Настройка»\«Панель управления»), которое входит в состав Windows. Это средство должным образом удаляет все файлы программного обеспечения, разбросанные по системе. Некоторые программы имеют собственные средства удаления. В этом случае лучше использовать эти средства. Используйте специальные приложения для удаления шпионских программ (которые скрыты в невидимых каталогах).

2) Дефрагментация диска.

Дефрагментация диска - операция, состоящая в эффективном размещении физических структур файлов. При **дефрагментации** кластеры каждого файла размещаются в одном месте на **диске**, что сокращает время обращения.

Дефрагментация — процесс обновления и оптимизации логической структуры раздела диска с целью обеспечить хранение файлов в непрерывной последовательности кластеров. Применяется в основном в отношении файловых систем FAT и NTFS. После дефрагментации ускоряется чтение и запись файлов, а следовательно и работа программ. Другое определение дефрагментации: перераспределение файлов на диске, при котором они располагаются в непрерывных областях.

Длинные файлы занимают несколько кластеров. Если запись производится на незаполненный диск, то кластеры, принадлежащие одному файлу, записываются подряд. Если диск переполнен, на нем может не быть цельной области, достаточной для размещения файла. Тем не менее, файл все-таки запишется, если на диске много мелких областей, суммарный размер которых достаточен для записи. В этом случае файл записывается в виде нескольких фрагментов.

Процесс разбиения файла на небольшие фрагменты при записи на диск называется *фрагментацией*. Если на диске много фрагментированных файлов, скорость чтения носителя уменьшается, поскольку поиск кластеров, в которых хранятся файлы, требует времени.

Использование дефрагментатора

В поставке DOS дефрагментатор запускается командой *defrag*, также используются утилиты сторонних производителей. В Windows есть встроенная программа дефрагментации FAT (а начиная с Windows 2000 и NTFS), которую запускают последовательностью команд: Пуск -> Программы -> Стандартные -> Служебные -> Дефрагментация диска.

1. После запуска программы выбирают проверяемый диск.

2. Программа анализирует диск, дает оценку использования диска до дефрагментации.
3. Программа просматривает содержимое жесткого диска, выдает сообщение о степени его фрагментации и рекомендацию (обычно программа не настаивает на дефрагментации, пока процент фрагментированных файлов менее 20 %. Но для повышения эффективности работы диска стоит его дефрагментировать даже при показателе 6-8 %).
4. Запуск программы дефрагментации выполняют щелчком на кнопке Дефрагментация.
5. Ход процесса дефрагментации отображается на шкале выполнения задания.
6. Если дефрагментация затянулась, ее можно прервать щелчком на кнопке Пауза или Остановка. Впоследствии дефрагментацию можно продолжить.

Несколько слов о NTFS, дефрагментации дисков и таблице файлов

NTFS - объектно ориентированная файловая система, которая обрабатывает все файлы как объекты с атрибутами. Практически все объекты представляют собой файлы, а все, что имеется в файлах представляет атрибуты этого файла. Каждый занятый сектор на томе NTFS принадлежит какому-либо файлу. И вовсе не обязательно что файлы, необходимые для работы системы или какой-нибудь программы располагаются в соседних секторах/кластерах... Фрагменты данных могут находится в различных кластерах жесткого диска. В результате при удалении файлов освобождающееся дисковое пространство также становится фрагментированным. Чем выше степень фрагментации жесткого диска, тем ниже производительность файловой системы. Образно выражаясь, существуют 2 файла, необходимых для работы определенной программы. Один находится в начале диска, другой, через пустое пространство, в другом. ОС приходится обращаться к обеим этим файлам одновременно, что существенно тормозит систему, тем более портит головку жесткого диска.... Как из-

вестно, система сильнее всего фрагментирует файлы когда свободное место кончается, когда приходится использовать мелкие дырки, оставшиеся от других файлов. Тут возникает первое свойство NTFS, которое прямо способствует серьезной фрагментации.Диск NTFS поделен на две зоны. В начала диска идет MFT зона - зона, куда растет MFT, Master File Table. Зона занимает минимум 12% диска, и запись данных в эту зону невозможна. Это сделано для того, чтобы не фрагментировался хотя бы MFT. Но когда весь остальной диск заполняется - зона сокращается ровно в два раза. И так далее. Таким образом мы имеем не один заход окончания диска, а несколько. В результате если NTFS работает при диске, заполненном на около 90% - фрагментация растет как бешенная. Попутное следствие - диск, заполненный более чем на 88%, дефрагментировать почти невозможно - даже API дефрагментации не может перемещать данные в MFT зону. Может оказаться так, что у нас не будет свободного места для маневра. Далее. NTFS работает себе и работает, и всё таки фрагментируется - даже в том случае, если свободное место далеко от истощения. Этому способствует странный алгоритм нахождения свободного места для записи файлов - второе серьезное упущение. Алгоритм действий при любой записи такой: берется какой-то определенный объем диска и заполняется файлом до упора.

Для решения этой проблемы в состав ОС Windows XP входит включена оснастка Дефрагментация диска. Это очень хороший дефрагментатор, который не нуждается в том, чтобы его заменяли другим ПО этого назначения. Почему это мы так утверждаем: потому что сами тестировали знаменитые Norton Utilities, которые не имеют себе равных. И встроенный дефрагментатор со своей задачей справлялся ничуть не хуже. в его основные функции входит:

- Возможность работы с томами, имеющие кластеры любого размера.
- Возможность работы с любыми файловыми системами Windows.
- Обрабатываются файлы, занимающие ЛЮБОЕ количество кластеров.
- Возможность дефрагментирования Master File Table (MFT - главная таблица файлов)

- В процессе дефрагментации кластеры диска организуются таким образом, чтобы файлы, папки и свободное пространство по возможности располагались непрервывно. В результате значительно повышается производительность файловой системы, поскольку сокращается количество операций ввода/вывода, необходимое для чтения определенного объема информации. Следует также отметить, что при дефрагментации свободное пространство не объединяется в одну непрерывную область, а располагается в нескольких областях. Это значительно скоращает время выполнения дефрагментации.

Несколько слов о NTFS, дефрагментации дисков и таблице файлов.
Время проведения этой операции зависит от размера винчестера, что понятно. А также от степени фрагментации, например том, размеров в 10 ГБ, который ни разу не был дефрагментирован и который заполнен на 80% дефрагментировался у меня около часа! В то время как 40 гиговые винты могут быть дефрагментированы за 20 минут...все относительно.
Запустить процесс можно также из командной строки, вы будете лишены GUI, но процесс будет идти быстрее. Для этого в командной строке необходимо выполнить команду defrag.exe.

3) Уплотнение диска.

Уплотнение файлов применяют для уменьшения их размеров при подготовке к передаче по каналам электронных сетей или к транспортировке на внешнем носителе малой емкости, например на гибком диске.

Уплотнение папок используют как средство архивации данных перед длительным хранением, в частности, при резервном копировании.

Уплотнение дисков служит целям повышения эффективности использования их рабочего пространства и, как правило, применяется к дискам, имеющим недостаточную емкость.

В зависимости от того, в каком объекте размещены данные, подвергаемые сжатию, различают:

- уплотнение (архивацию) файлов;

- уплотнение (архивацию) папок;
- уплотнение дисков.

Приемы и методы работы со сжатыми данными.

Регулярно возникает необходимость сжимания данных перед размещением их в архивах или передачей по каналам связи с целью уменьшения их размера или занимаемого ими дискового пространства. Соответственно, возникает и обратная задача восстановления данных их предварительно уплотненных архивов.

1. Теоретические основы сжатия данных.

Характерной особенностью большинства данных является определенная избыточность, которая зависит от:

а) типа данных (так, у видеоданных степень избыточности обычно в несколько раз больше, чем у графических, а у графических – больше, чем у текстовых);

б) принятой системы кодирования (так, кодирование текстовой информации с помощью русского алфавита дает в среднем избыточность на 20-30% больше, чем при использовании английского языка).

Использование избыточности: улучшение восприятия информации, особенно в неблагоприятных условиях (просмотр телепередач при наличии помех, восстановление поврежденного графического материала и т.д.); повышение качества информации при ее обработке. Избыточность следует уменьшать для хранения готовых документов, что достигается эффектом сжатия данных. Термин *сжатие данных* заменяют термином *архивация данных*, а программные средства, выполняющие эти операции *архиваторами*

Существует достаточно много алгоритмов сжатия данных, однако все они теоретически основаны на трех способах уменьшения их избыточности:

- изменение содержания данных;
- изменение их структуры;
- изменение содержания данных и их структуры.

Изменение содержания данных при их сжатии делает метод сжатия необратимым и полное восстановление информации невозможно. Такие методы называются *методами сжатия с регулярной потерей информации*. Они применимы для данных, где формальная утрата части содержания не приводит к значительному снижению потребительских свойств. Сюда относятся мультимедийные данные: звукозаписи, рисунки и др. (форматы сжатия: .JPG для графических данных, .MP3 для звуковых данных, .MPG для видеоданных). Эти методы, обладая более высокой степенью сжатия, чем обратимые методы, неприменимы к текстовым документам, базам данных, программному коду.

Изменение структуры данных при их сжатии делает метод сжатия обратимым. Исходные данные восстанавливаются путем применения обратного метода. Обратимые методы применимы для сжатия любых типов данных (примеры форматов сжатия без потери информации: .GIF, .TIF, .PCX – для графических данных; .AVI – для видеоданных, .ZIP, .ARJ, .RAR, .LZH – для любых типов).

Теоремы (примем без доказательства):

1. Для любой последовательности данных существует теоретический предел сжатия, который не может быть превышен без потери части информации.
2. Для любого алгоритма сжатия можно указать такую последовательность данных, для которой он обеспечит лучшую степень сжатия, чем другие методы.
3. Для любого алгоритма сжатия можно указать такую последовательность данных, для которой данный алгоритм вообще не позволит получить сжатия.

Таким образом, различные методы сжатия демонстрируют наивысшую эффективность для данных разных типов и разных объемов.

В основе существующих обратимых методов сжатия данных лежат теоретические алгоритмы RLE, KWE и алгоритм Хаффмана. Их свойства приведены в таблице:

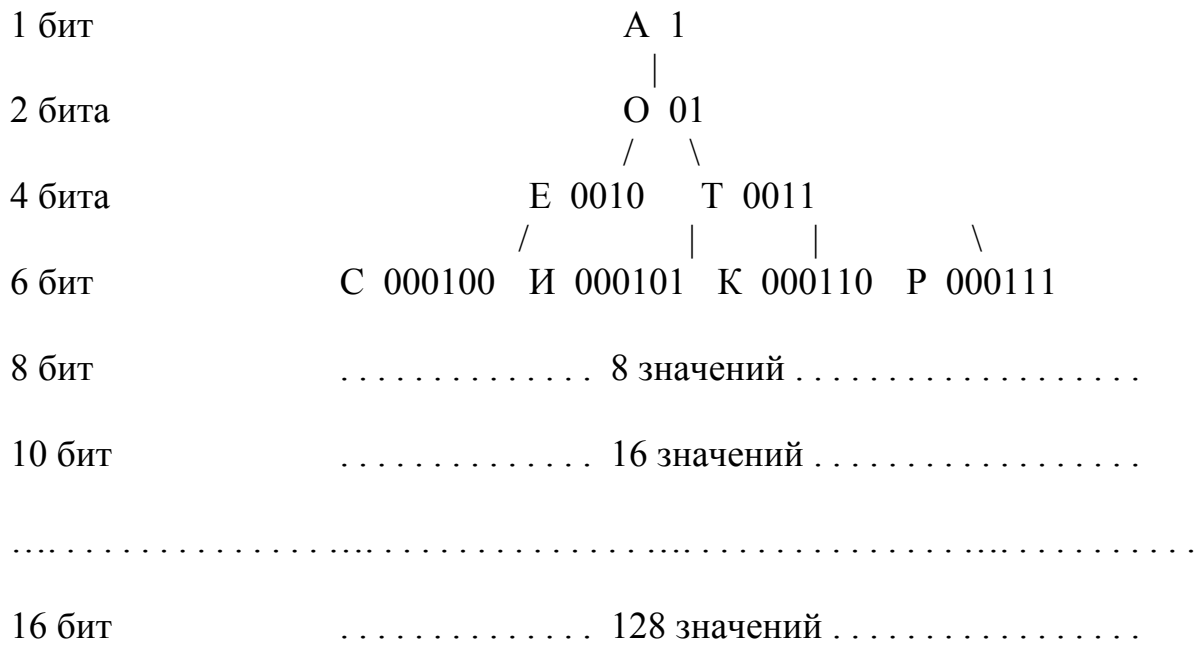
Алгоритм	Выходная структура	Сфера применения	Примечание
RLE (Run-Length Encoding)	Список (вектор данных)	Графические данные	Эффективность алгоритма не зависит от объема данных
KWE (Keyword Encoding)	Таблица данных (словарь)	Текстовые данные	Эффективен для массивов большого объема
Алгоритм Хафмана	Иерархическая структура (дерево кодировки)	Любые данные	Эффективен для массивов большого объема

В основе **алгоритмов RLE** лежит принцип выявления повторяющихся последовательностей данных и замены их простой структурой, в которой указывается код данных и коэффициент повтора. Например, для последовательности 0, 0, 0, 12, 12, 0, 23, 23, 23, 23 (10 байтов) образуется следующий вектор: 0, 3, 12, 2, 0, 1, 23, 4 (8 байтов) – здесь подчеркнуты коэффициенты повтора. В данном примере коэффициент сжатия равен 8/10 (80%). Алгоритмы RLE отличаются простотой, высокой скоростью работы, но недостаточным сжатием. Для текстовых данных эти методы неэффективны.

В основу **алгоритмов KWE** (по ключевым словам) положено кодирование лексических единиц (пример лексической единицы: слово – последовательность символов, ограниченная с обеих сторон пробелами или символами конца абзаца) исходного документа группами байтов фиксированной длины. Результат кодирования сводится в таблицу, которая прикладывается к результирующему коду и представляет собой словарь. Пары байтов, получаемых при кодировке слов, называются *токенами*. Эффективность метода зависит от длины документа: прикладываемый словарь значительно увеличивает длину кратких документов. Алгоритм эффективен для англоязычных текстовых документов и файлов баз данных.

В основе **алгоритма Хаффмана** лежит кодирование не байтами, а битовыми группами. Перед началом кодирования производится частотный анализ кода документа и выявляется частота повтора каждого встречаемого символа. Чем чаще встречается символ, тем меньшим числом битов он кодируется. Об-

разующаяся в результате кодирования иерархическая структура прикладывается к сжатому документу в качестве таблицы соответствия. Пример кодирования символов русского алфавита:



Как видно из схемы, с помощью 16 бит можно закодировать до 256 различных символов. При использовании последовательности длиной до 20 бит можно закодировать до 1024 лексических единиц (группы символов, слоги, слова). На файлах малых размеров алгоритм Хаффмана малоэффективен из-за прикладываемых к архиву таблиц соответствия. Наиболее эффективными оказываются архивы с размером словаря от 512 до 1024 единиц (длина кода до 18-20 бит).

Современные средства архивации данных используют более сложные (так называемые, «синтетические») алгоритмы, основанные на комбинации нескольких теоретических методов. Предварительно данные просматриваются, анализируются для индивидуальной настройки алгоритма с учетом особенности данных.

2. Программные средства сжатия данных.

Классические форматы сжатия данных:

Операционная система	Формат сжатия	Средство архивации	Средство разархивирования
MS-DOS	.ZIP	PKZIP.EXE	PKUNZIP.EXE
	.RAR	RAR.EXE	UNRAR.EXE
	.ARJ	ARJ.EXE	

Windows 9x	.ZIP	WinZip
	.RAR	WinRAR
	.ARJ	WinArj

Современные программные средства архивации данных называют *диспетчерами архивов*, ввиду своих больших функциональных возможностей, выходящих за рамки простого сжатия данных.

К базовым функциям, которые выполняют большинство современных диспетчеров архивов, относятся: извлечение файлов из архивов; создание новых архивов; добавление файлов в имеющийся архив; создание самораспаковывающихся архивов; создание распределенных архивов на носителях малой емкости; тестирование целостности структуры архивов; полное или частичное восстановление поврежденных архивов; защита архивов от просмотра и несанкционированной модификации.

Самораспаковывающиеся архивы. Такие архивы создают при необходимости передачи документа потребителю, если нет уверенности в наличии у пользователя требуемого программного средства. Самораспаковывающийся архив готовится с помощью обычного архива путем присоединения к нему небольшого программного модуля. Архив получает расширение .EXE, характерное для исполнимых файлов. Распаковка самораспаковывающегося архива происходит автоматически.

Распределенные архивы. При необходимости передачи большого архива на носителях малой емкости (например, на ГМД) возможно распределение одного архива в виде малых фрагментов на нескольких носителях. Диспетчер WinZip выполняет такое разбиение сразу на диски, а WinRAR и WinArj предварительно делает разбиение архива на фрагменты заданного размера на жестком диске, которые затем копируются на внешние носители. При использовании диспетчера WinZip каждый том архива дает файлы с одинаковыми именами, поэтому следует маркировать каждый диск пометками на наклейке, чтобы не перепутать последовательность немаркированных томов. Для того чтобы узнать номер тома по метке на диске используйте пункт Свойства контекстного меню

дисковода 3,5 (A:) в окне Мой компьютер. Номер тома можно узнать на вкладке Общие в поле Метка тома. В других архиваторах подобных проблем нет.

Защита архивов выполняется с помощью пароля, который запрашивается при попытке просмотра, распаковки или изменения архива. Для наибольшей защиты архива используйте в пароле не только символы английского алфавита и цифры, но и русский алфавит.

К дополнительным функциям диспетчеров архивов относятся сервисные функции, делающие работу более удобной (порой требуется подключение дополнительных служебных программ): просмотр файлов различных форматов без извлечения их из архива; поиск файлов и данных внутри архива; установка программ из архивов и проверка отсутствия компьютерных вирусов в архиве без предварительной распаковки; криптографическая защита архивной информации; декодирование сообщений электронной почты; создание самораспаковывающихся многотомных архивов; выбор и настройка коэффициента сжатия информации и др.

4) Проверка диска.

Одной из служебных программ является **Проверка диска** (Scandisk), позволяющая обнаруживать и исправлять различные ошибки на носителях информации. Эти ошибки могут возникнуть по разным причинам, например, из-за повреждений фрагментов магнитной поверхности жесткого диска. Еще одной причиной ошибок может служить внезапный сбой компьютера при записи информации на носитель. Часто заметить ошибки сразу бывает трудно и дальнейшая работа с таким носителем может только усугубить ситуацию.

Программа проверки дисков запускается автоматически при старте Windows, если вы неправильно завершили ее работу в предыдущий раз. В предыдущих версиях Windows при этом запускалась версия для ДОС, которая работала не очень надежно. Особенностью Windows XP является то, что всегда запускается Windows-версия программы проверки, которую мы и рассматриваем. Вы также можете в любое время провести проверку диска самостоятельно.

Нажмите кнопку **Выполнить проверку** (Check Now) в диалоге для запуска программы проверки. Будет запущена программа проверки дисков и появится ее рабочее окно (Рис. 9.2). Флажки в этом окне позволяют выбрать параметры проверки диска. Если установлен флажок **Автоматически исправлять ошибки** (Automatically Fix Errors), то программа будет сама принимать решение по исправлению найденных ошибок.

Если установить флажок **Проверять и восстанавливать поврежденные сектора** (Scan for and attempt recovery of bad sectors), то кроме проверки диска на наличие логических ошибок в файловой системе, также будет проведена проверка на наличие физических ошибок поверхности жесткого диска. Обычно этот флажок не устанавливают, так как такая проверка занимает много времени. Но иногда полную проверку все же следует проводить.

Установив нужные параметры, следует нажать кнопку **Запуск** (Start) для начала проверки. Проверка будет сопровождаться изменением прогресс-индикатора, расположенного в нижней части рабочего окна. По окончании проверки появится диалог с сообщением, что проверка завершена (Рис. 9.2). Если во время проверки были обнаружены ошибки, то вам будет предложено просмотреть отчет о проверке. Нажмите кнопку ОК, чтобы закрыть этот диалог.

Отметим, что при запуске программы проверки диска с помощью главного меню, из группы стандартных служебных программ вам будет предложено вначале выбрать диск для проверки. При запуске же из папки **Мой компьютер** (My Computer) такого предложения не будет, так как диск уже выбран, когда вы открываете диалог настройки свойств диска.

Тема 6. Инструментарий технологии программирования (2 часа)

Лекция №10

Инструментарий технологии программировани

План:

1) Состав и назначение инструментария технологии программирования. Средства для создания приложений.

2) Средства для создания приложений: локальные средства разработки программ; интегрированные среды разработки программ.

3) CASE- технологии создания информационных систем.

4) Программные продукты для создания приложений.

1) Состав и назначение инструментария технологии программирования.

Инструментарий технологии программирования обеспечивает процесс разработки программ и включает специализированные программные продукты, юторые являются инструментальными средствами разработчика. Программные продукты данного класса поддерживают все технологические этапы процесса проектирования, программирования (кодирования), отладки и тестирования создаваемых программ. Пол телями технологии программирования являются системные и прикладные программы.

Инструментарий технологии программирования – совокупность программ и программных комплексов, обеспечивающих технологию разработки, отладки и внедрения создаваемых программных продуктов.

В настоящее время бурно развивается направление, связанное с технологией создания программных продуктов. Это обусловлено переходом на промышленную технологию производства программ, стремлением к сокращению сроков, трудовых и материальных затрат на производство и эксплуатацию программ, обеспечению гарантированного уровня их качества. Это направление часто называют программотехникой.

Программотехника - технология разработки, отладки, верификации и внедрения прог-раммного обеспечения. Инструментарий технологии прогр аммирова -ния —программные продукты поддержки (обеспечения) технологии программирования.

В рамках этих направлений сформировались следующие группы программных продуктов:

- средства для создания приложений
- локальные средства, обеспечивающие выполнение отдельных работ по созданию программ.

Инструментарий технологии программирования делиться на

- Средства для создания приложений
- Средства для создания информационных систем
- САЗЕ-технология
- Локальные средства
- Интегрированные среды
- Языки и системы программирования
- Инструментальная среда пользователя
- Встроенные в систему реализации
- Независимые от системы реализации

Классификация инструментария технологии программирования

- Интегрированные среды разработчиков программ, обеспечивающие выполнение комплекса взаимосвязанных работ по созданию программ;

- САЗЕ-технология, представляющая методы анализа, проектирования и создания программных систем и предназначеннм автоматизации процессов разработки и реализации информационных систем.
- Средства для создания приложений
- Локальные средства разработки программ.

2) Средства для создания приложений: локальные средства разработки программ; интегрированные среды разработки программ.

2.1 Локальные средства разработки программ. Эти средства программных продуктов наиболее представительны на рынке и включают языки и системы программирования также инструментальную среду пользователя.

Язык программирования - формализованный язык для описаиия алгоритма решения задачи на компьютере.

Средства для создания приложений - совокупность языков и программирования, а также различные программные комплексы для отладки и поддержки создаваемых программ.

Трансляция может выполняться с использованием средств компиляторов или интерпретаторов.

Коммпиляхоры транслируют всю программу|, но без ее выполнения. Интерпретаторы, в отличие от компиляторов, выполняют операторную обработку и выполнение программы.

Существуют специальные программы, предназначенные для трассировки и анализа выполнения других программ, так называемые отладчики.

Отладчи позволяют:

- осуществить трассировку (отслеживание выполнения программы в пооперaном варианте,
- идентификацию места и вида ошибок в программе,
- наблюдение за внесением значений переменных, выражений и т.п.

Для отладки и тестирования праности работы программ создается база данных контрольного примера.

Системы программирования включают:

- компилятор;
- интегрированную среду разработчика программ;
- отладчик;
- средства оптимизации кода программ;

- набор библиотек (возможно с исходными текстами программ);
- редактор связей;
- сервисные средства (утилиты) для работы с библиотеками, текстовыми и двоичными файлами;
- справочные системы;
- документатор исходного кода программы;
- систему поддержки и управления проектом программного комплекса.

Средства поддержки проектов - новый класс программного обеспечения, предназначенный для отслеживания изменений, выполненных разработчиками программ; поддержки версий программы с автоматической разноской изменений; получения статистики о ходе работ проекта.

3) CASE- технологии создания информационных систем.

Средства CASE-технологии — относительно новое, сформировавшееся на рубеже 80-х гг. направление. Массовое применение затруднено крайне высокой стоимостью и предъявляемыми требованиями к оборудованию рабочего места разработчика.

CASE-технология — программный комплекс, автоматизирующий весь технологический процесс анализа, проектирования, разработки и сопровождения сложных программных систем.

Средства CASE-технологий делятся на две группы:

- встроенные в систему реализации — все решения по проектированию и реализации привязаны к выбранной системе управления базами данных (СУБД);
- независимые от системы реализации — все решения по проектированию ориентированы на унификацию начальных этапов жизненного цикла и

средств их документирования, обеспечивают большую гибкость в выборе средств реализации.

Основное достоинство CASE-технологии — поддержка коллективной работы над проектом за счет возможности работы в локальной сети разработчиков, экспорта/импорта любых фрагментов проекта, организационного управления проектом.

Некоторые CASE-технологии ориентированы только на системных проектировщиков и предоставляют специальные графические средства для изображения различного вида моделей:

- диаграмм потоков данных (DFD — data flow diagrams) совместно со словарями данных и спецификациями процессов;
- диаграмму "сущность-связь" (ERD — entity relationship diagrams) , являющуюся инфологической моделью предметной области *(см.* гл. 15);
- диаграмму переходов состояний (STD — state transition diagrams), учитывающую *события* и реакцию на них системы обработки данных.

Диаграмма DFD устанавливает связь источников информации с потребителями, выделяет логические функции (процессы) преобразования информации, определяет группы элементов данных и их хранилища (базы данных).

Описание структуры потоков данных, определение их компонентов хранятся в актуальном состоянии в словаре данных, который выступает как база данных проекта. Каждая логическая функция может детализироваться с помощью DFD нижнего уровня согласно методам нисходящего проектирования *(см.* гл. 18).

Выполняются автоматизированное проектирование спецификаций программ (задание основных характеристик для разработки программ) и ведение словаря данных.

Другой класс CASE-технологий поддерживает только разработку программ, включая:

- автоматическую генерацию кодов программ на основании их спецификаций;

■■ проверку корректности описания моделей данных и схем потоков данных;

■■ документирование программ согласно принятым стандартам и актуальному состоянию проекта;

■■ тестирование и отладку программ.

Кодогенерация программ выполняется двумя способами: создание каркаса программ и создание полного продукта.

Каркас программы служит для последующего ручного варианта редактирования исходных текстов, обеспечивая возможность вмешательства программиста; полный продукт не редактируется вручную.

В рамках CASE-технологий проект сопровождается целиком, а не только его программные коды. Проектные материалы, подготовленные в CASE-технологий, служат заданием программистам, а само программирование скорее сводится к кодированию переводу на определенный язык структур данных и методов их обработки, если не предусмотрена автоматическая кодогенерация.

Большинство CASE-технологий использует также метод "прототипов" для быстрого создания программ на ранних этапах разработки. Кодогенерация программ осуществляется автоматически — до 85-90% объектных кодов и текстов на языках высокого уровня, а в качестве языков наиболее часто используются Ада, Си, Кобол.

4) Программные продукты для создания приложений.

Программные продукты для создания приложений

Данный класс программных средств, как уже отмечалось выше, весьма представителен. Приведем характеристику некоторых продуктов, которые предлагает фирма Microsoft.

Макроассемблер MASM, обеспечивающий создание программ, быстро манипулирующих с данными большой размерности, поддерживающих различные форматы объектных файлов. Кроме того, можно создавать динами-

ческие библиотеки (DLL, VBX) для Microsoft Visual Basic, генерировать DOS-, Windows-приложения. Средства данного языка наиболее часто используются для разработки драйверов — специальных программ для эмуляции нестандартных устройств, подключаемых к компьютеру, различных преобразований форматов данных, поддержания интерфейсов доступа к данным в разнородных программных системах.

Компилятор Visual C++ for Windows Professional Edition 1.5 является системой программирования объектно-ориентированного типа, обеспечивающей просмотр иерархии классов объектов приложения (Source Browser), работу отладчика (Debugger), компилятора и др. В состав пакета входит библиотека классов MFC (Microsoft Foundation Classes Library), содержащая классы для реализации сложного пользовательского интерфейса, средства изготовления структуры пользовательского интерфейса (AppWizard), создания диалогов, меню, икон, растров, курсоров (App Studio), свойств новых интерфейсных классов, наследующих свойства классов MFC (Class Wizard). Компилятор полностью поддерживает стандарт OLE 2.0 системы Windows (см. ниже), ODBC (Open DataBase Connectivity) — для обеспечения доступа к данным в различных форматах, хранимых как в локальной базе данных, так и на сервере баз данных.

Microsoft IMSL Mathematical and Statistical Library(MATLAB) —математическая и статистическая библиотеки набора функций и примеров их использования (более 1000), которые можно вызвать из программ, написанных на языке C++.

Средства поддержки проектов Microsoft Delta for Windows, используемые для независимой от всего проекта новой версии программного модуля, отслеживания новых версий, автоматической разноски изменений по копиям проекта программной системы.

Технологические стандарты Microsoft, которые могут быть использованы разработчиками прикладных программ.

OLE (Object Linking and Embedding) 2.0 — позволяющий создавать приложения, включающие в свой состав объекты, полученные из других приложений.

Объект имеет две составляющие:

- внешнее представление объекта (presentation data);
- способ редактирования объекта (native data).

Любой объект может либо внедряться (embedding), либо связываться (linking) с приложением.

Технология OLE 2.0 обеспечивает:

- редактирование "чужого" объекта внутри приложения;
- экономию трудовых затрат на разработку программ за счет ссылок на существующие внешние объекты;
- информационную интеграцию приложений.

ODBC (Open DataBase Connectivity) — создание единого интерфейса доступа к различным базам данных на различных платформах.

Программа выступает в качестве *клиента,* а база данных — в качестве *сервера,* доступ реализуется с помощью драйвера. Разработчики новых СУБД обеспечивают создание соответствующих их форматам драйверов. Для создания прикладных программ, использующих стандарт ODBC, используется инструментарий ODBC Software Development Kit (SDK).

MAPI (Messaging Application Program Interface) — обеспечение независимости приложений от систем связи в режиме телекоммуникаций, который также работает по принципу драйвера.

MAPI поддерживает стандарт X.400 Association's Common Messaging Calls (CMC), а также ряд других интерфейсов (API, SDK, DDK).

Существуют также и интегрированные инструментальные среды для разработчиков программ других фирм:

- Delphi 2.0;
- Clarion for Windows 1.5 и др.

Специфика современной информационной технологии состоит в бурном развитии сетевых комплексов вычислительных машин, в создании программ для работы в архитектуре сети типа файл-сервер и клиент-сервер, ожидается, что начиная с середины 90-х годов 90% вновь создаваемых приложений будут являться приложениями типа клиент-сервер.

Рассмотрим систему Delphi 2.0, позволяющую создавать приложения типа клиент-сервер. Разработчику программ с использованием Delphi 2.0 предоставлены:

■■ объектно-ориентированный язык программирования;

■■ высокопроизводительный компилятор — скорость компилирования 120000 строк в минуту, это в настоящее время мировой рекорд скорости компиляции;

■■ объектно-ориентированная модель многократно используемых компонентов;

■■ средства наглядного (визуального) создания программ — набор визуальных средств для создания системы меню, экранных форм, отчетных форм и т.п., использование библиотеки визуальных компонентов и визуальных объектов;

■■ масштабируемая технология работы с базами данных — использование реляционно полного языка SQL, встроенная поддержка баз данных под управлением СУБД Oracle, Informix, Sybase, Interbase; применение локального сервера Interbase для отладки приложений;

■■ принцип "открытой системы", возможность добавления новых средств и перенос на другие платформы. Так, первоначально предполагается использовать среду Windows 3.1, затем — Windows 95 и Windows NT.

Все создаваемые программы средствами Delphi 2.0 разрабатываются как *экранные формы,* которые играют функцию окна и диалоговой панели одновременно. Форма содержит *элементы управления:* поля ввода, списки, текстовые метки, кнопки, которые поддерживают интерфейс пользователя с базой данных, обеспечивает запуск управляемых событиями процедур. Программа

создается на *визуальном уровне,* т.е. разработчик размещает в форме интефейсные элементы, каждый из которых рассматривается как *объект,* имеющий список свойств, реагирует на наступление указанных событий типа:

- нажатие левой кнопкой мыши один раз;
- двукратное нажатие левой кнопкой мыши;
- перемещение мыши (уход от объекта, фокусировка объекта и т.п.).

Свойства объектов можно заранее фиксировать либо изменять программным способом во время работы программы. Для каждого события создается программный код.

В состав системы входит *менеджер проектов,* который предназначен для управления проектами в среде Delphi, позволяя объединять ряд форм в единое приложение, добавлять и удалять файлы, перемещаться по файлам проекта, просматривая исходные тексты программ обработки событий и т.п. Для удобства редактирования объектов используется так называемый *браузер объектов* (Browser). Интегрированный отладчик позволяет выполнять пошаговую трассировку кода, задавать точки останова (Break points). Для создания пользовательских меню приложений служит *редактор меню,* позволяющий использовать готовые либо создавать новые *шаблоны меню. Графический редактор* дает возможность разработчику приложения создавать графические изображения, кнопки, иконки, использовать масштабирование и вставку внешних графических изображений.

Локальная версия среды разработки — Delphi Desktop Edition, предназначена для создания приложений, работающих с локальными базами данных (dBASE, Paradox). Разработчики могут создавать динамические библиотеки, которые будут доступны из программ, написанных на языках C++, Borland Pascal, Paradox for Windows, dBASE for Windows.

Для повышения производительности труда разработчиков обеспечивается многократное использование программных модулей. Например, объекты OLE можно импортировать и встраивать в любое место. По отзывам экспертов, Del-

phi наиболее перспективная среда разработчика Windows-ориентированных приложений, функционирующих в архитектуре I клиент-сервер.

Тема 7. Пакеты прикладных программ (4 часа)
Лекция №11-12

Пакеты прикладных программ

План:

1) Характеристика пакетов прикладных программ.

2) Методо-ориентированные и проблемно-ориентированные ППП.

3) Состав ППП.

4) ППП автоматизированного проектирования.

5) ППП общего назначения.

6) Офисные ППП.

7) Настольные издательские системы.

8) Программные средства мультимедиа.

1) Характеристика пакетов прикладных программ.

Пакет прикладных программ - комплект программ, предназначенных для решения задач из определенной проблемной области. Обычно применение пакета прикладных программ предполагает наличие специальной документации: лицензионного свидетельства, паспорта, инструкции пользователя и т.п.

Прикладное программное обеспечение - программное обеспечение, состоящее из:

- отдельных прикладных программ и пакетов прикладных программ, предназначенных для решения различных задач пользователей;
- автоматизированных систем, созданных на основе этих (пакетов) прикладных программ.

По современным взглядам, ППП — это совокупность совместимых программ для решения определенного класса задач.

Исходя из определения можно выделить следующие общие свойства ППП.

1) Совместимость программ, составляющих ППП, означает возможность их взаимного использования, общность структуры управляющих данных и используемых информационных массивов. Кроме того, ППП следует рассматривать как самостоятельное программное изделие, как особый вид прикладного ПО.

2) ППП всегда ориентируется на пользователей определенной квалификации как в программировании, так и в той области, к которой относятся задачи, решаемые с применением этого ППП.

3) Пакет состоит из нескольких программных единиц.

4) Пакет предназначен для решения определенного класса задач.

5) В пределах своего класса пакет обладает определённой универсальностью, т.е. позволяет решать все или почти все задачи этого класса.

6) В пакете предусмотрены средства управления, позволяющие выбирать конкретные возможности из числа предусмотренных в пакете. Пакет допускает настройку на конкретные условия применения.

7) Пакет разработан с учетом возможности его использования за пределами той организации, в которой он создан, и удовлетворяет общим требованиям к программному изделию.

8) Документация и способы применения пакета ориентированы на пользователя, имеющего определенный уровень квалификации в той области знаний, к которой относятся решаемые пакетом задачи.

9) Поскольку ППП предназначен для решения определенного класса задач, можно говорить о функциональном назначении пакета.

2) Методо-ориентированные и проблемно-ориентированные ППП.

Среди пакетов, предназначенных для решения прикладных задач пользователей, иногда выделяют *методо-ориентированные и проблемно-ориентированные пакеты.*

Методо-ориентированный пакет предназначен для решения задачи пользователя одним из нескольких методов, предусмотренных в пакете, причем метод либо назначается пользователем, либо выбирается автоматически на основе

анализа входных данных. Пример такого пакета — пакет математического программирования, позволяющий решить задачу выпуклого программирования либо методом штрафных функций, либо одним из вариантов методов возможных направлений.

Проблемно-ориентированные пакеты предназначены для решения групп (последовательностей) задач, использующих общие Данные. Это наиболее многочисленная группа пакетов. Проблемная ориентация может выражаться в общем характере операций» выполняемых пакетом. Типичные примеры таких пакетов — текстовые редакторы, табличные процессоры, пакет линейного программирования.

Проблемная ориентация может быть представлена и общей прикладной проблемой, решение которой распадается на отдельные задачи, для каждой из которых в пакете предусмотрен свой алгоритм. Типичные примеры — пакет для проведения расчетов межотраслевых балансов, пакеты, используемые в различных системах автоматизации проектирования.

В последние годы получили распространение так называемые интегрированные пакеты, представляющие собой пакеты широкого назначения, объединяющие текстовый редактор, процессор электронных таблиц, систему управления базой данных, пакет графического отображения данных (деловую графику) и средства обмена данными с удаленными абонентами. Наиболее популярны у пользователей интегрированные пакеты Мастер, Symphong, Framework.

3) Состав ППП.

При определении пакета программ было отмечено, что пакет состоит из нескольких программных единиц. Такие программные единицы обычно называют *программными модулями.* Пакет предназначен для решения задач определенного класса. Этот класс задач обычно называют *предметной областью пакета.* Применительно к ППП для решения расчетных задач предметная область определяет некоторую структуру данных, т.е. организацию входных, промежуточных и выходных данных. Говорят, что пакет использует информационную базу, соответствующую своей предметной области.

Для реализации выбранных пользователем конкретных действий пакет должен воспринимать от пользователя управляющую информацию. Эта управляющая информация представляется на формальном языке — входном языке пакета. Описание конкретного задания пользователя на входном языке пакета называют *программой на входном языке (ПВЯ).*

Решение каждой задачи в пакете сводится к выполнению соответствующего алгоритма.

Программные модули пакета, реализующие алгоритмы решения задач, предусмотренных в пакете, будем называть *обрабатывающими модулями.* Обрабатывающие модули выполняют преобразование данных, составляющих информационную базу пакета.

Для того чтобы преобразовать задание пользователя в последовательность вызовов обрабатывающих модулей, в пакет должны входить управляющие модули.

Чтобы обеспечить взаимодействие пакета с пользователем и управляющих модулей пакета с информационной базой и обрабатывающими модулями, в состав пакета включаются обслуживающие модули.

Таким образом, ППП можно рассматривать как объединение входного языка, информационной базы, управляющих, обслуживающих и обрабатывающих программных модулей. Совокупность обрабатывающих модулей часто называют функциональным наполнением пакета. Управляющие и обслуживающие модули называются системной частью пакета, или системным наполнением пакета.

Взаимодействие составных частей пакета схематически показано на рис. 5.2. Средствами операционной системы запускается головной управляющий модуль пакета (ведущий модуль). Затем организуются прием задания пользователя, представляемого в форме программ на входном языке (ПВЯ), и выполнение этого задания путем вызова в нужной последовательности обрабатывающих и обслуживающих модулей.

Под способом применения ППП будем понимать организацию взаимодействия пользователя с пакетом при решении задач. Выбор способа применения ППП зависит от многих факторов, из которых наиболее существенными являются возможности ОС и выбранного языка программирования, объемы обрабатываемых данных, продолжительность решения задачи, частота использования ППП, особенности квалификации пользователей пакета и требования к оперативности решения задач (допустимому времени ожидания результатов расчетов).

Способы применения существующих в настоящее время ППП весьма разнообразны, однако можно выделить некоторые типовые режимы, определяемые построением самого пакета и особенностями используемой ЭВМ и ОС.

Простейший режим с точки зрения построения ППП сводится к использованию отдельных программ пакета как подпрограмм некоторой главной программы, составляемой пользователем на каком-либо языке программирования, например ПЛ/1 или Си. В этом случае ППП состоит только из обрабатывающих модулей и может рассматриваться как расширение библиотеки подпрограмм используемого языка программирования.

Следующий по сложности реализации режим предполагает, что вся управляющая информация для конкретного выполнения пакета передается в виде законченной программы на входном языке при запуске пакета. Дальнейшая работа пакета проходит без участия пользователя. Такой режим по аналогии с соответствующим режимом ОС часто называют *пакетным.* Пакетный режим удобен, когда требуется решать много однотипных задач с использованием одной и той же программы на входном языке, когда время, затрачиваемое на решение каждой задачи, достаточно велико, когда программа на входном языке сложна и имеет значительный объем.

Большинство ППП, применяемых на персональных ЭВМ, ориентировано на *диалоговое взаимодействие* с пользователем в ходе решения задач.

Простейший диалоговый режим (вариант диалогового взаимодействия) состоит в том, что пользователь инициирует выполнение пакета, вводит задание

в форме программы на входном языке и на этом заканчивает управление выполнением пакета. Фактически этот режим отличается от пакетного только возможностью исправления ошибок в ПВЯ, повторного запуска пакета при неудачах.

Более сложный вариант диалогового режима, называемый также *режимом сопровождения,* предусматривает возможность динамического управления выполнением пакета. Управляющая информация вводится по частям и формируется пользователем в процессе работы с пакетом на основе анализа промежуточных результатов. Такая работа в большинстве случаев более естественна для пользователя, в частности, при использовании пакетов редактирования текстов, при работе с электронными таблицами, при решении многих расчетных задач.

4) ППП автоматизированного проектирования.

ППП автоматизированного проектирования. Программы этого класса предназначены для поддержания работы конструкторов и технологов, связанных с разработкой чертежей, схем, диаграмм, графическим моделированием и конструированием, созданием библиотеки стандартных элементов (темплетов) чертежей и их многократным использованием, созданием демонстрационных иллюстраций и мульт-фильмов.

Отличительными особенностями этого класса программных продуктов являются:

- высокие требования к технической части системы обработки данных,

-наличие библиотек встроенных функций, объектов, интерфейсов с графическими системами и базами данных.

5) ППП общего назначения.

ППП общего назначения

Данный класс содержит широкий перечень программных продуктов, поддерживающих преимущественно информационнйе технологии конечных пользователей. Кроме конечных пользователей этими программными продуктами

за счет встроенных средств технологии программирования могут пользоваться и программисты для создания усложненных программ обработки данных.

Представители данного класса программных продуктов:

Настольные системы управления базами данных (СУБД), обеспечивающие организацию и хранение локальных баз данных на автономно работающих компьютерах либо централизованное хранение баз данных на файл-сервере и сетевой доступ к ним.

Генераторы отчетов — самостоятельное направление разви-тия программных средств, обеспечивающих реализацию запросов и формирование отчетов в печатном или экранном виде в условиях сети с архитектурой клиент-сервер. Сервер отчетов подключается к серверу баз данных, используя все уровни передач и драйверы сервера баз данных. Серверы отчетов включают:

- программы планирования и учета времени для формирования отчетов по требованию
- пользователей, составление расписания выдачи и распространения отчетов по сети;
- программы управления очередью запросов на формирование отчетов;
- программы ведения словаря пользователей для разграничения доступа к сформированным отчетам;
- программы ведения архива отчетов и др.

Подготовленные отчеты рассылаются клиентам по электронной почте или с помощью другого транспортного агента. Серверы отчетов обычно поддерживают разнородные плат-формы, тем самым они эффективно работают в неоднородных вычислительных сетях.

Текстовые процессоры - автоматическое форматирование документов, вставка рисованных объектов и графики, составление оглавлений и указателей, проверка орфографии, шрифтовое оформление, подготовка шаблонов документов.

Табличный процессор - удобная среда для вычислений силами конечного пользователя; средства деловой графики, специализированная обработка (встроенные (|)ункции, работа с базами данных, статистическая обработка данных и др.).

Средства презентационной графики - специализированные программы, предназначенные для создания изображений и их показа на экране, подготовки слайд-фильмов, мультфильмов, видеофильмов, их редактирования, определения порядка следования изображений.

Интегрированные пакеты - набор нескольких программных продуктов, функционально дополняющих друг друга, поддерживающих единые информационные технологии, реализованные на общей вычислительной и операционной платформе.

Наиболее распространены интегрированные пакеты, компонентами которых являются:

- СУБД;
- текстовый редактор;
- табличный процессор;
- органайзер;
- средства поддержки электронной почты;
- программы создания презентаций;
- графический редактор.

Компоненты интегрированных пакетов могут работать изолированно друг от друга, но основные достоинства интегрированных пакетов проявляются при их разумном сочетании друг с другом. Пользователи интегрированных пакетов имеют унифицированный для различных компонентов интерфейс, тем самым обеспечивается относительная легкость процесса их освоения.

Отличительными особенностями данного класса программных средств являются: полнота информационных технологий для конечных пользователей; однотипный интерфейс конечного пользователя для всех программ, входящих в (динтегрированного пакета — общие команды в меню, стандартиые пиктограм

одних и тех же функций (сохранение на диске, печать, проверка орфограф шрифтовые оформления и т.п.), стандартное построение и работа с диалоговы окнами и др.;

общий сервис для программ интегрированного пакета (например, словарь и сред орфографии для проверки правописания, построитель диаграмм, конвертер дана идр.);

легкость обмена и ссылок на объекты, созданные программами интегрированно пакета (применяется два метода: ВВЕ — динамический обмен данными и ОЬЕ -динамическая компоновка объектами), единообразный перенос объектов (ш

- наличие единой языковой платформы для разработки макрокоманд, пользовательсю программ;
- возможность создания документов, интегрирующих в себе возможности различн

программ, входящих в состав интегрированного пакета.

Интегрированные пакеты эффективны и при групповой работе в сети многих полыо- 1 вателей. Так, из прикладной программы, в которой находится пользователь, можно отпра-вить документы и файлы данных другому пользователю, при этом поддерживаются стан-дарты передачи данных в виде объектов по сети или через электронную почту.

6) Офисные ППП.

Офисные ППП. Данный класс программных продуктов охватывает программы, обеспечивающие организационное управление деятельностью офиса:

Органайзеры (планирдвщики) — программное обеспечение для пл рования рабочего времени, составления протоколов встреч, расписаний, ведения запи и телефонной книжки.

В состав программ органайзеров входят: калькулятор, записная книжка, часы, кален-дарь и т.п. Наиболее часто подобное программное обеспечение разрабатывается для ноут,-буков, персональных компьютеров блокнотного трпа.

Программы-переводчики, средства проверки орфографии и правописания текста включают:

- программы-переводчики, предназначенные для создания подстрочника исходного
- текста на указанном языке;
- словари орфографии, используемые при проверке текстов; ..'.'..
- словари синонимов, используемые для стилевой правки текстов;

программы для распознавания считанной сканерами информации и преобразования в

текстовое представление.

Кним относятся:

7) Настольные издательские системы.

Настольные издательские системы. Данный класс программ включает программы, обеспечивающие информационную технологию компьютерной издательской деятельности:

- форматирование и редактирование текстов;
- автоматическую разбивку текста на страницы;
- создание заголовков;
- компьютерную верстку печатной страницы;
- монтирование графики;
- подготовку иллюстраций и т.п.

ППП AdoЬe Pagemaker обеспечивает подготовку многостраничных цветных публикаций, гибкий дизайн страниц, высококачественную печать. Формат печатной страни-цы А2, допустим максимальный размер публикации более 1060 мм.

Расширены возможности по верстке: неограниченное число страниц-шаблонов, кото-рые могут использоваться в одной публикации; применение различных эффектов к цвет-ным изображениям; настройка резкости и регулировка

цветов в импортированных файлах; возможно закрепление расположения объектов на странице, автоматическое выравнивание объектов.

Разработаны и включены новые цветовые библиотеки, используются новые технологии? которые расширяют цветовую гамму тради-ционной офсетной печати.

8) Программные средства мультимедиа.

Программные средства мультимедиа. Этот класс программных продуктов является относительно новым, он сформировался в связи с изменением среды обработки данных, появлением лазерных дисков высокой плот-ности записи с хорошими техническими параметрами по доступным ценам, расширением состава периферийного оборудования, подключаемого к персональному компьютеру, развитием сетевой технологии обработки, появлением региональных и глобальных информационных сетей, располагающих мощными информационными ресурсами. Основное наз-начение программных продуктов мультимедиа - создание и использование аудио- и видеоинформации для расширения информационного.пространства пользователя.

Программные продукты мультимедиа заняли лидирующее положение на рынке в сфере библиотечного информационного обслуживания, процессе обучения, организации досуга. Базы данных компьютерных изображений произведений искусства, библиотеки звуковых записей и будут составлять основу для прикладных обучающих систем, компью-терных игр, библиотечных каталогов и фондов.

Тема 8. Текстовый процессор – прикладной программный продукт (4 часа).

Вопросы: Работа с текстом: минимальный набор типовых операций – операции, производимые над абзацами документа, операции, производимые с фрагментами текста. Расширенный набор типовых операций: контекстный поиск и замена, операции сохранения, проверка правописания, использование шаблонов, автотекст, слияние документов. Работа издательских систем: особенности издательских систем, основы создания документа в издательских системах.

Лекция №13-14

Работа с текстом

Минимальный набор типовых операций

Минимальный набор типовых операций включает операции, производимые над документом в целом, над абзацами документа и над его фрагментами.

Операции, производимые над документом в целом. К операциям, производимым над документом, относятся:

- создание нового документа - присвоение документу уникального имени и набор всего текста документа на клавиатуре;
- загрузка предварительно созданного документа в оперативную память;
- сохранение документа - копирование документа из оперативной памяти во внешнюю;
- удаление документа - удаление созданного или загруженного документа с экрана;
- распечатка документа - создание твердой (бумажной) копии документа.

Операции, производимые над абзацами документа. Абзац является ключевым элементом в структуре документа для многих текстовых процессоров.

К операциям, производимым над абзацами относят установку границ абзацев и абзацных отступов, выравнивание, а также включение переноса слов.

Установку границ абзацев производят с помощью маркеров отступов, находящихся на координатной линейке, или соответствующими командами меню.

Выравнивание (выключка). Различают четыре вида горизонтального (влево, вправо, по центру, по ширине) и три вида вертикального выравнивания (вверх, вниз, по высоте).

Перенос. При выключенном режиме автоматического переноса слово, не поместившееся на строке, полностью переносится на следующую строку. Это не придает элегантности тексту; его правый край остается неровным. Для улучшения внешнего вида текста используют режим переноса. При ручном варианте переноса пользователь сам определяет место переноса, вводя дефис, и жестким переводом каретки (нажатием на клавишу Еп1:ег) переходит на следующую

строку. Использование такого режима переноса приводит к необходимости удаления дефисов при повторном форматировании текста документа.

При включенном режиме автоматического переноса реализуется мягкий вариант переноса: текстовый процессор сам делит слово на слоги и переносит его наилучшим способом. Этот режим не создает никаких трудностей при повторном форматировании.

Операции, производимые с фрагментами текста. Эти операции включают выделение фрагмента текста, его перемещение, копирование или удаление, которые были рассмотрены в предшествующем разделе главы. Кроме того, выделенный фрагмент текста можно напечатать, произвести поиск и замену символов, применить шрифтовое выделение и ряд других операций.

Расширенный набор типовых операций

Контекстный поиск и замена. Один из видов поиска - по месту -заключается в предварительной разметке текста, это удобно при работе, требующей многократного обращения к определенным местам документа. Для такого поиска используются, по аналогии с закладками в обычной книге, команды типа "создать закладку", "найти нужную закладку", "убрать закладку". Такие "закладки" чаще всего обозначаются цифрами или буквами алфавита.

Другой вид поиска - по образцу - заключается в том, что задается некоторый образец (символ, слово, группа слов или цепочка символов) и подается команда поиска. Текстовый редактор начинает просмотр документа. Просмотр приостанавливается каждый раз при обнаружении заданного образца, на экран выдается соответствующий фрагмент текста и пользователь может совершить нужные операции. Затем можно подать команду на продолжение или прекращение поиска.

Поиск по образцу удобен, например, при замене термина в документе. Так, если нужно заменить в тексте слово "личный" на слово "персональный", в качестве образца для поиска можно задать "личн", чтобы поиск не зависел от падежных окончаний. Текстовый редактор будет последовательно подводить курсор к каждому найденному образцу, а замену нужно будет производить

вручную с учетом рода, числа, падежа и других грамматических особенностей. Команда замены служит для замены одного контекста на другой. Перед подачей задаются образец поиска и образец для замены. Команда может выполняться как по всему тексту, так и по его выделенному фрагменту. Такая команда может быть использована, например, для замены в текстовом документе слова "ПЭВМ" на слово "ПК".

Команда замены часто имеет разные режимы выполнения: -Одноразовая и глобальная замена. Режим одноразовой замены прекращает поиск после нахождения первого заменяемого элемента. Глобальная замена заменяет все вхождения заменяемого элемента. -Автоматическая и ручная замена. Режим ручной замены требует подтверждения пользователя на замену после нахождения каждого заменяемого элемента. Автоматический режим такого подтверждения не требует.

- Чувствительность и нечувствительность к строчным и прописным символам. При выборе режима, чувствительного к строчным и прописным символам, регистр, в котором отображен заменяемый элемент, имеет значение при его поиске. Иными словами, если вы задали заменяемый элемент как, текстовый процессор не обратит внимания на встретившееся ему в тексте слово 1ВМ. -Направление поиска. Обычно команда поиска и замены реализует поиск, начиная от позиции текста, занимаемой в настоящее время курсором, и до конца документа. У ряда текстовых процессоров направление поиска можно изменить на обратное.

В ряде текстовых процессоров (например, \Л/ого!) с помощью специальных символов реализуется язык запросов, дающий возможность осуществлять контекстный поиск по сложным критериям.

Операции сохранения. Операция сохранения записывает отредактированный документ, находящийся в оперативной памяти, на диск для постоянного хранения. Тип сохраненного документа обычно присваивается текстовым процессором автоматически. Например, в текстовом процессоре \Л/ог<3 документу при-

сваивается тип .ООС. Большинство текстовых процессоров используют следующие три операции сохранения данных.

1. Сохранить и продолжить редактирование. Эта операция выполняется периодически, гарантируя сохранность более свежей копии создаваемого документа на случай возможной его утраты.

2. Сохранить и выйти. Мы используем эту операцию для сохранения отредактированного документа и выхода в операционную систему.

3. Выйти без сохранения. Эта операция позволяет выйти в операционную систему без сохранения документа, с которым работали.

В текстовых процессорах, имеющих многооконный режим работы, предусматриваются одна операция выхода и отдельные операции сохранения и закрытия окон. Если при выходе в закрываемом окне остается несохраненный измененный документ, то текстовый процессор дополнительно запрашивает, следует ли сохранить документ или нет.

Текстовые процессоры с помощью резервных файлов обеспечивают защиту созданных документов от возможной утраты. Для этого специальной командой сохранения обеспечивается режим, когда одновременно хранятся два файла одного и того же документа - текущий и резервный. После внесения изменений в документ предыдущая его версия автоматически сохраняется как резервный файл: ему присваивается расширение .ВАК, а отредактированная версия рассматривается как текущий файл. Тип текущего файла определяет сам пользователь или он присваивается по умолчанию (например, .TXT, или .ООС). Таким образом сохраняется только последняя, измененная версия документа; более ранние версии теряются. Хотя содержание текущего и резервного файлов не одинаково, возможность использования последнего (в случае повреждения или утраты текущего файла) представляется очень ценной. Автоматическое сохранение резервных файлов может оказаться недостаточным для обеспечения защиты важных документов и программ. В этом случае создаются дополнительные копии, хранящиеся отдельно от основных.

Важным фактором защиты создаваемых документов является функция автосохранения, которая может выполняться как обычная операция сохранения или как специальная операция сохранения текущего состояния текстового процессора в специальном файле. В последнем случае при аварийном прекращении работы это состояние может быть восстановлено, включая содержимое всех окон, положение курсоров в окнах и т.п.

Сохраняйте результаты своей работы. Для этого каждые 10-15 минут выполняйте операцию сохранения или установите режим автосохранения, если он имеется в текстовом процессоре.

Проверка правописания слов и синтаксиса. Режим проверки правописания и синтаксиса выполняется специальными программами (ЗреИег/ спесКег), которые могут быть автономными, например Орфо, либо встроенными в текстовый процессор. Эти программы значительно различаются по своим возможностям. Наиболее мощные из них проверяют не только правописание, но и склонение, спряжение, пунктуацию и даже стиль. Указанный режим используют для контроля одного слова, страницы или целого документа. Указанное слово сопоставляется с его написанием в словаре и в случае любых несоответствий выдается на экран для редактирования. При этом пользователю предлагается следующий выбор:

1) провести исправление;
2) игнорировать ошибку;
3) добавить данное слово во вспомогательный словарь.

Многие текстовые редакторы предлагают дополнительные услуги (например, варианты написания слова), облегчающие исправление ошибок. Но помните, что возможности программы зависят от полноты словаря. Поэтому постоянно пополняйте вспомогательный словарь, внося в него слова, отсутствующие в исходном словаре.

Кроме проверки ошибок пунктуации и выдачи предложений по их устранению этот режим обеспечивает выявление некоторых ошибок стиля, в частности неправильное использование заглавных и строчных букв, повторение одного и

того же слова несколько раз подряд, отсутствие пробела между словами, отсутствие второй кавычки и т. п. Все указанные ошибки выявляются на основе сравнения разработанного текста с хранящимися в памяти основными правилами. Заметим, что используемый в рамках текстового процессора набор таких правил существенно ограничен.

Словарь синонимов. Словарь синонимов поможет избежать повторений и сделает элегантным ваш стиль изложения. Его использование чрезвычайно просто. Наведите курсор на интересующее вас слово и введите команду просмотра словаря синонимов. Текстовый редактор выдаст на экран несколько синонимов. Выберите тот, который вам больше нравится, и он автоматически будет перенесен в документ.

Установка общих параметров страницы. Различают логическую и физическую страницы. Физическая страница (в некоторых текстовых редакторах вместо этого термина используется термин "размер бумаги") обычно имеет некоторый стандартный размер, например 210x297 мм (формат А4), а логическая страница образуется на поле физической за вычетом установленных пользователем границ (рис. 1.6). Количество данных на логической странице определяется, с одной стороны, плотностью печати (количеством знаков на строке), а с другой - разреженностью строк (интервалом между строками). Если вы собираетесь работать с двойными листами, конвертами или наклейками, вам следует установить новые размеры физической страницы.

Аналогично тому, как осуществляется жесткий и мягкий перевод каретки на новую строку, текстовый процессор осуществляет мягкий и жесткий переход на новую страницу.

Рис. 1.6. Соотношение логической и физической страниц

Мягкий переход осуществляется автоматически после заполнения последней строки на странице. Однако пользователь может начать новую страницу, не дожидаясь ее окончательного заполнения, подав команду жесткого перехода.

Существует также команда запрета разрыва страниц, которая используется, когда вы хотите, чтобы определенная часть документа (например, таблица) находилась на одной странице. В этом случае вы должны поставить команду

запрета разрыва страниц перед интересующей вас таблицей, чтобы предотвратить ее разрыв.

При разрыве абзаца многие текстовые процессоры обеспечивают контроль за так называемыми висячими строками. Висячей строкой называется первая строка или заголовок нового абзаца, оказавшийся на последней строке страницы, или последняя строка абзаца, оказавшаяся в начале страницы (Огрпап). Размещение абзаца при его разрыве может регулироваться по-разному. Например, не менее двух строк в конце страницы и не менее трех строк в начале.

Для введения нумерации страниц в создаваемом вами документе текстовый процессор предложит специальное меню, в котором вы сможете указать все интересующие вас условия нумерации: месторасположение на листе номера страницы, отказ от нумерации первой страницы, использование колонтитулов и другие. Номера страниц проставляются в колонтитуле.

Колонтитулом называется заголовочное данное, помещаемое в начале или конце каждой страницы документа. Колонтитулы обычно содержат номера страниц, название глав и параграфов, название и адрес фирмы и т.п. Колонтитулы могут различаться для четных и нечетных страниц, а также для первой страницы и последующих. Использование колонтитулов позволяет лучше ориентироваться в документе, а также использовать дополнительные возможности рекламы.

Использование шаблонов и стилей. Для унификации структуры и внешнего вида документов используются шаблоны. Шаблоном называют специальный вид документа, представляющий основные средства форматирования создаваемого документа.

Шаблон как понятие включает в себя ряд элементов:

- текст или форматирование, которые одинаковы в каждом документе определенного типа;
- стили;
- элементы автотекста;
- макрокоманды;

- панели инструментов;
- набор меню и "горячих клавиш".

Шаблон можно считать своего рода пустым документом с заданными свойствами, который "накладывается" на создаваемый документ (или на основе которого строится новый документ). Задаются как свойства собственно документа, так и свойства и способы работы с документом. Наличие шаблона облегчает создание, редактирование и оформление документа, позволяет сравнительно просто создавать однотипные документы.

Шаблоны позволяют составлять и хранить универсальные бланки документов различного типа: писем, служебных записок, доверенностей, платежных поручений. Составной частью шаблонов являются стили. Стиль как элемент шаблона предназначен для внешнего оформления документа и его абзацев. Стиль имеет имя, задавая которое пользователь задает определенный набор параметров форматирования, собранных вместе.

Стиль - это набор форматирующих команд, сохраняемый под своим именем для многократного использования. Стили упрощают задачу оформления и изменения вида документа, обеспечивают согласованность представления отдельных частей документа или документов одного типа, экономят время на процесс форматирования.

Текстовый процессор \Л/ог<3 обладает большим числом стандартных (встроенных) стилей, часть из них являются стилями символов, остальные -стилями абзацев. Список доступных стилей зависит от шаблона, выбранного в момент создания документа. По умолчанию во всех новых документах, основанных на шаблоне использует стиль Обычный, в котором установлены основные параметры форматирования: размер шрифта - 10 пунктов, выравнивание по левому краю, одинарный межстрочный интервал.

Начиная составлять определенный документ, вы сначала вызываете шаблон этого типа документов, а уже затем заполняете его. Составление документа при этом сводится к заполнению его определенных полей текстом. Один раз сде-

ланный на основе стандартов шаблон может в дальнейшем многократно использоваться для создания документов определенного вида.

Использование макросов. Макросом называют файл, в котором хранится программа последовательности действий, заданная пользователем. Макрос имеет уникальное имя.

С помощью макросов можно автоматизировать многие типовые технологические этапы при работе с документами, например, макрос, выполняющий последовательность команд по созданию стиля для каждого абзаца документа. После вызова макроса записанная в нем последовательность действий или команд будет в точности воспроизведена.

Макрос создают двумя способами:

- автоматически в специальном режиме текстового процессора путем записи последовательности действий пользователя;
- программированием последовательности команд, подлежащих выполнению.

Макрос может храниться в самом файле документа или в специальном стилевом файле как элемент общего окружения документа.

Слияние документов. В мощных текстовых процессорах имеется возможность слияния документов. Для выполнения этой процедуры необходимо иметь:

- главный документ, содержащий постоянную информацию; -документ-источник для хранения переменной информации.

Главным документом может быть стандартная форма справки или приглашения, стандартное письмо, надписи на конверте, то есть форма любого стандартного текста и так далее.

Документ-источник должен представлять собой базу данных (или таблицу), состоящую из однотипных записей. В документе-источнике содержатся данные, которыми будет заполняться главный документ.

Рассмотрим идею слияния документов на примере.

Писать письма многочисленным адресатам одного и того же содержания - малоинтересное занятие. Текстовый процессор автоматизирует этот трудоемкий

процесс, соединяя постоянную часть письма с данными, характеризующими адресатов. При этом пользователь должен создать специальный адресный файл. Адресный файл состоит из однотипных записей с именами и адресами клиентов. Каждая запись включает в себя определенное число полей, в которые записываются данные каждого клиента. Так, в первое поле всегда записывается фамилия, во второе - имя, в третье - отчество, в четвертое -адрес и т. п.

Допустим, вы хотите разослать письма всем абитуриентам, сдавшим экзамены и поступившим в Университет в этом году. Для этого надо подготовить общую часть письма и адресный файл всех зачисленных абитуриентов (рис. 1.7). В общей части письма специальными операторами (на рис. 1.7 - это Р1", Р2", РЗ" и Р4") вы укажете, где и какие поля из адресного файла должны быть помещены для каждого получателя. В результате текстовый процессор автоматически напечатает все письма, содержащие помимо основной части адрес и данные каждого абитуриента.

Работа издательских систем

1) Особенности издательских систем

Компьютерная издательская система призвана на высоком техническом уровне создавать книги, каталоги, буклеты, отчеты, письма, приглашения и другую печатную продукцию. Отличительными особенностями редакторов издательских систем является наличие средств для подготовки текста, иллюстраций, графиков, для использования различных шрифтов, других оформительских элементов. Издательские системы имеют развитые средства для организации размещения материала разнотипного характера на странице. Многими качествами издательских систем обладает редактор Йога[1], широкое распространение получают издательские системы РадеМаКег, Уеп1:ига РиЬМзпег, (гйепеа? РиЬИзпег. - В традиционной полиграфии, чтобы увидеть свой заказ в окончательном виде, вам придется работать с дизайнером несколько дней, а может быть и больше, потом внести свои коррективы и ждать еще столько же. В издательской системе вы видите спроектированный документ на экране монитора сразу же в окончательном виде.

- В традиционной полиграфии текст вашего заказа набирается наборщиком. Помимо того, что набор требует немалого времени, в текст вносят дополнительные ошибки. В издательской системе вы импортируете свой текст из текстового процессора.

- В полиграфии выполненные вами рисунки проходят стадию уменьшения, чтобы поместиться на странице. В издательской системе уменьшение рисунков до необходимых размеров и вставка их в текст легко производятся вами.

Компьютерная издательская система предоставляет пользователю огромные возможности для создания высококачественной полиграфической продукции.

Компоненты издательской системы. Компьютерные издательские системы основываются на сочетании современных компьютеров, лазерных принтеров и сложного программного обеспечения, дающего возможность манипулировать текстом и графикой. Центральным элементом этих систем следует считать специализированные пакеты прикладных программ, позволяющие создавать требуемый текст на экране монитора. Поскольку граница между высшим уровнем существующих текстовых процессоров и низшим уровнем используемых издательских систем постепенно стирается, весь изложенный в данном разделе материал по издательским системам может быть в равной степени применим и к высокоразвитым текстовым процессорам.

Помимо лазерных принтеров необязательным, но весьма желательным элементом издательских систем являются сканеры. Сканеры конвертируют любую картинку, фотографию или текст в файл одного из графических форматов, обеспечивая тем самым возможность их вставки в проектируемый документ.

Сравнение текстового редактора и издательской системы. Текстовый редактор и издательская система призваны решать одну и ту же задачу создания документов, но они ориентируются на различные категории пользователей. Текстовый процессор предназначен для рядового пользователя, для которого главным является удобство и простота интерфейса. Издательская система ориентируется на профессионалов, для которых важнее всего полный контроль над видом создаваемого документа. Указанные ориентации постепенно сбли-

жаются по мере совершенствования компьютеров и программ. На сегодняшний день различия проявляются лишь в деталях. Например, текстовый процессор не имеет экспорта в формате, обязательного для издательских систем, которые должны иметь развитую систему импорта-экспорта документов.

С другой стороны, издательские системы могут уступать текстовым процессорам в качестве изображения документа на экране монитора, поскольку для первых из них более важным является качество вывода на печать (или другое специализированное устройство вывода).

Создание издательской системы вызвано особыми требованиями работников, связанных с книгоизданием:

-способностью системы создавать газетные колонки;

- способностью изменять шрифты в пределах одного документа;

- способностью включать графику в текст;

- возможностью просмотра разработанного документа до его распечатки.

Поскольку высокоразвитые текстовые процессоры сегодня обладают всеми указанными свойствами издательских систем, в ряде случаев лучше говорить не о дополнительных возможностях этих систем, а о том, насколько проще (или труднее) их использовать по сравнению с текстовыми процессорами для выполнения тех же самых функций.

Режимы работы с текстом и графикой. Все известные текстовые процессоры и издательские системы могут работать в двух режимах -текстовом и графическом.

В текстовом режиме экран дисплея можно условно рассматривать как совокупность прямоугольников, расположенных вплотную друг к другу. Как правило, экран содержит 2000 таких прямоугольников, составляющих 25 строк по 80 прямоугольников в каждой. Каждый прямоугольник - это символьная позиция экрана, в которой может находиться один символ: строчные и прописные буквы алфавита, числа, знаки пунктуации и специальные символы. Множество используемых символов в текстовом режиме работы дисплея определяется кодовой таблицей, находящейся в ПК.

В графическом режиме экран дисплея можно представить состоящим из отдельных точек, имеющих свои координаты. Высвечивая определенные точки, мы получаем на экране тот или иной графический образ. Точность изображения определяется разрешающей способностью выбранного режима экрана, задается в виде произведения количества точек по горизонтали и вертикали экрана. Так, например, экран монитора типа УСА в графическом режиме разделяется на 480 строк по 640 точек в строке (всего на экране более 300000 точек, из которых, как из мозаики, формируется изображение).

Графический режим незаменим при подготовке сложных формульных текстов, для работы с шрифтами разных размеров, с иероглифами и т.д. Конечно, графический режим позволяет выводить и все символы из кодовой таблицы ПК, при этом каждый из них изображается с помощью матрицы размером 8Х8 точек.

Преимуществом текстового режима являются его большая скорость и меньшая потребность в оперативной памяти. Недостатком - способность изображать только 256 знаков, находящихся в определенных ячейках экрана. Следствие указанного недостатка - это невозможность просмотра в текстовом режиме (текстового процессора или издательской системы) графических образов. Так, если в вашем документе имеется картинка и вы работаете в текстовом процессоре, вы должны создавать документ (его текстовую часть) в текстовом режиме, затем его просматривать в графическом режиме и вносить изменения снова в текстовом режиме.

Издательские системы почти постоянно работают в графическом режиме, поскольку они допускают внесение изменений в текст и в графическом режиме. Основной целью издательских систем является не ввод текста, а верстка.

В заключение отметим, что следует взвешенно относиться к возможностям издательских систем. Это дорогие, большие и сложные программные продукты, требующие значительного времени и усилий как на освоение, так и на использование. Для перехода с текстовых процессоров на издательские системы должны быть веские основания. Даже при полном освоении всех возможностей

издательских систем не ожидайте автоматического повышения художественного уровня вашей печатной продукции. Для этого потребуется немало времени и сил, чтобы развить хороший оформительский вкус и освоить традиции книгоиздания.

Основы создания документа

ШРИФТЫ. Одним из основных вопросов создания документа является выбор типа шрифта, его стиля и размера, а также определения требований к плотности печати на строке и странице. Важность этих решений связана с тем, что успех того или иного документа в большой степени зависит от того, как он выглядит.

Выбор шрифта должен соответствовать информации, которую вы хотите передать читателю: шрифт не должен быть ни слишком заметным, ни слишком ординарным. При этом различные части документа (заголовок, сам текст документа, примечания) нужно набирать шрифтами различных размеров. Правильный выбор шрифта, его стиля (жирный, курсив, обычный) и размера акцентирует содержание сообщения, не привлекая внимания к самому шрифту.

Шрифт представляет собой комплект букв, цифр и специальных символов, оформленных в соответствии с едиными требованиями. Шрифты отличаются друг от друга рисунком (гарнитурой), начертанием (прямой, курсивный), и размером (кеглем).

Назначение шрифта - представить документ в удобном для чтения виде. Красивый шрифт как произведение искусства способствует лучшему художественному оформлению текста. Известно немало случаев, когда для какого-то конкретного издания разрабатывался специальный шрифт. Например, для нового издания библии по заказу Российского Библейского общества разработан шрифт Октава (Скриптура Руссика).

Типы шрифтов. Тип шрифта в полиграфии принято называть гарнитурой шрифта. Гарнитура определяет набор художественных решений, отличающий данный шрифт от других. Каждая гарнитура имеет определенное название, например, Т1тез, НеК/ейса, Соипег.

Каждый тип шрифта имеет несколько начертаний символов, например, полужирный, курсив, полужирный курсив, обычный. Кроме того, можно ввести подчеркивание символов и фрагментов текста.

Размеры шрифтов. Размер шрифта или иначе кегль - это его вертикальный размер, измеряемый в пунктах (1 пункт равняется 0,376 мм). Для большинства документов используются шрифты в 10-12 пунктов, в то время как газетная полоса может иметь шрифт в 8 пунктов. Размер шрифта более 14 пунктов обычно используется лишь для заголовков и выделений. Заметим, что шрифты различных типов, но одного размера необязательно выглядят одинаковыми по величине.

Разреженность строк и символов. Низкая разреженность строк (т.е. небольшое расстояние между соседними строками) затрудняет чтение, а высокая - делает документ малоинформативным. Для правильного выбора разреженности строк, также измеряемой в пунктах, необходимо учитывать размер используемого шрифта. В современных издательских системах разреженность строк определяется автоматически на уровне 120% от выбранного размера шрифта. Во многих текстовых процессорах имеется возможность регулировать расстояние между символами в словах, создавая более уплотненное или разреженное их начертание. Такой прием часто применяется для заголовков.

Растровые шрифты. При работе с документами на персональных компьютерах применяют растровые и векторные шрифты. Растровые шрифты строятся в виде битового массива точек - растра. Каждая буква растрового шрифта представляется как группа закрашенных квадратиков (точек) на сетке растра (рис. 1.8).

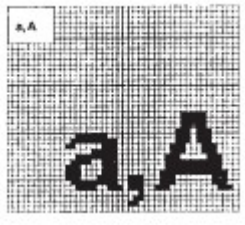


Рис. 1.8. Шрифтовой растр

Растровые шрифты не масштабируются. При попытке увеличения размера растровых шрифтов происходит резкое ухудшение их качества - лестничный эффект, поэтому приходится хранить отдельную копию шрифта для каждого используемого размера. Растровые шрифты просты в отображении, поэтому применяются в качестве экранных шрифтов, во всех матричных и в большинстве лазерных и струйных принтерах, на графопостроителях. При отображении растровых шрифтов в натуральную величину они дают более высокое качество изображения, нежели векторные.

Векторные шрифты. Векторные шрифты хранятся в виде набора математических уравнений, описывающих начертание символа. Они не содержат готовых растров символов, поэтому каждый раз при их использовании растры должны строиться заново. Векторные шрифты не имеют лестничного эффекта при масштабировании, что позволяет увеличивать или уменьшать их размер в любых разумных пределах.

Шрифты твуетуре. Из-за различной разрешающей способности принтера и монитора изображения символов на экране и в напечатанном документе могут отличаться.

Шрифты ТшеТуре (в переводе - верная печать) обеспечивают полное соответствие документа на бумаге и на экране. Они относятся к масштабируемым шрифтам и все размеры шрифтов от 4 до 127 пунктов не имеют искажений.

Верстка страниц

Общие сведения. Верстка страниц играет такую же роль, как чертеж перед началом строительства дома. Верстка определяет организацию страницы, задавая соотношения между границами, колонками и расстоянием между колонками. Первое важное решение на уровне верстки связано с выбором количества колонок на странице. Одноколоночная верстка обычно используется для книг и писем, в то время как двух- и трехколоночныи формат больше подходит для журналов и газет. Далее размеченные колонки наполняются текстом и рисунками. На рис. 1.9 приведены варианты верстки документа.

Рис. 1.9. Различные варианты верстки документа

Ширина строк. Ширина строки (колонки) определяется количеством знаков, которые могут быть на ней помещены. Обычно оптимальной считается ширина строки в пределах от 45 до 60 символов.

Обращаем внимание на наличие связи между шириной строки (колонки) и размером выбранного шрифта: чем меньше размер шрифта, тем короче строка. Иными словами, меньший размер шрифта дает возможность поместить больше символов на заданной площади листа.

Вставка рисунков. В процессе верстки рисунки так же, как и текст, размещаются в границах установленных колонок. При размещении рисунков следует соблюдать ряд правил.

- Располагайте рисунки сразу после упоминания о них в тексте. -Оставляйте по сантиметру пространства между текстом и рисунком.

- Старайтесь использовать рисунки той же ширины, что и сопровождающий их текст.

Заметим, что для придания странице большей выразительности возможно расположение рисунков, выходящих за рамки колонки (т.е. рисунков, расположенных вне спроектированной верстки).

Дополнительные принципы проектирования. Основой проектирования документов является верстка. Существует ряд дополнительных принципов проектирования документов, на которых нам хотелось бы остановиться.

Что можно и что нельзя в верстке. Давние традиции книгопечатания сформулировали следующие рекомендации:

- Не пишите убористо. Оставляйте на листе достаточно пустого места, чтобы дать глазу передышку. Между краями листа и текстом оставляйте хотя бы по паре сантиметров.
- Организуйте элементы вашего документа так, чтобы он выглядел единым целым, занимал при этом минимальную площадь страницы.
- Не используйте на странице много различных шрифтов и рисунков: помните, что они лишь украшение текста.
- Не используйте слишком длинных строк, длинные строки утомляют глаз читателя.

Принцип баланса. Соблюдение принципа баланса требует, чтобы визуальный вес левой и правой частей страницы (или левой и правой страниц разворота) был примерно одинаков. При наличии нескольких рисунков на странице достижение баланса превращается в нелегкую задачу.

Выделение. Чтобы акцентировать внимание читателя, издательские системы предлагают целый набор средств. Простейшие из них - изменение типа и размера шрифта. Заголовки следует набирать шрифтом большего размера, чем подзаголовки, которые, в свою очередь, должны быть больше основного текста. Также эффективно использование стилей шрифтов -полужирного, курсива и подчеркивания, однако, ими следует пользоваться умеренно. Не следует выделять заглавными буквами или подчеркивать длинные фрагменты текста. Важным средством выделения является заполнение фрагмента серым (цветным) тоном и/или выделение фрагмента в рамку.

Полезные советы работающим с текстовым процессором

Используйте режим просмотра готового документа перед его распечаткой. Многие текстовые редакторы предлагают вам заблаговременно (до печати) посмотреть, как будет выглядеть готовый документ после его распечатки на бумаге. Часто своевременное использование этого режима дает возможность лучше организовать документ, не допуская его перепечатки.

Учитесь быстро печатать. Главным ограничением в производительности текстового редактора является скорость ввода данных с клавиатуры. Не пожалейте времени и сил, чтобы научиться печатать "слепым методом" (не глядя на клавиатуру) и десятью пальцами. Рекомендуем вам использовать для этого специальные компьютерные программы-тренажеры.

Сначала пишите, потом редактируйте. Создавая документ, не пытайтесь редактировать текст все время. Опыт показывает, что при этом падает производительность и теряются мысли.

Удаляйте с осторожностью. Вместо того чтобы удалять текст большими кусками, мы советуем вам лучше сдвигать эти куски в конец документа или переписывать их во временные файлы. При создании документа работайте в режиме вставки, не связанной с возможной потерей данных.

Сохраняйте чаще. Перебои в снабжении электроэнергией, зависание программных продуктов и другие компьютерные неприятности могут свести на нет многие часы и дни вашей работы. Поэтому хорошей привычкой является постоянное использование команды сохранения, особенно в случаях, когда вас позвали к телефону или кто-то позвонил в дверь.

Храните копии всех важных документов. Вы можете создавать резервные файлы, переписывать ваши документы на отдельные дискеты или просто распечатывать их в конце сеанса работы.

Используйте небольшие файлы. Большие файлы обрабатываются текстовыми процессорами значительно дольше, чем маленькие. Кроме того, в случае повреждения или потери файла ущерб от маленького файла будет меньше.

Сделайте, чтобы вам было удобно. Проверьте, удобно ли вам сидеть за компьютером. В удобной позе вы дольше сохраните состояние работоспособности.

Делайте передышки. Короткие перерывы в работе помогут вам и вашим глазам не уставать в процессе работы.

Тема 9. Универсальные пакеты прикладных программ для обработки данных (2 часа)

Лекция №15

Вопросы: Электронные таблицы (Excel) – основные понятия, функциональные возможности, технология работы. Пакеты для инженерных и научных расчетов (Mathcad) – основные понятия, технология работы.

Технология работы с электронной таблицейю

1) Обобщенная технология работы

Хотя работа в каждой электронной таблице имеет свои особенности, можно говорить о некоторой типичной технологии работы с ней. Схема такой технологии приведена на рис. 1.9.

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

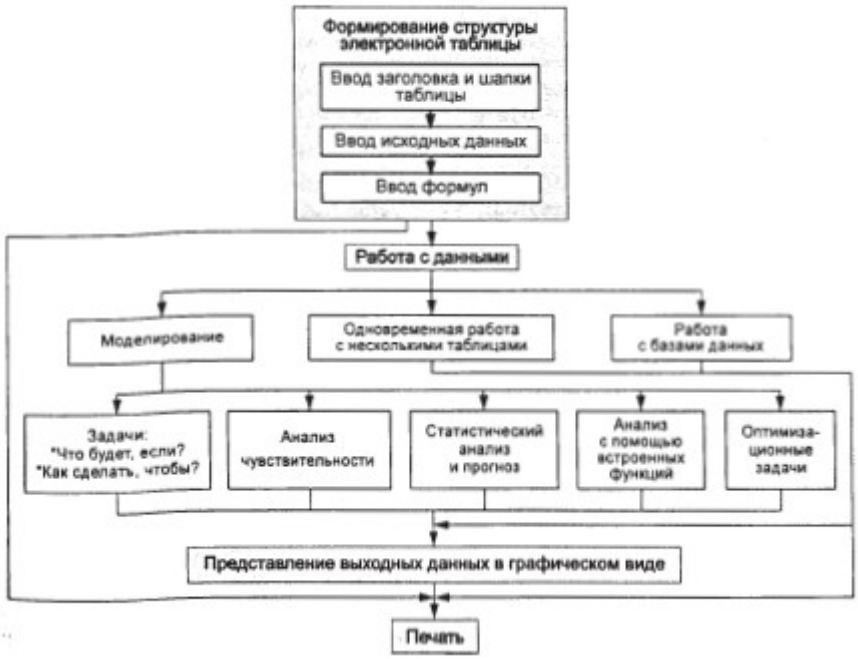


Рис. 1.9. Обобщенная технология работы с электронной таблицей

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

Математические модели помогают пользователю на основе имеющейся таблицы получить новую информацию решением типовых задач компьютерного моделирования: "Что будет, если ?", анализ чувствительности и др. Решение задач типа: "Что будет, если?" дает возможность пользователю узнать, как изменятся выходные параметры при изменении одной или нескольких входных величин (условий). Расширением таких задач являются задачи анализа чувствительности, позволяющие определить, как изменится выходной параметр при изменении одной или нескольких входных величин с заданным шагом в определенном диапазоне значений. Обратной по отношению к задаче: "Что будет, если?" является задача: "Как сделать, чтобы?". Она возникает в случае, если ваша цель - достижение определенной выходной величины и вы ищете значения входных параметров, обеспечивающих достижение этой цели. Различные виды анализа данных, содержащихся в исходной таблице, можно проводить с использованием встроенных функций и процедур. Так, входящие в состав электронной таблицы статистические функции могут использоваться в статистическом анализе или для прогноза содержащихся в таблице данных. Использование финансовых функций позволяет произвести анализ эффективности планируемых капитальных вложений, рассчитать стоимость ценных бумаг или размер амортизационных отчислений. Для решения оптимизационных задач используются специальные модели математического программирования.

Часто фирма имеет центральный офис и несколько филиалов. В таких условиях появляется задача объединения различных документов и отчетов, приходящих из этих филиалов. Решение подобной задачи требует использования специальных многотабличных связей и программных методов для манипулирования с

файлами и генерации отчетных форм. Одновременная работа с несколькими таблицами - это одна из возможностей работы с данными в электронных таблицах.

Иногда при работе с большими электронными таблицами требуется найти ту или иную строку (столбец) или отсортировать строки (столбцы) в нужном порядке. Для этого в электронной таблице предусмотрен ограниченный программный инструментарий СУБД, позволяющий манипулировать строками и столбцами как компонентами баз данных.

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

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

2) Проектирование электронной таблицы

Рассмотрим проектирование электронных таблиц на примере формирования прогноза финансовой деятельности некоторой компании за 5 лет (табл. 1.2). Доход в любом году определяется как произведение объема продаж в натуральном выражении на продажную цену. Размер прибыли при этом рассчитывается как разность между доходом и суммой расходов данного года.

Пользователем задаются исходные данные первого года. Данные всех последующих лет вычисляются электронной таблицей на основе допущений о характере их изменений в будущем. Эти данные расположены в нижнем левом углу табл. 1.2. Расходы и продажные цены определяются с учетом заданного роста цен, а объемы продаж - с учетом роста объемов продаж. При изменении данных допущений электронная таблица должна немедленно пересчитывать значения всех прогнозируемых финансовых характеристик. Для этого электронная таблица проектируется таким образом, чтобы прогнозируемые параметры определялись по формулам, зависящим от исходных прогнозных допущений.

Проектирование электронной таблицы, расчеты по которой мы видим в табл. 1.2, основано на использовании уже рассмотренных нами команд копирования и перемещения.

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

Создание электронной таблицы начинается с ввода названий столбцов и строк. Столбцами в нашей таблице являются годы, составляющие прогнозный период, а строками - прогнозируемые финансовые показатели. Кроме того, в таблицу входят прогнозные допущения.

Далее вводятся данные первого года (в нашем примере 1995), которые частично задаются как числовые данные (объем продаж, продажная цена), а частично - как формулы. Так, например, в ячейку B5 записывается формула для вычисления дохода +B3*B4, а в ячейку B7 - формула для вычисления прибыли +B5-B6. Одновременно вводятся числовые данные, выражающие прогнозные допущения.

Таблица 1.2

Финансовый прогноз деятельности компании

	A	B	C	0	E	P
1	Прогноз деятельности			компа-		
				нии		
2		1995	1996	1997	1998	1999
3	Объем продаж,	10000	1180	13924	16430	19388
	шт.		0			
4	Цена	$2.00	$2.10	$2.21	$2.32	$2.43
5	Доход	$2000	$247	$30702	$3804	$47132
		0	80		0	
6	Расходы	$1500	$157	$16537	$1736	$18232
		0	50		4	
7	Прибыль	$5000	$903	$14165	$2067	$28900
			0		6	
8						

9	Прогнозные до-	
	пущения	
1	Рост объема про-	18.00
0	даж	%
1	Рост цен	5.00%
1		

Наиболее сложный момент проектирования нашей таблицы - это ввод формул в столбец второго года (1996). Эти формулы учитывают результаты первого года и, кроме того, отражают прогнозные допущения. Так, например, объем продаж в 1996 г. определяется как объем продаж 1995 г., увеличенный с учетом процента роста, указанного в прогнозных данных (рис. 1.10).

Использование в указанной формуле относительных и абсолютных адресов позволит скопировать ее в оставшиеся колонки. Абсолютный адрес ячейки, содержащей процент роста объема продаж, используется в формулах расчета объемов продаж во все годы, начиная с 1996 г. Относительный адрес ячейки, содержащей объем продаж предыдущего года, дает возможность его подстройки при копировании формулы, поскольку сохраняется логика расчета объема продаж для последующих лет.

Рис. 1.10. Использование абсолютных и относительных адресов

Пересчет остальных параметров из столбца В в столбец С выполняется аналогичным образом.

Столбцы О, Е, Р заполняются простым копированием формул, содержащихся в столбце С. Команда копирования при этом автоматически подстроит содержащиеся в них относительные адреса ячеек.

Построенная электронная таблица дает возможность создавать финансовые прогнозы, изменяя прогнозные допущения. Вы можете, например, изменив

одно или несколько прогнозных допущений, определить, что произойдет с прибылью в 1999 г. (табл. 1.3).

Полученные результаты могут быть также представлены в графическом виде.

Таблица 1.3

Электронная таблица для финансового прогнозирования в режиме просмотра формул

	A	B	C	0	E	P
1	Прогноз деятельности компании					
2		1995	1996		1998	
			1997		1999	
?	Объем продаж	10000	(1+$B$10)*B3		(1+$B$10)*E	
	шт				3	
/\	Цена	$200	(1+$B$11)*B4		(1+BИ)*E4	
^	Доход	+B3*B4	+C3*C4		+P3*P4	
в	Расходы	15000	(1+$B$11)*B6		. . .	
			. . .		(1+$B$11)*E	
					6	
7	Прибыль	+B5-B6	+C5-C6		. . . +P5-P6	
8						
9	Прогнозные					
	допущения					
10	Рост объема	18.00%				
	продаж					
11	Рост цен	5.00%				

3) Объединение электронных таблиц

При работе с электронными таблицами часто возникает необходимость их объединения. Среди инструментов объединения электронных таблиц отметим:

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

Организация межтабличных связей

Связи между таблицами осуществляются путем использования внешних ссылок (адресов ячеек), содержащих помимо имени столбца и номера строки имя файла, данные из которого используются. Так, например, если мы хотим ис-

пользовать данные из ячейки С2 таблицы, содержащейся в файле, в нужную нам ячейку текущей таблицы мы можем записать внешнюю ссылку следующим образом.

При организации межтабличных связей учитывают возможность комплектования связанных таблиц в рабочую книгу. При этом таблица, на которую есть внешние ссылки, рассматривается как дополнительная. Таблица, в ячейках которой есть внешние ссылки на другие таблицы, считается основной. При загрузке таблицы, содержащей внешние ссылки, необходимо также загрузить все связанные с ней вспомогательные таблицы. В противном случае в ячейках основной таблицы, имеющих внешние ссылки, появятся сообщения об ошибке или представленные вам результаты окажутся неверными. Между отдельными таблицами возможны двусторонние связи (таблица А ссылается на таблицу В, а В, в свою очередь, прямо или опосредованно, например через таблицу С, ссылается на А).

Консолидация электронных таблиц или их частей

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

Команда объединения файлов

Многие современные табличные процессоры имеют в своем арсенале команду объединения файлов. Эта команда имеет три формы, используемые для копирования, суммирования или вычитания данных из исходных таблиц в объединенную таблицу. Технология создания электронной таблицы, объединяющей данные нескольких исходных таблиц, такова: мы создаем электронную таблицу в оперативной памяти и засылаем в нее данные из исходных электронных

таблиц, находящихся на жестком (или гибком) диске. Процесс начинается с подготовки шаблона объединенной электронной таблицы.

Пример. Допустим, что интересующая нас компания имеет три магазина, от которых она получает регулярные отчеты в форме электронных таблиц. Однако, если в целом дела идут нормально, руководство компании мало интересует финансовая деятельность каждого из магазинов. Ему хотелось бы увидеть результаты деятельности всей компании и уяснить, какой из магазинов приносит прибыль, а какой - убытки. Здесь возникает задача объединения данных из трех отчетов (электронных таблиц) в один. Эта задача может быть решена установлением межтабличных связей или объединением файлов электронных таблиц.

Допустим, что исходные отчеты, поступающие от магазинов компании, имеют вид, указанный в табл. 1.4.

Для объединения данных по прибыли из нескольких отчетов, поступающих от различных магазинов, создается объединенный отчет, подобный указанному в табл. 1.5.

В ссылках, находящихся в ячейках этого отчета, указываются имена исходных файлов, содержащих данные о каждом из магазинов.

При использовании команды объединения файлов в режиме копирования в результате объединения данных по прибыли из нескольких отчетов, поступающих от различных магазинов, будем иметь объединенный отчет, подобный указанному в табл. 1.6.

В ряде случаев полученные в табл. 1.6 данные могут оказаться недостаточными для руководства компании, которое интересуют суммарные данные не только по прибыли, но и по продажам и затратам. Здесь используется команда объединения файлов в режиме суммирования, которая обеспечивает иной порядок формирования данных в объединенном отчете. Объединенный отчет (электронная таблица) в этом случае будет формироваться в таком же виде, как и отчеты, получаемые от магазинов.

Таблица 1.4

Отчет, поступающий от одного из магазинов

	A	B	C	0	E
1	Компания 1&M Магазин #1. Данные за 1995 г. по кварталам				
2		1-й	2-й	3-й	4-й
3	Объем продаж, дол.	84,000	92,000	1 1 0,000	102,000
4					
5	Зарплата	48,000	48,000	68,000	68,000
6	Себестоимость	31 ,000	32,500	36,000	35,000
7	Суммарные затраты	79,000	80,500	104,000	103,000
8					
9	Прибыль	5,000	1 1 ,500	6,000	-500

Таблица 1.5

Объединенный отчет, полученный организацией межтабличных связей (в режиме просмотра формул)

	A	I	в	1	0		E
1	Компания 1&M. Все магазины. Данные по кварталам:						
2		1-й		2-й	3-й		4-й
3	Магазин 1	[имяфайла !]B9		[имяфайла!]C9	[имя файла1]09	[и мя	файла1]E9
4	Магазин 2	[имя файла2]B9		[имя файла2]C9	[имя файла2]09	[и мя	файла2]E9
5	Магазин 3	[имя файлаЗ]B9		[имя файлаЗ]C9	[имя файлаЗ]09	[и мя	файлаЗ]E9
6	Общая прибыль	511M(BЗ, B4,B5)		511M(CЗ,C 4.C5)	5УМ (03, 04, 05)		511M(EЗ,E 4,E5)

Таблица 1.6

Так, например, зарплата в первом квартале в объединенном отчете будет определяться суммой заработной платы во всех магазинах в первом квартале.

Аналогично используется команда объединения файлов в режиме вычитания. Этот режим может быть использован, например, в случае, когда текущие показатели вычисляются как разность показателей этого и прошлого годов.

4) Электронная таблица для поддержки принятия решений

Покажем роль электронной таблицы как средства поддержки принятия решений на примере анализа получения кредита.

Попытаемся использовать возможности электронной таблицы для решения вопроса о возможности покупки в кредит автомобиля. Допустим, вы хотите знать, "осилите" ли вы ежемесячный платеж за покупаемую машину, величина которого зависит от ее цены, первоначального платежа и условий предоставления кредита (ссуды). Иными словами, вас интересует:

- можете ли вы позволить себе определенный месячный платеж за машину?
- что будет, если вы согласитесь на меньший автомобиль и получите скидку от его производителя?
- что будет, если вы в следующее лето заработаете некоторую дополнительную сумму для первоначального платежа?
- что будет, если вы увеличите срок возврата ссуды и получите более низкую процентную ставку?

Ваше решение о выборе и покупке автомобиля зависит от ответов на эти и другие вопросы. На рис. 1.11 анализ данной ситуации проведен при помощи электронной таблицы.

Показатели	Альтернатива 1	Альтернатива 2	Альтернатива 3	Альтернатива 4
Цена автомобиля	$ 14 999	$13 999	$13 999	$13 999
Скидка производителя	$ 0	$1 000	$1 000	$1 000
Первоначальный платеж	$0	$0	$3 000	$3 000
	----------	----------	----------	----------
Полные затраты	$14 999	$12 999	$9 999	$9 999
Процентная ставка (%)	13.00	13.00	13.00	12.00
Срок возврата ссуды (годы)	3	3	3	4
Ежемесячный платеж	$505.38	$437.99	$336.91	$263.31

Рис. 1.11. Анализ ситуации с помощью электронной таблицы

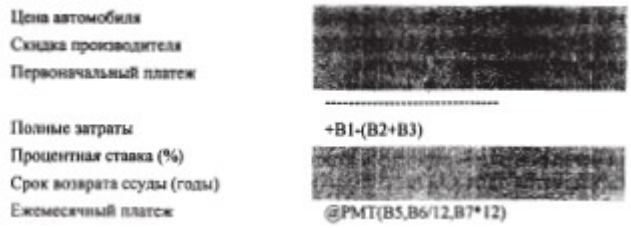


Рис. 1.12. Шаблон таблицы

На рис. 1.12 мы видим шаблон (пустую таблицу), имеющий соответствующие названия строк и столбцов, а также формулы без числовых данных. В шаблон дополнительно вводятся следующие числовые данные: цена автомобиля, скидка производителя, первоначальный взнос, годовая процентная ставка и время возврата ссуды. После ввода указанных данных электронная таблица автоматически вычисляет значение ежемесячного платежа, используя специальную функцию @PMT.

Функция @PMT(Ру,Ba1:e,Мрег) вычисляет сумму периодического платежа, необходимую для погашения ссуды РУ с процентной ставкой Ba1:e за число платежных периодов Мрег. При этом значения, которые определяются для Ba1:e, должны коррелироваться с единицами, используемыми для Мрег. Если платежи делаются ежегодно, Мрег измеряется в годах. Если платежи производятся ежемесячно, Мрег представляет собой число платежных месяцев. Для расчета ежемесячных платежей при использовании годовой процентной ставки ее следует разделить на 12. Так, например, в рамках таблицы на рис. 1.11 функция @PMT используется в следующем виде:

@PMT(14999,13/12,36).

Заметим, однако, что в электронной таблице аргументы функций могут быть представлены не самими их значениями, а адресами ячеек, в которых эти значения находятся:

@PMT(B5,B6/12,B7*12).

Указанный шаблон позволяет рассмотреть несколько альтернатив и получить представление о полезности электронных таблиц для принятия решений.

Альтернатива 1, которую мы видим, не кажется нам слишком привлекательной, поскольку покупка машины по цене 14999 дол. - это неприемлемая для нас величина ежемесячного платежа, превышающего 500 дол.

Соглашаясь на меньший автомобиль и получая при этом скидку, мы несколько уменьшаем размер ежемесячного платежа, доводя его до 437 дол. (альтернатива 2).

Далее мы видим альтернативу 3 - необходимость получения дополнительного дохода в 3000 дол. для внесения первоначального платежа.

Последняя альтернатива 4 покупки - увеличение срока возврата ссуды до 4 лет с более низкой процентной ставкой, возможно, устроит нас.

Таким образом иллюстрируется, как использовать электронную таблицу для поддержки принятия решений. Пользователь определяет проблему, вводит необходимое количество переменных и затем строит электронную таблицу в нескольких версиях, в каждой из которых варьируется одна или несколько переменных.

2. Табличный процессор MS EXCE1. 2.1. Знакомство с табличным процессором MS Excel

Табличный процессор M8 Excel используется для обработки данных. Обработка включает в себя:

- проведение различных вычислений с использованием мощного аппарата функций и формул;
- исследование влияния различных факторов на данные;
- решение задач оптимизации;
-получение выборки данных, удовлетворяющих определенным критериям;
- построение графиков и диаграмм;
- статистический анализ данных.

При запуске M3 Excel на экране появляется рабочая книга "Книга 1", содержащая 16 рабочих листов. Каждый лист представляет собой таблицу, состоящую из строк и столбцов. В этих таблицах хранятся данные, с которыми вам предстоит работать. Вы можете вводить в таблицы любую информацию: текст, чис-

ла, даты и время, формулы, рисунки, диаграммы, графики. Вся вводимая информация может быть обработана при помощи специальных функций. При помощи графических средств МЗ Excel вы можете рисовать, строить графики и диаграммы. МЗ Excel предоставляет в ваше распоряжение средства для работы с базами данных.

Тема 10. Сервисное программное обеспечение (2 часа)

Лекция №16

Вопросы: Программы архиваторы: понятие процесса архивации файлов, основные виды программ архиваторов. Помещение файлов в архив. Извлечение файлов из архива. Удаление файлов из архива. Многофункциональные интегрированные архиваторы RAR и ZIP.

10. АРХИВАЦИЯ ФАЙЛОВ 10.1. Что такое архивный файл?

Архивный файл (архив) представляет собой набор из одного или нескольких файлов, помещенных в сжатом виде в единый файл, из которого их можно извлечь в первоначальном виде. Архив содержит оглавление, позволяющее узнать, какие файлы содержатся в архиве. В оглавлении архива для каждого содержащегося в нем файла хранится следующая информация:

имя файла;

сведения о каталоге, в котором содержится файл;

дата и время последней модификации файла;

размер файла на диске и в архиве;

код циклического контроля для каждого файла, используемый для

проверки целостности архива.

Для архивации файлов используются программы-упаковщики, которые позволяют за счет применения специальных методов "упаковки" информации сжимать информацию на дисках, т. е. создавать копии нескольких файлов в один архивный файл.

Как правило, программы для упаковки файлов позволяют помещать копии файлов на диске в сжатом виде в архивный файл, извлекать файлы из архива, просматривать оглавление архива и т.д. Разные программы отличаются форма-

том архивных файлов, скоростью работы, степенью сжатия файлов при помещении в архив, удобством использования.

Наиболее распространенные программы-упаковщики имеют приблизительно одинаковые возможности и ни одна из них не превосходит другие по всем параметрам: одни работают быстрее, другие обеспечивают лучшую степень сжатия, среди них нет лидера: разные файлы лучше сжимаются разными программами. Среди наиболее распространенных программ можно назвать ARJ, PKZIP, LHA, PKPAK, PAK.

Из перечисленных программ-упаковщиков наиболее популярны - ARJ, PKZIP/PKUNZIP, а также встроенный в Norton Commander упаковщик NCZIP, совместимый с PKZIP. Эти программы обеспечивают высокую скорость работы и большую степень сжатия информации. Программа PKZIP/PKUNZIP стала фактическим стандартом сжатия файлов, а программа ARJ, обеспечивая почти такую же степень сжатия, отличается разнообразным сервисом и умеет создавать многотомные архивы (последовательности архивных файлов), располагающиеся на нескольких дискетах.

Имена архивных файлов программ PKZIP и ARJ имеют расширения .ZIP и .ARJ соответственно.

Программа ARJ выполняет все функции по обслуживанию своих архивов. Так она может и помещать файлы в архив, и извлекать файлы из архива, и делать "саморазворачивающиеся" файлы и т. д. А для .ZIP-файлов различные функции по обслуживанию архивов выполняются разными программами:

PKZIP - помещение файлов в архив;

PKUNZIP - извлечение файлов из архива;

PKZIPFIX - восстановление поврежденного архивного файла;

ZIP2EXE - создание "саморазворачивающихся" файлов.

Кроме того разработчиками этих программ и независимыми программистами были созданы различные вспомогательные программы для обработки архивов, например: PKZFIND - поиск файла на диске и в .ZIP-архивах; REARJ -преобразование всех архивов в .ARJ-архивы; ARJSORT - сортировка .ARJ-ар-

хивов; ARCVIEW - просмотр и диалоговая модификация архивов.

10.2. Режимы программ PKZIP/PKUNZIP и ARJ

Программы-упаковщики имеют большое количество функций, выбор нужных функций выполняется в программной строке при вызове программ. Задание функций программы ARJ осуществляется с помощью задания команды и режимов. Код команды - это одна буква, она указывается в командной строке сразу за именем программы и задает вид деятельности, который должна выполнить программа. Например, A - добавление файла в архив; T - тестирование (проверка) архива; E - извлечение файлов из архива; M - пересылка файлов в архив и т.д.

Задание функций программ PKZIP/PKUNZIP осуществляется только с помощью указания режимов. Режимы могут указываться в любом месте командной строки после имени программы, они задаются либо с предшествующим знаком "-", либо с предшествующим знаком "/".

10.3. Помещение файлов в архив

При помещении файлов в архив используются следующие форматы вызова:

PKZIP <режимы> <имя архива> (имена файлов)...

ARJ <команда> <режимы> <имя-архива> (каталог) (имена файлов)...

Для начала архивации вводится команда программы-упаковщика и выполняются запрошенные действия. На экране изображаются имена помещаемых в архив файлов и процент обработанной части файла. По окончании сжатия каждого файла напротив его имени выводится степень сжатия.

Программы PKZIP и ARJ имеют три режима помещения файлов в архив:

Add - добавление в архив всех файлов;

Update - добавление в архив новых файлов;

Freshen - добавление новых версий имеющихся в архиве файлов.

Задание этих режимов осуществляется следующим образом:

Режим	PKZIP	ARJ
Add	по умолчанию	команда A
Update	режим -U	команда U

Freshen режим -F команда F

Примеры. PKZIP myzip - добавление в архивный файл MYZIP.ZIP всех файлов из текущего каталога; ARJ a myarj - добавление в архивный файл MYARJ.ARJ всех файлов из текущего каталога; PKZIP docfiles *.doc a:*.doc добавление в архивный файл DOCFILES.ZIP всех файлов с расширением .DOC из текущего каталога и из корневого каталога на диске A; ARJ a docfiles *.doc a:*.doc - добавление в архивный файл DOCFILES.ARJ всех файлов с расширением .DOC из текущего каталога и из корневого каталога на диске A; PKZIP -f a:myarc b:*.* - добавление в архив A:MYARC.ZIP новых версий файлов этого архива из корневого каталога диска B: ARJ f a:myarc b:*.* - добавление в архив A:MYARC.ARJ новых версий файлов этого архива из корневого каталога диска B:

Пересылка файлов в архив

Очень часто требуется не копировать, а пересылать файлы в архив. Иначе говоря, те файлы, которые были успешно добавлены в архив, должны удаляться с диска. Для пересылки файлов в архив можно использовать следующие режимы программ PKZIP и ARJ:

PKZIP - режим -M (можно указывать вместе с режимами -A.-U или -F); ARJ - режим -D (можно указывать вместе с командами A,U или F) или M; Команда M программы ARJ эквивалентна команде A с режимом -D, она задает добавление файлов в архив с удалением исходных файлов.

Примеры: PKZIP -m myarc - пересылка в архивный файл MYARC.ZIP всех файлов из текущего каталога; ARJ a -d myarc - пересылка в архивный файл MYARC.ARJ всех файлов из текущего каталога; ARJ m -jtl myarc - пересылка в архивный файл MYARC.ARJ всех файлов из текущего каталога с дополнительным контролем правильности помещенных в архив копий файлов; PKZIP -m -u docfiles*.doc a:*.doc - пересылка в архивный файл DOCFILES.ZIP всех файлов с расширением .DOC из текущего каталога и из корневого каталога на диске A:, кроме тех копий, которых нет в архиве DOCFILES.ZIP:

Извлечение файлов из архива

Для извлечения .ZIP-файлов из архива используется программа PKUNZIP, а .ARJ-файлы извлекает сама программа ARJ.

Извлечение происходит по следующим форматам:

PKUNZIP <режимы> <имя архива> (имена файлов)...

ARJ <команда> <режимы> <имя архива (каталог\)> (имена файлов)

Поясним параметры команд:

<команда>- одна буква, которая задает действие, которое должна выполнить программа ARJ. Например, E - извлечение файлов из архивов, X - извлечение файлов из архива в соответствующие каталоги и т.д.;

<режим> - указывается с предшествующим знаком "-" или "/"; они задают или уточняют требуемые от программы действия;

<имя-архива>- задает имя архива, откуда извлекаются файлы. Если расширение у архивного файла не указано, подразумевается .ZIP для программы PKUNZIP и ARJ - для программы ARJ. В имени архива можно употреблять символы * и ? - в этом случае обрабатывается несколько архивных файлов;

<каталог>- задает каталог, в который помещаются извлекаемые файлы. Если каталог не указан, подразумевается текущий каталог;

<имена-файлов>- указывают, какие файлы извлекаются из архива. При задании имен файлов можно использовать символы * и ?. По умолчанию подразумеваются все файлы, имеющиеся в архиве.

После ввода команды программы архивации начинают извлечение файла из архива. На экране изображаются имена извлекаемых из архива файлов. При указании имен, извлекаемых из архива программы PKUNZIP, используют символ "/" вместо "\" в качестве разделителя имен каталогов и файлов.

Примеры:

PKUNZIP a:archive -o - извлечение всех файлов из архива A:ARCHIVE.ZIP и помещение их в текущий каталог. Файлы на диске с тем же именем затираются без предупреждений.

ARJ e a:archive -jvo d:\ - извлечение всех файлов из архива A:ARCHIVE.ARJ в корневой каталог диска D:. Файлы на диске с тем же именем затираются без предупреждений.

PKUNZIP a:archive -n -извлечение всех файлов из архива A:ARCHIVE.ZIP в текущий каталог. Если на диске имеется файл с тем же именем, что и в архиве, то файлы на диске с более ранней датой последней модификации затираются без предупреждения, а файлы с той же или более поздней датой остаются без изменений.

ARJ e -u a:archive c: - извлечение всех файлов из архива A:ARCHIVE.ARJ в текущий каталог на диске C:. Файлы на диске с такой же или с более поздней датой последней модификации не изменяются (извлечение соответствующих файлов из архива не производится), а для файлов с более ранней датой, чем у соответствующего файла в архиве, выводится запрос о том, надо ли его затирать.

Просмотр оглавления архива

С помощью функций просмотра оглавления архива можно узнать, какие файлы содержатся в архиве и получить подробную информацию об этих файлах: в каком каталоге на диске находится файл, каковы дата и время последней модификации файла, каков размер файла на диске и в архиве.

Для просмотра оглавления архива используются следующие команды: PKUNZIP -V <имя архива> (имена файлов)...

ARJ L <имя архива> (имена файлов)...

Примеры:

PKUNZIP a:\archive -v - вывод информации о файлах в архиве A:\ARCHIVE.ZIP;

ARJ L * *.txt - вывод информации о файлах с расширением TXT из всех архивных файлов типа ARJ из текущего каталога.

PKUNZIP A:\archive -VN - вывод оглавления архива A:\ARCHIVE.ZIP в отсортированном в алфавитном порядке имен файлов.

Для вывода информации о файлах в файл или на принтер в конце команды

следует написать:

```
> имя файла        - для вывода в файл:
> PRN              - для вывода на принтер.
```

Вывод оглавления архива программой PKUNZIP имеет вид:

```
   Searching ZIP:  HELP.ZIP
Length  Metod   Size  Ratio Data   Time    CRC-32  Attr Name
------  -----   ----  ----- ----   ----    ------  ---- ----
    54  Stared     54   0% 22-01-98 16:45 8aa099b4  --w- DIRIN
228252  A-Xtra 114051  51% 16-02-98 16:21 676b9463  --w- DOS.E1
346901  A-Xtra 179753  49% 22-01-98 03:43 ec660077  --w- DOS50.H1
 34881  A-Xtra  16751  52% 01-09-98 01:24 de2456a4  --w- HELP.EXE
465408  A-Xtra 212661  55% 11-10-98 20:20 456a3fa3  --w- TECH.H1
-------         ------ ---                          --------
1075496        623270 52%                                5
```

В столбцах следующая информация:CBC-32 - код циклического контроля файла;A1лЬШ:e - атрибуты файла (s-системный, п-спрятанный, w-дпя чтения и записи, r-только для чтения, *-файл защищен паролем). Из данного примера видно, что % места в архиве, сохраненного благодаря сжатию файла в среднем коло 50%.

Более подробную информацию о работе с архивными файлами, например о таких вопросах как:

- архивация файлов из подкаталогов;
- проверка целостности архивов;
- использование защиты файла с паролем;
- восстановление поврежденных архивов;
- архивация на дискеты;
- многотомные архивы;
- вывод файлов на экран и на печать;
- и другие вопросы архивации можно получить из рекомендованной литературы.

Тема 11. Антивирусные программные средства (2часа)

Лекция №17.

Вопросы: Сущность и проявление компьютерных вирусов. Жизненный цикл вируса. Объекты воздействия и деструктивные функции вирусов. Классификация компьютерных вирусов. Основные виды вирусов. Программы обнаружения и защиты от вирусов. Проблемы защиты от макровирусов. Сетевые вирусы. Основные меры по защите от вирусов.

11. Компьютерные вирусы и антивирусные программы

11.1. Что такое компьютерный вирус?

Компьютерный вирус - это специально написанная, как правило, небольшая по размерам программа, которая может записывать (внедрять) свои копии (возможно, измененные) в компьютерные программы, расположенные в исполнимых файлах, системных областях дисков, драйверах, документах и т. д., причем эти копии сохраняют возможность к "размножению". Процесс внедрения вирусом своей копии в другую программу (системную область диска и т. д.) называется заражением, а программа или иной объект, содержащий вирус - зараженным.

Когда зараженная программа начинает работу, то сначала управление получает вирус. Вирус находит и "заражает" другие программы, а также выполняет какие-нибудь вредные действия (например, портит файлы или таблицу размещения файлов на диске, "засоряет" оперативную память и т.д.). Для маскировки вируса действия по заражению других программ и нанесению вреда могут выполняться не всегда, а, скажем, при выполнении определенных условий. После того как вирус выполнит нужные ему действия, он передает управление той программе, в которой он находится, и она работает так же, как обычно. Тем самым внешне работа зараженной программы выглядит так же, как и незараженной.

Многие разновидности вирусов устроены так, что при запуске зараженной программы вирус остается резидентно, т.е. до перезагрузки DOS, в памяти компьютера и время от времени заражает программы и выполняет вредные действия на компьютере.

Все действия вируса могут выполняться достаточно быстро и без выдачи каких-либо сообщений, поэтому пользователю очень трудно заметить, что в компьютере происходит что-то необычное.

Пока на компьютере заражено относительно мало программ, наличие вируса может быть практически незаметно. Однако при прошествии некоторого времени на компьютере начинает твориться что-то странное, например:

- некоторые программы перестают работать или начинают работать неправильно;
- на экран выводятся посторонние сообщения, символы и т.д.;
- работа на компьютере существенно замедляется;
- некоторые файлы оказываются испорченными и т.д.;

К этому моменту, как правило, уже достаточно много (или даже большинство) тех программ, которыми Вы пользуетесь, являются зараженными вирусом, а некоторые файлы и диски - испорченными. Более того, зараженные программы с вашего компьютера могли быть уже перенесены с помощью дискет или по локальной сети на компьютеры ваших коллег и друзей.

Некоторые разновидности вирусов ведут себя еще более коварно. Они вначале незаметно заражают большое число программ или дисков, а потом причиняют очень серьезные повреждения, например, форматируют весь жесткий диск на компьютере. А бывают вирусы, которые стараются вести себя как можно более незаметно, но понемногу и постепенно портят данные на жестком диске компьютера.

Таким образом, если не предпринимать мер по защите от вируса, то последствия заражения компьютера могут быть очень серьезными. Например, в начале 1989г. вирусом, написанным американским студентом Моррисом, были заражены и выведены из строя тысячи компьютеров, в том числе принадлежащих министерству обороны США. Автор вируса был приговорен судом к трем месяцам тюрьмы и штрафу в 270 тыс. долл. Наказание могло быть и более строгим, но суд учел, что вирус не портил данные, а только размножался.

Для того чтобы программа-вирус была незаметной, она должна быть небольшой. Поэтому, как правило, вирусы пишутся на языке ассемблера. Некоторые авторы таких программ создали их из озорства, некоторые - из стремления "насолить" кому-либо (например, уволившей их фирме) или из ненависти ко всему роду человеческому. В любом случае созданная программа-вирус может (потенциально) распространиться на всех компьютерах, совместимых с тем, для которого она была написана, и причинить очень большие разрушения.

Следует заметить, что написание вируса - не такая уж сложная задача, вполне доступная изучающему программирование студенту. Поэтому еженедельно в мире появляются все новые и новые вирусы.

11.2. Испорченные и зараженные файлы

Компьютерный вирус может испортить, т.е. изменить ненадлежащим образом, любой файл на имеющихся в компьютере дисках. Но некоторые виды файлов вирус может "заразить". Это означает, что вирус может "внедриться" в эти файлы, т.е. изменить их так, что они будут содержать вирус, который при некоторых обстоятельствах может начать свою работу.

Следует заметить, что тексты программ и документов, информационные файлы баз данных, таблицы табличных процессоров и другие файлы не могут быть заражены вирусом, он может их только испортить.

Вирусом могут быть "заражены" следующие виды файлов:

1. Исполняемые файлы, т.е. файлы с расширениями имен .COM и .EXE, а также оверлейные файлы, загружаемые при выполнении других программ. Вирусы, заражающие файлы, называются файловыми вирусами^ Вирус в зараженных исполнимых файлах начинает свою работу при запуске той программы, в которой он находится. Наиболее опасны те файловые вирусы, которые после своего запуска остаются в памяти резидентно - они могут заражать файлы и вредить до следующей перезагрузки компьютера. А если они заразят любую программу, запускаемую из файла AUTOEXEC.BAT или CONFIG.SYS, то и при запуске с жесткого диска вирус снова начнет свою работу.

2. Загрузка операционной системы и главная загрузочная запись жесткого диска. Вирусы, поражающие эти области, называются загрузочными вирусами или бутовыми (от слова boot-загрузчик). Такой вирус начинает свою работу при начальной загрузке компьютера и становится резидентным, т.е. постоянно находится в памяти компьютера. Механизм распространения - заражение загрузочных записей, вставляемых в компьютер дискет. Часто такие вирусы состоят из двух частей, поскольку загрузочная запись и главная загрузочная запись имеют небольшой размер и в них трудно разместить целиком программу вируса.

Часть вируса, не помещающаяся в них, располагается в другом кластере в области данных диска (обычно такой кластер объявляется дефектным, чтобы программа вируса не была затерта при записи данных на диск). Дискеты, через которые распространяются загрузочные вирусы, могут быть не только системными, но и любыми другими. Для заражения компьютера загрузочным вирусом достаточно иметь всего один раз зараженную дискету в дисководе A: в момент перезагрузки компьютера. При этом вирус заразит жесткий диск компьютера. И после этого при загрузке с жесткого диска компьютера будет запускаться вирус.

3. Вирусы, меняющие файловую систему на диске, обычно называемые DIR - вирусами. Такие вирусы прячут свое тело в некоторый участок диска (обычно - в последний кластер диска) и помечают его в таблице размещения файлов (FAT) как конец файла. Для всех .COM и .EXE-файлов, содержащихся в соответствующих элементах каталога, указатели на первый участок файла заменяются ссылкой на участок диска, содержащий вирус, а правильный указатель в закодированном виде прячется в неиспользуемой части элемента каталога. Поэтому при запуске любой программы в память загружается вирус, после чего он остается в памяти резидентно, подключается к программам DOS для обработки файлов на диске и при всех обращениях к элементам каталога выдает правильные ссылки.

Таким образом, при работающем вирусе файловая система на диске кажется совершенно нормальной. При поверхностном просмотре зараженного диска на "чистом" компьютере также ничего странного не наблюдается. Разве лишь при попытке прочесть или скопировать с зараженной дискеты программные файлы из них будут прочтены или скопированы только 512 или 1024 байта, даже если файл гораздо длиннее. А при запуске любой исполнимой программы с зараженного таким вирусом диска этот диск, как по волшебству, начинает казаться исправным (неудивительно, ведь компьютер при этом становится зараженным).

При анализе на "чистом" компьютере с помощью программ ChkDsk или NDD файловая система зараженного DIR-вирусом диска кажется совершенно испорченной. Так, программа ChkDsk выдает кучу сообщений о пересечениях файлов и о цепочках потерянных кластеров. Не следует исправлять эти ошибки программами ChkDsk или NDD- при этом диск окажется безнадежно испорченным. Для исправления зараженных этими вирусами дисков надо использовать только специальные антивирусные программы (например, последние версии Aidstest).

4. Драйверы устройств, т.е. файлы, указываемые в приложении Devise файла CONFIG.SYS. Вирус, находящийся в них, начинает свою работу при каждом обращении к соответствующему устройству. Вирусы, заражающие драйверы устройств, очень мало распространены, поскольку драйверы редко переписывают с одного компьютера на другой.

5. Системные файлы DOS (MSDOS.SYS и IO.SYS) - их заражение возможно так называемыми вирусами семейства ЗАРАЗА. Так как первый из них выводил сообщение: В BOOT СЕКТОРЕ - ЗАРАЗА!. Вирус этого семейства делает следующее:

- копирует содержимое файла IO.SYS в конец логического диска; сдвигает элементы корневого каталога, начиная с третьего, на один элемент к концу каталога;
- копирует первый элемент корневого каталога (соответствующий файлу IO.SYS) в освободившийся третий элемент корневого каталога и устанавливает в нем номер начального кластера, указывающий на место, куда было скопировано содержимое файла IO.SYS;
- записывает свое тело в место, где находится файл IO.SYS (как правило, в начале области данных логического диска): у первого элемента корневого каталога диска устанавливает признак "метка тома".

В результате в корневом каталоге появятся два системных файла IO.SYS. При этом система перестанет загружаться с жесткого диска, так как вирус в своем теле хранит адрес начального сектора исходного файла IO.SYS.

5. Командные файлы - заражаются достаточно редко. Обычно эти вирусы формируют с помощью команд командного файла (команд ECHO и др.) исполняемый файл на диске (как правило, в формате .COM), запускают этот файл, он выполняет размножение вируса и вредящие действия, после чего данный файл стирается. Вирус в зараженных командных файлах начинает свою работу при выполнении командного файла, в котором он находится. Иногда вызов зараженного командного файла вставляется в файл AUTOEXEC.BAT.

6. Документы Word для Windows версий 6.0 и 7.0. Так как сейчас редакторы Word для Windows более всего распространены, то в 1995 г. появилась новая разновидность вируса, заражающая файлы документов, созданные этими редакторами - макровирусы. Это стало возможным, поскольку в Word для Windows встроен мощный язык макрокоманд WordBasic. При этом макрокоманды не видны в редактируемом документе - для их просмотра и редактирования надо выбрать в группе меню Tools (Сервис) пункт Macro (Макрос), а много ли пользователей вообще что-то слышали об этом пункте меню... Возможности WordBasic позволяют писать на нем вирусы. Запуск вируса происходит при открытии на редактирование зараженных документов. При этом макрокоманды вируса записываются в глобальный шаблон NORMAL.DOT, так что при новых сеансах работы с Word для Windows вирус будет автоматически активирован. При наличии вируса при сохранении редактируемых документов или записи документов или записи документов на диск под новым именем (командой Save As) вирус копирует свои макрокоманды в записываемый на диск документ, так что тот оказывается зараженным.

В принципе, возможно заражение и других объектов, содержащих программы в какой либо форме - текстов программ, электронных таблиц и т.д. Электронные таблицы содержат макрокоманды, в том числе и макрокоманды, автоматически выполняющиеся при открытии таблицы. Поэтому они могут содержать вирусы. Подобные вирусы обнаружены в табличном процессоре Excel.

Как правило, каждая конкретная разновидность вируса может заражать только один или два типа файлов. Чаще всего встречаются вирусы, заражающие ис-

полнимые файлы. Некоторые вирусы заражают только .COM-файлы, некоторые - только .EXE-файлы, а большинство - и те и другие. На втором месте по распространенности загрузочные вирусы. Некоторые вирусы заражают и файлы, и загрузочные области дисков. Вирусы, заражающие драйверы устройств, встречаются крайне редко, обычно такие вирусы умеют заражать и исполнимые файлы.

Что вирус не может заразить? Вирус является программой, поэтому объекты, не содержащие программ и не подлежащие преобразованию в программы, не могут быть заражены вирусом. Например, графические файлы форматов .BMP, .PCX, GIF, WMF и др. содержат только описания рисунков, поэтому как бы их вирус не изменял, при просмотре или другом использовании графического файла можно получить искаженный рисунок или сообщение о неправильном формате файла, но вирус при этом не может быть запущен. Таким образом, не содержащие программы объекты вирус может только испортить, но не заразить. К числу таких объектов относятся текстовые файлы (кроме командных файлов и текстов программ), документы простых текстовых редакторов документов типа Л ЕКСИКОНа или Multi-Edit, информационные файлы баз данных и т. д.

Чтобы предотвратить свое обнаружение, некоторые вирусы применяют довольно хитрые приемы маскировки. Это "невидимые" и самомодифицирующиеся вирусы.

"Невидимые" вирусы - это такие резидентные вирусы (файловые и загрузочные), которые предотвращают свое обнаружение тем, что перехватывают обращения DOS (и тем самым прикладных программ) к зараженным файлам и областям диска и выдают их в исходном (незараженном) виде. Разумеется, этот эффект наблюдается только на зараженном компьютере - на "чистом" компьютере изменения в файлах и загрузочных областях диска можно легко обнаружить.

Некоторые антивирусные программы (например, AVSP фирмы "Диалог-МГУ") используют для борьбы с вирусами свойство "невидимых" файловых вирусов "вылечивать" зараженные файлы.

Самомодифицирующиеся вирусы - это вирусы, использующие оригинальный способ - модификацию своего тела для того, чтобы укрыться от обнаружения. Многие вирусы хранят большую часть своего тела в закодированном виде, чтобы с помощью дизассемблеров нельзя было разобраться в механизме их работы. Самомодифицирующиеся вирусы используют этот прием и часто меняют параметры этой кодировки, а кроме того, изменяют и свою стартовую часть, которая служит для раскодировки остальных команд вируса.

11.3. Основные методы защиты от компьютерных вирусов

Для защиты от вирусов можно использовать:

- общие средства защиты информации, которые полезны также и как страховка от физической порчи дисков, неправильно работающих программ или ошибочных действий пользователей;
- профилактические меры, позволяющие уменьшить вероятность заражения вирусом;
- специализированные программы для защиты от вирусов;

Общие средства защиты информации полезны не только для защиты от вируса. Имеются две основные разновидности этих средств.

копирование информации - создание копий файлов и системных областей дисков;

разграничение доступа предотвращает несанкционированное использование информации, в частности, защиту от изменений программ и данных вирусами, неправильно работающими программами и ошибочными действиями пользователей.

Несмотря на то, что общие средства защиты информации очень важны для защиты от вирусов, все же их одних недостаточно. Необходимо и применение специализированных программ для защиты от вирусов. Эти программы можно разделить на несколько видов: детекторы, доктора (фаги), ревизоры (програм-

мы контроля изменений в файлах и системных областях дисков), доктора-ревизоры, фильтры (резидентные программы для защиты от вирусов) и вакцины (иммунизаторы). Сейчас мы приведем только краткие определения этих понятий, а затем рассмотрим их более подробно.

Программы-детекторы позволяют обнаруживать файлы, зараженные одним из нескольких известных вирусов.

Программы-доктора. или фаги, "лечат" зараженные программы или диски, "выкусывая" из зараженных программ тело вируса, т.е. восстанавливая программу в том состоянии, в котором она находилась до заражения вирусом.

Программы-ревизоры сначала запоминают сведения о состоянии программ и системных областей дисков, а затем сравнивают их состояние с исходным. При выявлении несоответствий об этом сообщается пользователю.

Доктора-ревизоры - это гибриды ревизоров и докторов, т.е. программы, которые не только обнаруживают изменения в файлах и системных областях дисков, но и могут в случае изменений автоматически вернуть их в исходное состояние.

Программы-фильтры располагаются резидентно в оперативной памяти компьютера и перехватывают те обращения к операционной системе, которые используются вирусами для размножения и нанесения вреда, и сообщают о них пользователю. Пользователь может разрешить или запретить выполнение соответствующей операции.

Программы-вакцины, или иммунизаторы, модифицируют программы и диски таким образом, что это не отражается на работе программ, но тот вирус, от которого производится вакцинация, считает эти программы или диски уже зараженными. Эти программы крайне неэффективны и далее не рассматриваются.

11.4. Стратегия защиты от вирусов

Ни один тип антивирусных программ по отдельности не дает, к сожалению, полной защиты от вирусов. Поэтому никакие простые советы типа "вставьте команду запуска Aidstest в AUTOEXEC.BAT" не будут достаточными.

Наилучшей стратегией защиты от вирусов является многоуровневая, "эшелонированная" оборона. Опишем структуру этой обороны.

Средствам разведки в "обороне" от вирусов соответствуют программы-детекторы, позволяющие проверять вновь полученное программное обеспечение на наличие вирусов.

На переднем крае обороны находятся программы-сторожа (или фильтры) - это резидентные программы для защиты от вируса. Эти программы могут первыми сообщить о работе вируса и предотвратить заражение программ и дисков.

Второй эшелон обороны составляют программы-ревизоры, программы доктора и доктора-ревизоры. Ревизоры обнаруживают нападение вируса даже тогда, когда он сумел "просочиться" через передний край обороны. Программы-доктора применяются для восстановления зараженных программ, если ее копий нет в архиве, но они не всегда лечат правильно. Доктора-ревизоры обнаруживают нападение вируса и лечат зараженные программы, причем контролируют правильность лечения.

Самый глубокий эшелон обороны - это средства разграничения доступа. Они не позволяют вирусам и неверно работающим программам, даже если они проникли в компьютер, испортить важные данные.

И наконец, в "стратегическом резерве" находятся архивные копии информации и "эталонные" дискеты с программными продуктами. Они позволяют восстановить информацию при ее повреждении.

Программы-фильтры

Одной из причин, из-за которых стало возможным такое явление, как компьютерный вирус, является отсутствие в операционной системе MS DOS эффективных средств для защиты информации от несанкционированного доступа. Из-за отсутствия средств защиты компьютерные вирусы могут незаметно и безнаказанно изменять программы, портить таблицы размещения файлов и т.д.

В связи с этим различными фирмами и программистами разработаны программы-фильтры, или резидентные программы для защиты от вируса, которые в определенной степени восполняют указанный недостаток DOS. Эти програм-

мы располагаются резидентно в оперативной памяти компьютера и "перехватывают" те обращения к операционной системе, которые используются вирусами для размножения и нанесения вреда. Такими "подозрительными" действиями являются, в частности, изменение .COM и .EXE-файлов, снятие с файла атрибута "только для чтения", прямая запись на диск (запись по абсолютному адресу), форматирование диска, установка "резидентной" (постоянно находящейся в оперативной памяти) программы.

При каждом запросе на "подозрительное" действие на экран компьютера выводится сообщение о том, какое действие затребовано и какая программа желает его выполнить. Можно либо разрешить выполнение этого действия, либо запретить его. Если указанное в сообщении действие не нужно для выполнения данной программы (например, никакая программа не должна изменять командный процессор COMMAND.COM), то это действие следует запретить, так как оно скорее всего вызвано вирусом.

Некоторые программы-фильтры не "ловят" подозрительные действия, а проверяют вызываемые на выполнение программы на наличие вирусов. Это, понятно, вызывает замедление работы компьютера.

Степень защиты, обеспечиваемую программами-фильтрами, не следует переоценивать, поскольку многие вирусы для своего размножения и нанесения вреда обращаются непосредственно к программам операционной сис темы (BIOS), не используя стандартный способ вызова этих программ через прерывания, а резидентные программы для защиты от вируса перехватывают только эти прерывания. Кроме того, программы-фильтры не помогают от заражения винчестера вирусами, которые распространяются через загрузочный сектор, поскольку такое заражение происходит при загрузке DOS, т.е. до запуска любых программ или установки драйверов.

Однако преимущества использования программ-фильтров весьма значительны - они позволяют обнаружить многие вирусы на самой ранней стадии, когда вирус еще не успел размножиться и что-либо испортить. Тем самым можно свести убытки от вируса к минимуму.

Программы-детекторы

Программы-детекторы позволяют обнаруживать файлы, зараженные одним из нескольких известных вирусов. Некоторые программы-детекторы также выполняют эвристический анализ файлов и системных областей дисков, что часто (но отнюдь не всегда) позволяет обнаруживать новые, не известные программе-детектору, вирусы. Многие программы-детекторы позволяют также "лечить" зараженные файлы или диски, удаляя из них вирусы (лечение поддерживается только для вирусов, известных программе-детектору).

Программы-ревизоры

Программы-ревизоры имеют две стадии работы. Сначала они запоминают сведения о состоянии программ и системных областей дисков (загрузочного сектора и сектора с таблицей разбиения жесткого диска). Предполагается, что в этот момент программы и системные области дисков не заражены. После этого с помощью программы-ревизора можно в любой момент сравнить состояние программ и системных областей дисков с исходным. О выявленных несоответствиях сообщается пользователю.

Многие пользователи включают команду запуска программы-ревизора в командный файл AUTOEXEC.BAT, чтобы проверка состояния программ и дисков проходила при каждой загрузке операционной системы. Это позволяет обнаружить заражение компьютерным вирусом, когда он еще не успел нанести большого вреда. Более того, та же программа-ревизор сможет найти поврежденные вирусом файлы.

Анализ изменений. Многие программы-ревизоры являются довольно "интеллектуальными" - они могут отличать изменения в файлах, вызванные, например, переходом к новой версии программы, от изменений, вносимых вирусом, и не поднимают ложной тревоги, мешая Вам работать. Дело в том, что вирусы обычно изменяют файлы весьма специфическим образом и производят одинаковые изменения в разных программных файлах. Понятно, что в нормальной ситуации такие изменения практически никогда не встречаются, поэтому про-

грамма-ревизор, зафиксировав факт таких изменений, может с уверенностью сообщить, что они вызваны именно вирусом.

Невидимые вирусы. Следует заметить, что многие программы-ревизоры не умеют обнаруживать заражение "невидимыми" вирусами, если такой вирус активен в памяти компьютера. Но некоторые программы-ревизоры, например ADinf фирмы "Диалог-Наука", все же умеют делать это, не используя вызовы DOS для чтения диска (правда, они работают не на всех дисководах). Другие программы часто используют различные полумеры: пытаются обнаружить вирус в оперативной памяти, требуют вызова из файла AUTOEXEC.BAT, надеясь работать на "чистом" компьютере, и т.д. Увы, против некоторых "хитрых" вирусов все это бесполезно. Если Вы имеете такую программу-ревизор, запускайте хотя бы время от времени ее "чистую" копию с защищенной от записи дискеты после перезагрузки DOS с такой дискеты.

Режимы проверки. Для проверки того, не изменился ли файл, некоторые программы-ревизоры проверяют длину файла. Но эта проверка недоста точна - некоторые вирусы не изменяют длину зараженных файлов. Более надежная проверка - прочесть весь файл и вычислить его контрольную сумму. Изменить файл так, чтобы его контрольная сумма осталась прежней, практически невозможно. Но полностью читать все проверяемые файлы на диске весьма долго.

Чтобы обеспечить и достаточную надежность проверки, и приемлемое время ее проведения, многие программы-ревизоры имеют режим, в котором они проверяют неизменность только критически важных участков программных файлов, которые чаще всего и меняются вирусом: заголовка EXE-файла, первых выполняемых команд файла и т.д. Это позволяет проводить ежедневную проверку наличия изменений в файлах. А для особо строгой проверки такие программы-ревизоры обычно имеют и режим полного чтения файла.

Доктора-ревизоры. В последнее время появились очень полезные гибриды ревизоров и докторов- программы, которые не только обнаруживают изменения в файлах и системных областях дисков, но и могут в случае изменений автоматически вернуть их в исходное состояние.Такие программы могут быть гораздо

более универсальными, чем программы-доктора, поскольку при лечении они используют заранее сохраненную информацию о состоянии файлов и областей диска. Это позволяет им вылечивать файлы даже от тех вирусов, которые не были созданы на момент написания программы.

Конечно, доктора-ревизоры - это не панацея. Они могут лечить не от всех вирусов, а только от тех, которые используют "стандартные", известные на момент написания программы, механизмы заражения файлов. Кроме того, никто не может излечить программы при заражении вирусами" грубиянами" типа AIDS, которые записывают себя в середину программ, не заботясь о том, будет ли после этого работать программа или нет. Но все же защита от 90-95% вирусов - это совсем неплохо. В качестве примера докторов-ревизоров можно привести ADinf+ADinfExt фирмы "Диалог-Наука" и комплексную антивирусную систему AVSP фирмы "Диалог-МГУ".

Профилактика против заражения вирусом

Известно, что легче предупредить болезнь, чем лечить ее. Если квалифицированно и своевременно будут применяться меры "компьютерной гигиены", то на компьютере вряд ли заведутся вирусы. Во-первых, необходимо использовать общие меры, обеспечивающие сохранность данных: регулярное создание резервных копий, использование методов ограничения доступа к данным и т.д. Во-вторых, выполнить следующие мероприятия.

1. Все поступающие извне данные должны подвергаться проверке. С этой целью все принесенные дискеты или полученные извне (по электронной почте, по сетям) файлы перед использованием следует проверить на наличие вируса с помощью программ-детекторов. Не используйте и не запускайте принесенные извне программы, если их назначение Вам непонятно. Если полученные файлы содержатся в архивах, следует извлечь их из архива и проверить программами-детекторами сразу после этого. Если файлы из архивов можно извлечь только программой установки пакета программ, то надо выполнить установку этого пакета и сразу после этого проверить записанные на диск файлы. Установка пакета должна проводиться при включенной резидентной программе-стороже для

защиты от вирусов. Не следует переписывать программное обеспечение с других компьютеров, так как оно может быть заражено вирусом.

2. Необходимо проводить ежедневную проверку дисков на наличие вирусов. Один способ - вставить в командный файл AUTOEXEC.BAT, выполняемый при начальной загрузке DOS, вызов программы или командного файла для проверки на наличие вирусов. При использовании антивирусного комплекта DSAV "Диалог-Наука" можно из файла AUTOEXEC.BAT, вызвать командный файл, запускающий программу-ревизора ADinf, который составляет список измененных файлов, используемый затем программами-детекторами Aidstest и Dr.Web (это позволяет существенно сократить время проверки). Пример такого командного файла включен в комплект поставки антивирусного комплекта DSAV.

При работе в среде Windows, Windows-95 и т.д. для ежедневного выполнения проверки на наличие вирусов можно использовать программы-планировщики типа Scheduler из Norton Desktop for Windows, System Agent из Microsoft Plus! (пакета дополнений для Windows 95), Norton Commander Scheduler из Norton Commander для Windows 95 и т.д.

3. Особо следует сказать о защите от вирусов документов Word для Windows. Вирусы, заражающие документы Word для Windows, запускаются благодаря тому, что в них имеется макрокоманда AutoOpen, автоматически запускающаяся при открытии документа. Однако запуск этой макрокоманды (как и других AutoExec, AutoNew, AutoClose и AutoExit, выполняющихся при запуске Word, создании нового документа, закрытий документа и выходе из Word) блокируется при нажатии клавиши Shift. Так что Вы можете нажимать Shift при открытии полученных со стороны документов, и никакой вирус, заражающий документы Word для Windows, Вам будет не страшен.

4. Обновление версий антивирусных программ необходимо делать ежемесячно, так как новые вирусы появляются еженедельно. Для некоторых программ (Norton AntiVirus) для обновления не надо приобретать новую версию программы, а следует лишь переписать с помощью модема новую версию базы

данных со сведениями о вирусах. Обычно это можно делать бесплатно. Фирма "Диалог-Наука" позволяет приобрести годовой абонемент, позволяющий получать самые последние версии программ AidsTest, Doctor Web, ADinf и ADinfExt либо по почтовой рассылке, либо по электронной почте. Обновление версий происходит не реже одного раза в месяц.

Действия при заражении вирусом

Вы можете подозревать наличие вируса, если:

антивирусная программа сообщает об обнаружения неизвестного вируса;

на экран или принтер начинают выводиться посторонние сообщения, символы и т. д.;

некоторые файлы оказываются испорченными;

некоторые программы перестают работать или начинают работать неправильно;

работа на компьютере существенно замедляется.

При заражении компьютера вирусом (или при подозрении на это) важно соблюдать пять правил.

1. Прежде всего не надо торопиться и принимать опрометчивые решения. Непродуманные действия могут привести к потере части данных, которые можно было бы восстановить, а также к повторному заражению компьютера.

2. Немедленно выключите компьютер, чтобы вирус не продолжал своих разрушительных действий.

3. Все действия по обнаружению вида заражения и лечению компьютера следует выполнять только при правильной загрузке компьютера с защищенной от записи "эталонной" дискеты с операционной системой. При этом следует использовать только программы (исполняемые файлы), хранящиеся на защищенных от записи дискетах.

4. Лечение от вируса обычно несложно, но иногда (при существенных разрушениях, причиненных вирусом) оно бывает затруднительным. Если у пользователя нет должного опыта, то целесообразно прибегнуть к помощи коллег.

5. Лечение компьютера от вируса - процесс творческий, поэтому любые рекомендации по этому поводу не надо воспринимать как догму. Тем более, что вирусы очень часто изменяются и поэтому методы борьбы с ними также должны меняться.

Использование резидентной программы-сторожа позволяет обнаружить вирус на самом раннем этапе, когда он не успел еще активизироваться, заразить другие программы или диски и испортить какие-либо данные.

Когда вирус активизировался, а значит уже заразил или испортил данные на дисках компьютера, то надо перегрузить компьютер, начать выявление вируса и затем лечение.

Правильно компьютер должен перезагружаться так:

1. Приготовьте безвирусную, защищенную от записи системную дискету. Вставьте дискету в дисковод компьютера.
2. Нажмите на кнопку перезагрузки - RESET или выключите и снова включите компьютер.
3. Сразу после начала загрузки в ответ на приглашение нажмите клавиши входа в программу конфигурирования компьютера - SETUP.
4. Проверьте правильность установки, порядка загрузки. Если загрузка происходит сначала с жесткого диска, а затем с дискеты, то эту установку надо выключить.
5. Выйдите из программы конфигурирования и сохраните конфигурацию.
6. Загрузите компьютер с системной дискеты и при этом вирус не будет запущен.

Лечение начинается программами-детекторами, которыми поочередно проверяются все логические диски. При обнаружении вируса следует по справочнику выяснить возможные последствия заражения и меры по их устранению. Если вирус безобидный, то кроме его удаления ничего делать не надо. Если вирус изменил некоторые участки жесткого диска, то вероятнее всего, придется заново инсталировать на диск поврежденные файлы.

Если программа-детектор не обнаружила вирусы, то следует запустить программу-ревизор, которая после обнаружения вируса предлагает его исправить или отказаться от исправления. В большинстве случаев лечение происходит успешно. Иногда теряется часть данных. Реже поврежденный файл следует удалить и затем восстановить его из других источников.

После окончания лечения диска следует проверить целостность файловой системы и поверхности диска программой Norton Disk Doctor. Если повреждения файловой системы значительны, то целесообразно скопировать с диска на дискеты все нужные файлы, резервных копий которых не имеется, заново отформатировать диск, а затем заново установить все пакеты программ с дистрибутивов, а собственные данные - с резервных копий.

Для исключения повторного заражения вирусом с дискет, все дискеты должны быть проверены антивирусными программами.

Тема 12. Права доступа в сети (2час)

Лекция №18

Вопросы: Основные положения. Мое сетевое окружение. Как найти в сети другие компьютеры. Как найти файлы и папки на других компьютерах. Как получить доступ. Как предоставить другим пользователям доступ к вашим файлам. Выдача разрешений на доступ к сети.

Какими бывают сети

Сеть — группа компьютеров, соединенных друге другом с помощью специального оборудования, обеспечивающего обмен информацией между ними. Соединение между двумя компьютерами может быть непосредственным (двухточечное соединение) или с использованием дополнительных узлов связи.

В дальнейшем компьютер, который подключен к сети, называется рабочей станцией (Workstation). Как правило, с этим компьютером работает человек. В сети присутствуют н такие компьютеры, на которых никто не работает. Они используются в качестве управляющих центров в сети и как накопители информации. Такие компьютеры называют серверами.

Если компьютеры расположены сравнительно недалеко друг от друга и соединены с помощью высокоскоростных сетевых адаптеров (скорость передачи данных — 10-100 Мбит/с), то такие сети называются локальными. При использовании локальной сети компьютеры, как правило, расположены в пределах одной комнаты,, здания пли в нескольких близко расположенных домах.

Для объединения компьютеров или целых локальных сетей, которые расположены ачител ы IDM расстоянии друг от друга, используются модемы, а также выделенные или спутниковые каналы связи. Такие сети носят название глобальных. Обычно скорость передачи данных в таких сетях значительно ниже, чем в локальных.

Как устроена сеть

Существуют два вида архитектуры сети: одноранговая (Peer-to-peer) и клиент/сервер (Client/Server), На данный момент архитектура клиент/сервер практически вытє-снила одноранговую.

Если используется одноранговая сеть, то все компьютеры, входящие в нее, имеют одинаковые права. Соответственно, любой компьютер может выступать в роли сервера, предоставляющего доступ к своим ресурсны, или клиента, использующего ресурсы других серверов,

В сети, построенной на архитектуре клиент/сервер, существует несколько основных компьютеров — серверов. Остальные компьютеры, которые входят в сеть, носят название клиентов, или рабочих станций.

Сервер — это компьютер, который обслуживает другие компьютеры в сети. Существуют разнообразные виды серверов, отличающиеся друг от друга услугами, ко торые они предоставляют: серверы баз данных, файловые серверы, принт-серверы, почтовые серверы, веб-серверы и т, д.

Одноранговая архитектура получила распространение в небольших офисах или в домашних локальных сетях. В большинстве случаев, чтобы создать такую сеть, вам понадобится пара компьютеров, которые снабжены сетевыми картами, и кабель. В качестве кабеля используют витую пару четвертой или пятой категории. Витая пара получила такое название потому, что пары проводов

внутри кабеля перекручены (это позволяет избежать помех и внешнего влияния). Все еще можно встретить достаточно старые сети, которые используют коаксиальный кабель. Такие сети морально устарели, а скорость передачи информации в них не превышает 10 Мбит/с.

После того как сеть будет создана, а компьютеры соединены между собой, нужно настроить все необходимые параметры программно. Прежде всего убедитесь, что на соединяемых компьютерах были установлены операционные системы с поддержкой работы в сети (Linux, FreeBSD, Windows NT, Windows XP) или системы с поддержкой сетевых функций (Windows 95, Windows for Workgroups).

Все компьютеры в одноранговой сети объединяются в рабочие группы, которые имеют свои имена (идентификаторы),

В случае использования архитектуры сети клиент/сервер управление доступом осуществляется на уровне пользователей. У администратора появляется возможность разрешить доступ к ресурсу только некоторым пользователям. Предположим, что вы делаете свой принтер доступным для пользователей сети. Если вы не хотите, чтобы кто угодно печатал на вашем принтере, то следует установить пароль для работы с этим ресурсом. При одноранговой сети любой пользователь, который узнает ваш пароль, сможет получить доступ к вашему принтеру. В сети клиент/ сервер вы можете ограничить использование принтера для некоторых пользователей вне зависимости от того, знают они пароль или нет.

Чтобы получить доступ к ресурсу в локальной сети, построенной на архитектуре клиент/сервер, пользователь обязан ввести имя пользователя (Login — логин) и пароль (Password). Следует отметить, что имя пользователя является открытой информацией (например, вам обязательно" нужно знать имя пользователя, чтобы отправить ему электронное письмо), а пароль — конфиденциальной.

Процесс проверки имени пользователя называется идентификацией. Процесс проверки соответствия введенного пароля имени пользователя — аутентифика-

цией. Вместе идентификация и аутентификация составляют процесс авторизации. Часто термин «аутентификация» используется в широком смысле: для обозначения проверки подлинности.

Из всего сказанного можно сделать вывод о том, что единственное преимущество одноранговой архитектуры — это ее простота и невысокая стоимость. Сети клиент/сервер обеспечивают более высокий уровень быстродействия и защиты.

Архитектура клиент/сервер предусматривает использование одного или нескольких серверов. В зависимости от предоставляемых услуг существуют различные виды серверов: серверы печати, баз данных, почтовые, веб-серверы и т. п.

Достаточно часто один и тот же сервер может выполнять функции нескольких серверов, например файлового н веб-сервера. Естественно, общее количество функций, которые будет выполнять сервер, зависит от нагрузки и его возможностей. Чем выше мощность сервера, тем больше клиентов он сможет обслужить и тем большее количество услуг предоставить. Поэтому в качестве сервера практически всегда назначают мощный компьютер с большим объемом памяти н быстрым процессором (как правило, для решения серьезных задач используются многопроцессорные системы).

Общий доступ к файлам

Теперь пора перейти к наиболее распространенному заданию, которое призвана выполнять локальная сеть, — совместному доступу к файлам и папкам. Данная функция даст возможность создать в пределах организации библиотеку документов, шаблонов и т. д.

Прежде всего вам нужно выбрать свойства папки, ресурсы которой вы хотите сделать открытыми Д.1Я пользователей сети, и перейти на вкладку, отвечающую за доступ. Внешним вид окна показан на рис, 2.25.

Здесь можно настроить следующие параметры.

Q Открыть общий доступ н этой папке — дает право пользователям сети копировать файлы, однако изменить или удалить ваши файлы (а также записать свои) пользователи не смогут.

а Общий ресурс — устанавливает для папки сетевое имя, под которым она будет отображаться в сети. Обратите внимание на то, что сетевое имя не обязательно должно соответствовать реальному названию папки.

Разрешить изменение файлов по сети — дает пользователям право копировать в эту папку свои файлы и наменять или удалять ваши. Не следует открывать полный доступ к системным папкам (Windows, Program Files) и папкам, которые содержат важные данные, так как кто-то из пользователей может случайно их удалить. Идеальный вариант — создать папку, специально предназначенную для передаваемых файлов, и открыть к ней полный доступ.

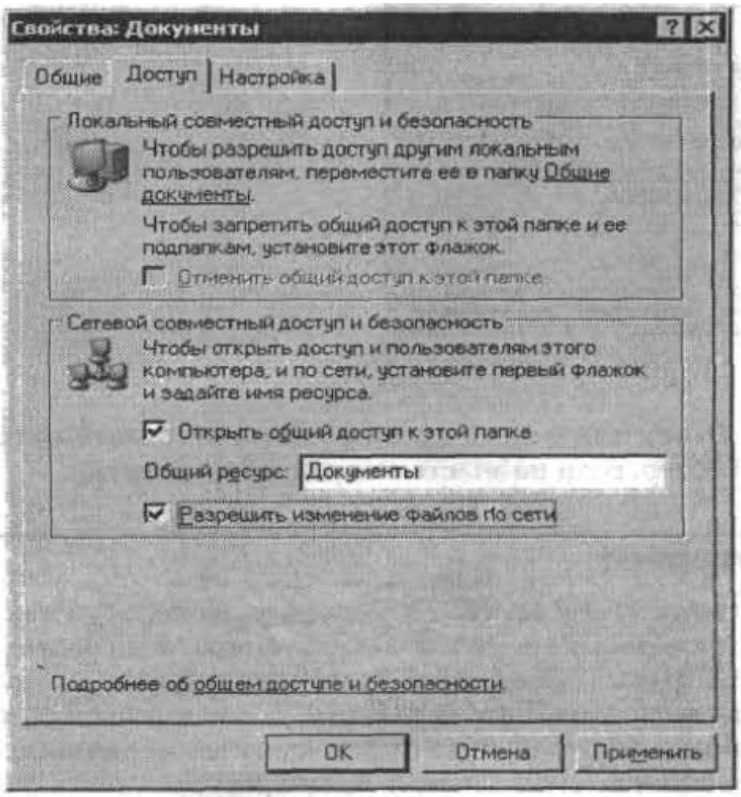


Рис. 2.25. Открываем общий доступ к файлам и папкам

В Windows XP используются два режима совместного доступа к папкам и файлам: простой и расширенный, который позволяет работать с паролями и некоторыми дополнительными функциями. Обычно для нормальной работы в малой сети полностью достаточно режима простого общего доступа к файлам и

папкам, однако для более высокого уровня и контроля над доступом пользователей к информации следует включить расширенный общий доступ к файлам. Чтобы сделать это, вам следует в любом окне выбрать Сервис > Свойства папки, перейти на вкладку Вид и снять флажок Использовать простой общий доступ к файлам.

Добавление сетевых дисков

Чтобы ускорить и упростить доступ к сетевым дискам, используемым наиболее часто, можно добавить их в Мой компьютер н обращаться к ним, как к обычным жестким дискам вашего компьютера. Чтобы сделать это, вам понадобится щелкнуть на значке сетевого окружения правой кнопкой мыши и в контекстном меню выбрать пункт Подключить сетевой диск (рис. 2.26),

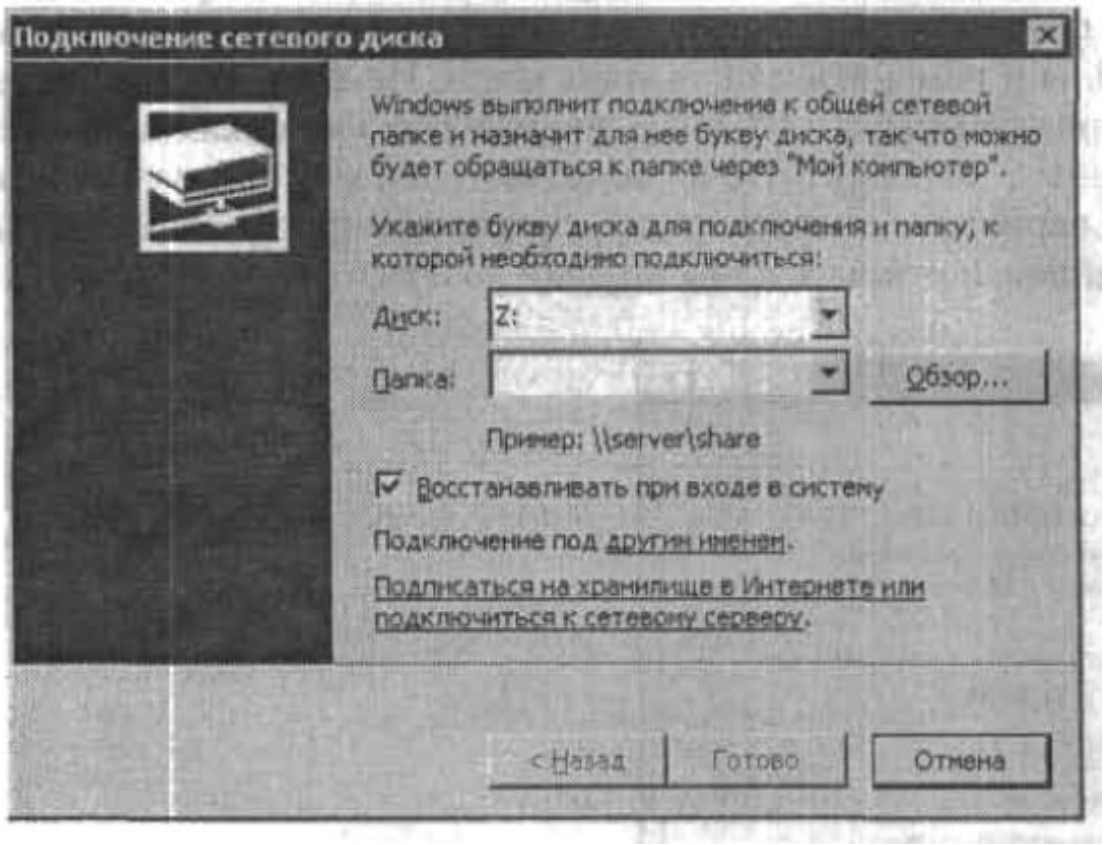


Рис. 2.26. Подключаем сетевой диск

В открывшемся окне назначьте букву для нового диска и укажите точный путь к нему (воспользуйтесь кнопкой Обзор, если не знаете точный сетевой путь).

Доступ в Интернет

Теперь займемся организацией совместного доступа в Интернет. Используя наш пример, касающийся конфигурирования клиентского компьютера, мы создаем запрос к серверу с IP-адресом 192,168.0.1, Этот адрес будет использоваться как адрес шлюза {машины в сети, через которую все остальные компьютеры будут выходить в Интернет), Данный адрес следует указать и в качестве первичного DNS (DNS — этосервис, позволяющий посимвольному имени ресурса узнать его реальный IP-адрес), После того как необходимые поля заполнены, нажмите кнопку ОК и вернитесь в меню свойств сетевого соединения. Далее перейдите на вкладку Дополнительно, Для клиентских компьютеров флажок Защитить мое подключение к Интернету должен быть снят.

Когда компьютеры сети будут настроены, вам будет необходимо установить доступ в Интернет на сервере. После того как сервер будет успешно подключен, вам нужно зайти в папку Сетевые подключения и открыть окно редактирования свойств того подключения, которое вы хотите сделать общим, В окне Свойства нужно выбрать вкладку Дополнительно и установить флажок Разрешить другим пользователям сети использовать подключение к Интернету данного компьютера (рис. 2.27).

Сразу после этого вы сможете настроить следующие параметры.

J Устанавливать вызов по требованию. Данный параметр касается модемных соединений. Если соответствующий флажок установлен, то пользователи сети получат возможность удаленно использовать модем для соединения с провайдером.

Разрешить другим пользователям сети управление общим доступом к подключению к Интернету. Если данный флажок установлен, другие пользователи смогут корректировать свои права доступа к удаленным соединениям (что нежелательно).

LJ Параметры, Данный параметр содержит список протоколов, доступных удаленным пользователям при использовании данного соединения (например, исключительно отправка и получение электронной почты).

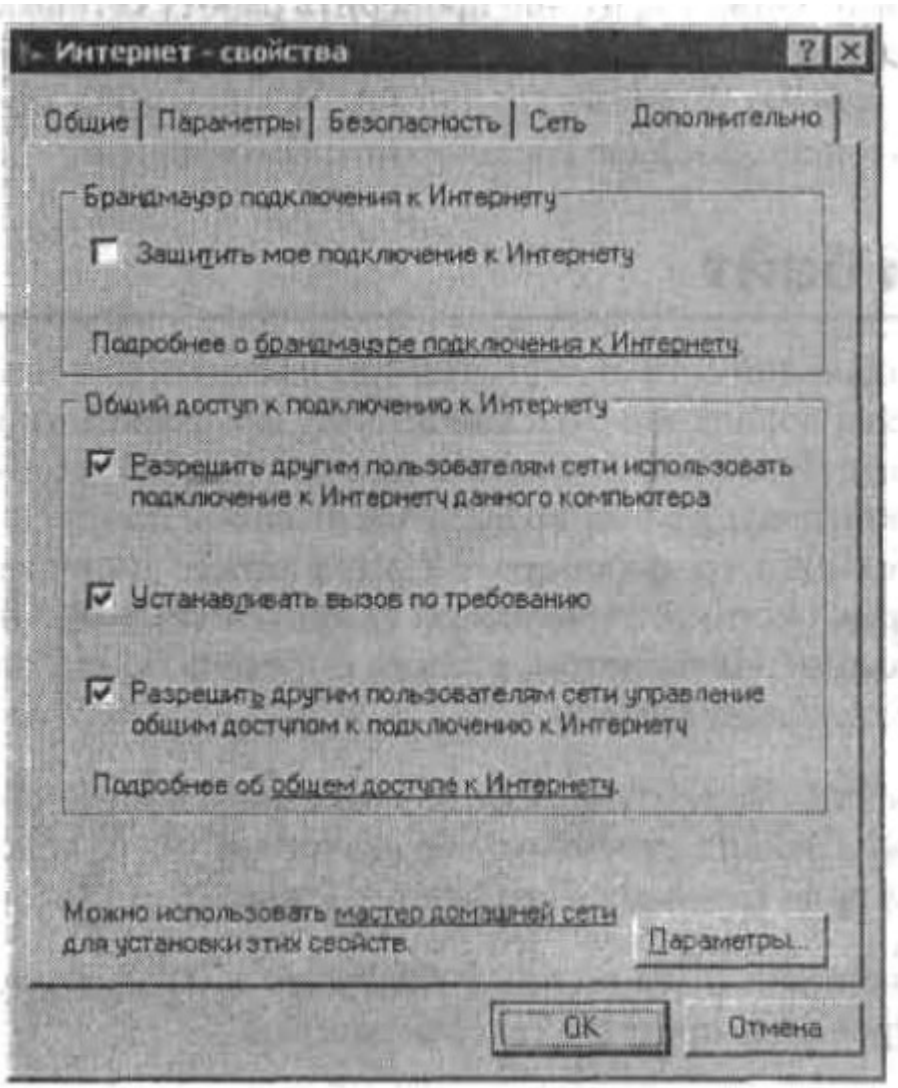


Рис. 2.27, Открываем общий доступ к Интернету

Обратите внимание на флажок Защитить мое подключение к Интернету, Его установка разрешит использование брандмауэра Windows, представляющего собой систему защиты в виде своеобразного барьера между внутренней и внешней сетями. Безусловно, мощность брандмауэра не может сравниваться со специальными аппаратными межсетевыми барьерами, однако м он способен защитить ваши данные от несанкционированного доступа.

Нажмите кнопку ОК для сохранения всех сделанных изменений. После этого можно начинать проверку работы сети, войти в Интернет с сервера и клиентских компьютеров.

6. Методические указания для проведения и выполнения лабораторных занятий.

Тема 1. Создание презентации.

Создание презентаций

Цель: Приобрести навыки создания презентации. Освоить основные приемы создания презентации, научиться использовать данные из других источников, создавать анимационные слайды и создавать гиперссылки.

Процесс создания презентации в Microsoft PowerPoint состоит из таких действий, как выбор общего оформления, добавление новых слайдов и их содержимого, выбор разметки слайдов, изменение при необходимости оформления слайдов, изменение цветовой схемы, применение различных шаблонов оформления и создание таких эффектов, как эффекты анимации при демонстрации слайдов. Приведенные ниже сведения касаются средств, доступных на начальном этапе этого процесса.

Область задач **Создание презентации** в Microsoft PowerPoint предоставляет ряд следующих вариантов для создания новой презентации.

Создать. Слайды имеют минимум элементов оформления и цвета к ним не применены.

Создать из имеющейся презентации. Презентация создается на основе уже имеющейся презентации с заданным оформлением. Создается копия имеющейся презентации, позволяющая создать новую презентацию, внеся изменения в оформление и содержимое исходной презентации.

Создать с помощью шаблона. Презентация создается на основе имеющегося шаблона Microsoft PowerPoint, содержащего основные элементы оформления, шрифты и цветовую схему. Кроме стандартных шаблонов Microsoft PowerPoint можно использовать самостоятельно созданные шаблоны.

Шаблоны с предлагаемым содержимым. Для применения шаблона оформления, включающего предлагаемый текст для слайдов, используется мастер ав-

тосодержимого. Затем в предложенный текст вносятся необходимые изменения.

Шаблон на веб-сервере. Создание презентации с помощью шаблона, находящегося на веб-узле.

Шаблоны на Microsoft.com. Создание презентации на основе дополнительных шаблонов Microsoft PowerPoint из библиотеки шаблонов Microsoft Office. Эти шаблоны упорядочены по типам презентаций.

Примечание. Гиперссылка данного раздела ведет в Интернет. Вы можете вернуться в справку в любой момент.

Вставка содержимого из других источников

Можно также вставить слайды из других презентаций или текст из других приложений, например из MicrosoftWord.

О меню и панелях инструментов

Меню содержит список команд. Рядом с некоторыми из этих команд имеются значки, позволяющие быстро связать команду со значком. Большинство меню находится в строке меню, которая представляет собой панель инструментов, расположенную в верхней части экрана. Панели инструментов могут содержать кнопки, меню или сочетание кнопок и меню.

Отображение только используемых команд и кнопок

Microsoft Office автоматически настраивает меню и панели инструментов в соответствии с частотой использования команд. При первом запуске приложения Office отображаются все команды. Затем, по мере работы, меню и панели инструментов настраиваются таким образом, чтобы отображать только наиболее часто используемые команды.

Отображение всех команд меню

Чтобы найти команду, которая используется редко или вообще никогда ранее

не использовалась, щелкните стрелки в нижней части меню, чтобы отобразить полный список команд. Для развертывания меню также можно дважды

щелкнуть его название. При развертывании одного меню все меню отображаются полностью до тех пор, пока не будет выбрана команда или выполнено другое действие. При выборе команды в развернутом меню эта команда незамедлительно добавляется в сокращенную версию меню. Если команда будет использоваться редко, она будет удалена из сокращенной версии меню.

Расположение панелей инструментов в одной строке

Панели инструментов можно расположить друг за другом в одной строке. Например, при первом запуске приложения Microsoft Office панель инструментов **Стандартная** располагается рядом с панелью инструментов **Форматирование**. При размещении в одной строке нескольких панелей инструментов может не хватать места для отображения всех кнопок. В этом случае отображаются наиболее часто используемые кнопки.

Просмотр всех кнопок панели инструментов

Можно изменить размеры панели инструментов, чтобы на ней помещалось больше кнопок, а можно отобразить все кнопки. Для просмотра списка кнопок, не помещающихся на встроенной закрепленной панели инструментов, нажмите кнопку **Параметры панелей инструментов** в конце этой панели инструментов. При использовании кнопки, которая не отображается на панели инструментов, эта кнопка перемещается на панель инструментов, а другая кнопка, которая давно не использовалась, перемещается в список **Параметры панелей инструментов**.

Пользовательская настройка меню и панелей инструментов

Пользователи могут настраивать меню и панели инструментов по своему усмотрению: добавлять и удалять кнопки и меню панелей инструментов, создавать пользовательские панели инструментов, скрывать и отображать панели инструментов, а также перемещать их. Строку меню можно настраивать так же, как и любую встроенную панель инструментов, в частности быстро добавлять и удалять кнопки и меню, но скрыть строку меню невозможно.

Установка мастеров и шаблонов

При первой установке Microsoft Word на компьютер программа установки делает доступными многие мастеры и шаблоны. Дополнительные шаблоны и мастеры можно найти в Интернете.

Мастеры и шаблоны, поставляемые вместе с Microsoft Word

Если во время установки нажать кнопку **Установить**, некоторые из этих мастеров и шаблонов устанавливаются непосредственно на компьютер. Остальные загружаются программой установки Microsoft Windows при первом обращении к ним.

Если дважды щелкнуть шаблон или мастер в диалоговом окне **Шаблоны**, то Microsoft Word немедленно создаст новый документ, если выбранный мастер или шаблон установлены на компьютере. В противном случае программа установки сначала загрузит мастер или шаблон, и только после этого документ будет создан.

Чтобы выбрать и установить дополнительные шаблоны и мастеры, при первом запуске программы установки следует нажать кнопку **Настроить** вместо кнопки **Установить**.

Чтобы сделать доступными все шаблоны и мастеры Microsoft Word, запустите программу установки еще раз, разверните в списке пункт **Microsoft Word**, затем — **Мастеры и шаблоны** и выберите параметр **Дополнительные шаблоны и макросы**.

Мастеры и шаблоны в Интернете

При наличии доступа в Интернет можно загрузить дополнительные шаблоны и мастеры из Галереи шаблонов Microsoft Office.

Примечание. Гиперссылка в данном разделе указывает на веб-узел. В любой момент можно вернуться назад к справке.

Чтобы найти необходимые мастеры или шаблоны, следуйте инструкциям на веб-странице.

Чтобы просмотреть дополнительные возможности, при создании нового документа в области задач **Создание документа** щелкните ссылку **Шаблоны на моих веб-узлах.**

Поворот объекта

1. Выделите автофигуру, рисунок или объект WordArt, который требуется повернуть.
2. Выполните одно из следующих действий.

Поворот на произвольный угол

1. Перетащите маркер поворота объекта в нужном направлении.
2. Щелкните за пределами объекта, чтобы зафиксировать поворот.

Поворот на 90 градусов влево или вправо

- На панели инструментов **Рисование** нажмите кнопку **Действия** и выберите команду **Повернуть/отразить**, а затем — команду **Повернуть влево** или **Повернуть вправо**

Создание в рисунке прозрачных областей

Прозрачные области можно создавать в большинстве рисунков за исключением анимированных рисунков в формате GIF. Для внесения изменений в такие рисунки следует воспользоваться соответствующей программой для редактирования анимированных рисунков в формате GIF, а затем снова вставить файл.

1. Выделите рисунок, в котором требуется создать прозрачные области.
2. На панели инструментов **Настройка изображения** нажмите кнопку

   Установить прозрачный цвет .
3. Щелкните цвет, который требуется сделать прозрачным.

Примечание. Кнопка **Установить прозрачный цвет** доступна для точечных рисунков, которые еще не содержат прозрачных областей, а также для некоторых, хотя и не для всех картинок.

Применение одинаковых значений длины и ширины к различным объектам

1. Выделите все объекты (автофигуры, рисунки, объекты WordArt или картинки), для которых требуется задать одинаковые размеры.
2. В меню **Формат** выберите команду **Объект**, **Автофигура**, **Рисунок** или **Объект WordArt**.

   Название команды меню **Формат** зависит от типов выделенных объектов.
3. На вкладке **Размер** введите нужные значения в поля **Высота** и **Ширина.**

О форматировании текста с помощью стилей

Стилем называется набор параметров форматирования, который применяется к тексту, таблицам и спискам, чтобы быстро изменить их внешний вид. Стили позволяют одним действием применить сразу всю группу атрибутов форматирования.

Например, вместо форматирования названия в три приема, когда сначала задается размер 16 пунктов, затем шрифт Arial и, наконец, выравнивание по центру, то же самое можно сделать одновременно, применив стиль заголовка.

Ниже приведены типы стилей, которые можно создать и применить.

- Стиль абзаца полностью определяет внешний вид абзаца, то есть выравнивание текста, позиции табуляции, междустрочный интервал и границы, а также может включать форматирование знаков.
- Стиль знака задает форматирование выделенного фрагмента текста внутри абзаца, определяя такие параметры текста, как шрифт и размер, а также полужирное и курсивное начертание.
- Стиль таблицы задает вид границ, заливку, выравнивание текста и шрифты.
- Стиль списка применяет одинаковое выравнивание, знаки нумерации или маркеры и шрифты ко всем спискам.

С помощью области задач **Стили и форматирование** можно создать, просмотреть и применить стили. Непосредственное форматирование, применяемое к тексту, также хранится в данной области и может быть применено еще раз.

Создание гиперссылок

Если автоматическое форматирование гиперссылок не отключено, Microsoft Word автоматически создает гиперссылки при вводе адресов существующих веб-страниц, таких как www.microsoft.com.

Для создания более настроенной гиперссылки выполните одно из следующих действий, в зависимости от того, на что должна указывать данная ссылка.

Создание шаблона документа

1. Выполните одно из следующих действий.

   Создание шаблона на основе существующего документа

   1. В меню **Файл** выберите команду **Открыть**
   2. Откройте нужный документ.

   Создание нового шаблона на основе существующего шаблона

   3. В меню **Файл** выберите команду **Создать**.
   4. В области задач **Создание документа**, в группе **Создание с помощью шаблона** выберите **Общие шаблоны**.
   5. Выберите шаблон, похожий на вновь создаваемый, установите переключатель **Создать** в положение **Шаблон**, а затем нажмите кнопку **OK**.
2. В меню **Файл** выберите команду **Сохранить как**.
3. В поле **Тип файла** выберите **Шаблон документа**. При сохранении файла, созданного в качестве шаблона, тип файла «Шаблон документа» всегда устанавливается по умолчанию.
4. По умолчанию файл сохраняется в папке **Шаблоны**, выбранной в поле **Папка**. Если требуется сохранить шаблон таким образом, чтобы он отображался не на вкладке **Общие**, перейдите в подпапку папки **Шаблоны**, имя которой соответствует названию нужной вкладки.
5. В поле **Имя файла** введите имя нового шаблона и нажмите кнопку **Сохранить**.

6. Добавьте в новый шаблон текст или рисунки, которые должны появляться во всех новых документах, основанных на этом шаблоне, и удалите все те элементы, которых в документах быть не должно.
7. Измените нужным образом размеры полей и страниц, ориентацию страниц, стили и другие параметры форматирования.
8. На панели инструментов **Стандартная** нажмите кнопку **Сохранить** , а затем выберите в меню **Файл** команду **Закрыть**.

Примечания

- Гиперссылка в данном разделе ведет в Интернет, однако можно перейти назад к справке в любое время. Для получения сведений о дополнительных шаблонах и мастерах посетите Галерею шаблонов Microsoft Office.
- Чтобы сделать элементы автотекста и макросы доступными только в документах, созданных на основе данного шаблона, сохраните их в новом шаблоне, а не в шаблоне Normal.dot.

Анимация текста

1. Выделите текст, подлежащий анимации.
2. В меню **Формат** выберите команду **Шрифт**, а затем откройте вкладку **Анимация**.
3. Выберите нужный эффект в списке **Вид**.
   Для получения справочных сведений о параметре нажмите кнопку с вопросительным знаком, а затем щелкните интересующий параметр.

Примечание. Одновременное применение нескольких эффектов анимации невозможно.

Тема 2. Изучение табличного процессора MS Excel.

Работа в табличном процессоре MS EXCEL

Цель: Освоить основные приемы работы с электронными таблицами.

1. Знакомство с табличным процессором MS Excel

Табличный процессор MS Excel используется для обработки данных. Обработка включает в себя:

- проведение различных вычислений с использованием мощного аппарата функций и формул;
- исследование влияния различных факторов на данные;
- решение задач оптимизации;
- получение выборки данных, удовлетворяющих определенным критериям;
- построение графиков и диаграмм;
- статистический анализ данных.

При запуске MS Excel на экране появляется рабочая книга "Книга 1", содержащая 16 рабочих листов. Каждый лист представляет собой таблицу, состоящую из строк и столбцов. В этих таблицах хранятся данные, с которыми вам предстоит работать. Вы можете вводить в таблицы любую информацию: текст, числа, даты и время, формулы, рисунки, диаграммы, графики. Вся вводимая информация может быть обработана при помощи специальных функций. При помощи графических средств MS Excel вы можете рисовать, строить графики и диаграммы. MS Excel предоставляет в ваше распоряжение средства для работы с базами данных.

1.1. Знакомство с экраном MS Excel

На рис. 2.1 показан экран MS Excel

Строка формул используется для ввода и редактирования значений или формул в ячейках или диаграммах. MS Excel выводит в этой строке постоянное значение или формулу активной ячейки.

Поле имени - это окно слева от строки формул, в котором выводится имя ячейки, где расположен маркер. В поле имени можно задать имя ячейки или интервала ячеек. Для диаграмм поле имени идентифицирует выделенный элемент диаграммы.

Кнопки прокрутки ярлычков осуществляют прокрутку ярлычков рабочей книги. Эти кнопки не выделяют ярлычки. Для выделения ярлычка нужно нажать на нем мышью. Крайние кнопки осуществляют прокрутку к первому и послед-

нему ярлычку рабочей книги. Внутренние кнопки осуществляют прокрутку к предыдущему и следующему ярлычку рабочей книги. Для прокрутки нескольких ярлычков за один раз удерживайте нажатой клавишу Shift во время нажатия мышью кнопок прокрутки. 2.1.3. Панели инструментов в окне MS Excel

В главном окне MS Excel на рис. 2.1 вы видите две панели инструментов. Наиболее простой способ получения информации о кнопках панели состоит в следующем. Вы перемещаете указатель мыши на интересующую вас кнопку. При этом рядом с кнопкой появляется ее название в окошке желтого цвета, а в левой части строки состояния - краткая характеристика действий, которые произойдут при нажатии этой кнопки. Кнопки панели инструментов "Форматирование" выполняют функции обработки информации в ячейках, устанавливают шрифт и его размер, цвет ячеек и шрифта, стиль текста и рамок.

2.1.6. Работа с файлами в MS Excel

В MS Excel под рабочую книгу отводится один файл со стандартным расширением .XLS. Команды работы с файлами собраны в пункте Файл основного меню.

2.2. Ввод и редактирование данных

Любая обработка информации начинается с ее ввода в компьютер. В MS Excel вы можете вводить числа, текст, даты, время, последовательные ряды данных и формулы. После того как данные введены, возникает вопрос о том, в каком виде их представлять на экране. Для представления данных в MS Excel существуют различные категории форматных кодов.

Расположение данных в таблицах предполагает наличие у них заголовков.

Добавление столбца. Установите маркер в какой-нибудь ячейке столбца, перед которым нужно вставить новый столбец, и выполните команду Вставка | Столбец. Аналогично для вставки строки выполните команду Вставка | Строка. Помимо столбцов и строк в таблицу можно вставлять и ячейки. Для этого выполните команду Вставка | Ячейки.

2.2.2. Формат данных

Данные в MS Excel выводятся на экран в определенном формате. По умолчанию информация выводится в формате Общий. Можно изменить формат представления информации в выделенных ячейках. Для этого выполните команду Формат | Ячейки. Появится окно диалога "Формат ячеек",в котором нужно выбрать вкладку "Число". В левой части окна диалога "Формат ячеек" в списке "Числовые форматы" приведены названия всех используемых в MS Excel форматов.

Рис. 2.1. Экран MS Excel

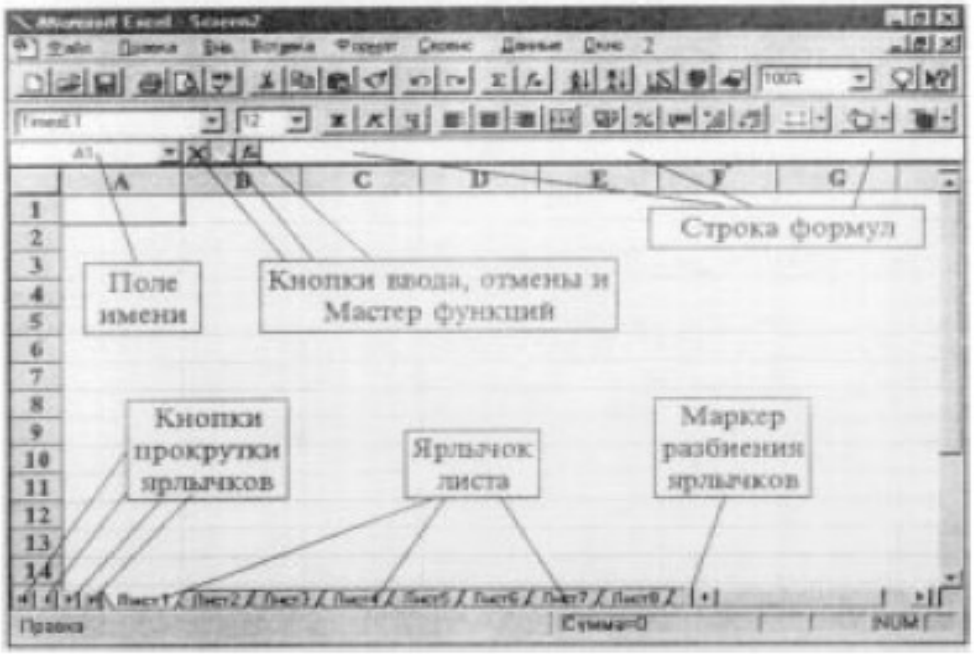


Рис. 2.2. Главное окно MS Excel с активной строкой формул

Ввод чисел и текста

Любую информацию, которую мы обрабатываем на компьютерах, можно представить в виде чисел или текста.

Ввод чисел. При работе с числами важно уметь легко изменять вид вводимых чисел: число знаков после запятой, вид целой части, порядок и знак числа. При вводе чисел можно использовать только следующие символы: 1234567890-+/,Ee. MS Excel помогает вам определить, является ли введенная информация числом. Числа в MS Excel отображаются в категориях Числовой, Экспоненциальный, Финансовый, Денежный, Процентный, Дробный. Если вы хотите вводить числа, не связывая себя какими-либо форматами, то MS Excel будет их вводить в коде формата Общий.

Вы имеете возможность создавать собственные числовые форматы при помощи следующих символов: "#", "О", "?", "," (запятая), "_" (символ подчеркивания), " " (пробел), "%" (знак процента). Допускается также использование различных цветов для выделения определенных чисел. Пробел, следующий за знаком шаблона, масштабирует число, округляя его до тысяч.

Символ "_" (подчеркивание) используется для пропуска места на ширину символа. Он выводит пробел на ширину, соответствующую ширине следующего за ним символа шаблона.

Для использования символа "%" существует специальный числовой формат - Процентный. При построении пользовательских экспоненциальных форматов мы будем использовать следующие символы:E-, E+, e-, e+.

Ввод текста. Любая последовательность введенных в ячейку символов, которая не может быть интерпретирована MS Excel как число, формула, дата, время дня, логическое значение или значение ошибки, интерпретируется как текст. Введенный текст выравнивается в ячейке по левому краю.

Чтобы ввести текст, выделите ячейку и наберите текст с клавиатуры. Ячейка может вмещать до 255 символов.

Стиль представления данных

Одним из способов упорядочения данных является введение стиля. Вы вводите телефонные номера - один стиль, заполняете таблицу продаж -другой стиль, вводите данные в телефонно-адресную книгу - третий стиль. После этого для изменения представления данных достаточно только изменить нужный стиль и данные, отображаемые этим стилем, изменятся автоматически.

Создание стиля. Для создания стиля используется команда Формат | Стиль. Выполнение этой команды открывает окно диалога "Стиль".

Ввод последовательных рядов данных

В MS Excel разработан механизм ввода рядов данных. Под рядами данных подразумеваются данные, отличающиеся друг от друга на фиксированный шаг. При этом данные не обязательно должны быть числовыми.

Для создания рядов данных выполните следующие действия:

1. Введите в ячейку первый член ряда.

2. Выделите область, где будет расположен ряд. Для этого нужно подвести указатель мыши к черной точке в правом нижнем углу выделенной ячейки (в этот момент белый крестик переходит в черный) и нажать левую кнопку мыши. Далее, удерживая нажатой кнопку мыши, выделите нужную часть строки или столбца. После того как вы отпустите кнопку мыши, выделенная область заполнится данными. Можно построить ряд данных и другим способом, если указать шаг построения. Для этого нужно ввести вручную второй член будущего ряда, выделить обе ячейки и продолжить выделение до нужной области. Две первых ячейки, введенные вручную, задают шаг ряда данных.

Формирование заголовков таблиц

Сначала нужно ввести общий заголовок таблицы, а затем названия полей. Они должны находиться в одной строке и следовать друг за другом. Вы можете расположить заголовок таблицы для чтения снизу-вверх, сверху-вниз, выровнять по правому, левому, нижнему или верхнему краю ячейки.

Общее правило форматирования выглядит следующим образом: выделите нужную область и нажатием Ctrl+1 вызывайте средства форматирования ячейки -

окно диалога "Формат ячеек", на вкладке "Выравнивание",в области "Ориентация" выберите вариант отображения текста.

Кнопка стандартной панели управления "Центрировать по столбцам" центрирует текст самой левой ячейки горизонтально в пределах всех выделенных ячеек, находящихся справа. При выборе кнопки "Центрировать по столбцам" данные центрируются по столбцам, даже если ширина столбцов потом изменяется. После центрирования данные по-прежнему принадлежат самой левой ячейке. Повторное нажатие кнопки устанавливает обычное выравнивание для ячеек.

Выбор рамок выполняет кнопка "Линии рамки". Чтобы убрать рамки, нужно выделить те ячейки, в которых это нужно сделать, и нажать верхнюю левую клетку в списке стилей кнопки "Линии рамки".

При печати часто возникает необходимость убрать сетку линий таблицы. Для этого нажмите кнопку "Сетка", расположенную в категории кнопок "Элементы управления". В качестве альтернативного способа удаления линий сетки вы можете воспользоваться командой Параметры | Сервис. На вкладке "Вид" окна диалога "Параметры" снимите флажок "Сетка". После нажатия кнопки ОК сетка исчезнет.

Работа с функциями и формулами

Формулой в MS Excel называется последовательность символов, начинающаяся со знака равенства "=" или "+". В эту последовательность символов могут входить постоянные значения, ссылки на ячейки, имена, функции или операторы. Результатом работы формулы является новое значение, которое выводится как результат вычисления формулы по уже имеющимся данным. Если значения в ячейках, на которые есть ссылки в формулах, меняются, то результат изменится автоматически. В строке формул отражается содержимое ячейки (формула), в которой расположен курсор.

Внесение изменений в формулу

Вы можете редактировать как параметры функций, используемых в формуле, так и непосредственно символы формулы. Для внесения изменений в формулу нажмите мышью на строке формул или клавишу F2. Затем внесите изменения

и нажмите кнопку ввода в строке формул или клавишу Enter. Если вы хотите внести изменения в формулу непосредственно в ячейке, где она записана, то дважды нажмите мышью на ячейке с этой формулой.

Для отмены изменений нажмите кнопку "Отмена" в строке формул или клавишу Esc.

Вы можете вносить изменения в параметры функции с помощью мастера функций. Для этого установите курсор в ячейку с формулой и нажмите кнопку "Мастер функций", расположенную на панели инструментов Стандартная. Появится окно диалога "Изменение функций". Внесите необходимые изменения и нажмите кнопку Готово или ДалееX

Значения ошибок в формулах

MS Excel выводит в ячейку значение ошибки, когда формула для этой ячейки не может быть правильно вычислена. Если формула содержит ссылку на ячейку, которая содержит значение ошибки, то эта формула также будет выводить значение ошибки (за исключением тех случаев, когда используются специальные функции рабочих листов ЕОШ, ЕОШИБКА или ЕНД, которые проверяют наличие значений ошибок).

Перемещение и копирование формул

После того как формула введена в ячейку, вы можете ее перенести, скопировать или распространить на блок ячеек. При перемещении формулы в новое место таблицы ссылки в формуле не изменяются, а ячейка, где раньше была формула, становится свободной. При копировании формула перемещается в другое место таблицы, ссылки изменяются, но ячейка, где раньше находилась формула, остается без изменения. Формулу можно распространить на блок ячеек.

При копировании формул возникает необходимость управлять изменением адресов ячеек или ссылок. Изменятся только те атрибуты адреса ячейки, перед которыми не стоит символ "$". Если перед всеми атрибутами адреса ячейки поставить символ "$", то при копировании формулы ссылка не изменится.

Для перемещения формулы или блока формул подведите указатель мыши к тому месту границы ячейки или блока, где изображение указателя мыши изменяется с белого крестика на белую стрелку. Затем нажмите левую кнопку мыши и, удерживая ее, перемещайте ячейку или блок в нужное место таблицы. Завершив перемещение, отпустите кнопку мыши. Если в записи формулы есть адреса ячеек, они при перемещении формулы не изменяются.

Для копирования формулы или блока формул подведите указатель мыши к тому месту границы ячейки или блока, где изображение указателя изменяется с белого крестика на белую стрелку. Затем нажмите клавишу Ctrl и левую кнопку мыши и, удерживая клавишу и кнопку нажатыми, перемещайте ячейку или блок в нужное место таблицы. Для завершения копирования отпустите кнопку мыши и клавишу Ctrl. Если в записи формулы есть относительные адреса ячеек, при копировании формулы они изменятся.

Распространение формул

Помимо копирования и перемещения формулу можно распространить на часть строки или столбца. При этом происходит изменение относительных ссылок. Для распространения формулы выполните следующие действия:

1. Установите курсор в ячейку с формулой.
2. Подведите указатель мыши к черной точке на границе ячейки. Изображение указателя изменяется на черный крестик.
3. Нажмите левую кнопку мыши и, удерживая ее нажатой, перемещайте курсор до нужного места. Для завершения распространения формулы отпустите кнопку.

Распространение блока формул происходит по той же схеме. В случае распространения одной формулы на блок ячеек сначала распространите ее на часть строки или столбца и далее на весь блок.

Функции даты и времени

Представление даты и времени имеет одну особенность. При вводе даты или времени вы вводите последовательность символов, которая не является числом, но с этими символами можно производить вычисления: сравнивать, при-

бавлять, вычитать. Поэтому в MS Excel, наряду с текстовым представлением даты и времени, существует и числовое представление.

В действительности Excel хранит даты и время в ячейках в виде так называемых сериальных чисел. Сериальное число даты представляет собой порядковый номер данного дня в столетии, а сериальное число времени -долю от 24-часового дня.

Наличие текстового и числового форматов представления даты и времени практически не затрудняет работу. В ячейке с форматом «Общий» результат будет представлен в нужном виде без вашего участия, а если ячейки ранее были отформатированы для чисел, то результатом работы функций дат и времени будет число. В этом случае вам придется самостоятельно позаботиться о введении нужного формата.

Примером может быть функция расчета количества дней между заданными датами или формирования даты последнего дня указанного месяца.

Логические функции

Логические функции являются неотъемлемым компонентом многих формул. Всякий раз, когда вам необходимо выполнить те или иные действия в зависимости от выполнения каких-либо условий, вы используете логические функции. В MS Excel вы можете использовать следующие логические функции: ЕСЛИ, И, ИЛИ, ИСТИНА, ЛОЖЬ, НЕ. Результатом работы логических функций И, ИЛИ, ИСТИНА, ЛОЖЬ, НЕ является логическое значение ИСТИНА или ЛОЖЬ, а результатом работы логической функции ЕСЛИ может быть число, текст или ссылка на выполнение каких-либо действий.

Функция НЕ (арг) изменяет на противоположное логическое значение своего аргумента.

Аргумент функции арг - это значение или выражение, результатом вычисления которого является ИСТИНА или ЛОЖЬ.

Диаграммы и графики

Процедура построения графиков и диаграмм в MS Excel отличается как широкими возможностями, так и необычайной легкостью. Любые данные в таблице

всегда можно представить в графическом виде. Для этого используется Мастер диаграмм, который вызывается нажатием на кнопку с таким же названием, расположенную на стандартной панели управления.

После нажатия кнопки "Мастер диаграмм" нужно выделить на рабочем листе место для размещения диаграммы. Установите курсор мыши на любой из углов создаваемой области, нажмите кнопку мыши и, удерживая ее нажатой, выделите прямоугольную область. После того как вы отпустите кнопку мыши, вам будет предложена процедура построения диаграммы, состоящая из пяти шагов. На любом шаге вы можете нажать кнопку "Готово", в результате чего построение диаграммы завершится. С помощью кнопок "Далее" и "Назад", вы можете управлять процессом построения диаграммы.

Кроме того, для построения диаграммы можно воспользоваться командой Вставка | Диаграмма.

Если выделена область данных, самый быстрый способ построить диаграмму - нажать клавишу F11.

Построив диаграмму, вы можете добавлять и удалять ряды данных. Для удаления данных нужно выделить их на диаграмме и нажать клавишу Del или выполнить команду Правка | Очистить | Ряды. Команда Вставка | Новые данные позволяет добавить на диаграмму новые данные.

В процессе построения диаграммы вам предстоит определить место, где будет расположена диаграмма, и ее тип. Вы должны также определить, где и какие надписи должны присутствовать на диаграмме. В результате вы получаете хорошую заготовку для дальнейшей работы.

Термин «диаграмма активна» означает, что в углах и на серединах сторон поля диаграммы расположены маркеры, которые имеют вид маленьких черных квадратиков. Диаграмма становится активной, если вы нажмете кнопку мыши в любом месте диаграммы (предполагается, что вы находитесь вне диаграммы, то есть курсор установлен в ячейке активного листа книги). Когда диаграмма активна, вы можете изменять размеры поля и перемещать ее по рабочему листу.

По умолчанию диаграмма строится по всей выделенной области, то есть считается, что строки и столбцы под метки не выделены. Однако когда в верхней строке и в левом столбце выделенной области находится текст, MS Excel автоматически выделяет их под метки. Метки столбцов являются текстом легенды. Термин "Легенда" обозначает прямоугольник, в котором указывается, каким цветом или типом линий отображаются на графике или диаграмме данные из той или иной строки. Если 1-й столбец не содержит меток, MS Excel вводит собственные метки: "ряд1", "ряд2" и т.д.

После построения графика нужно его отредактировать, в частности, изменить цвет и стиль линий, которыми изображены серии чисел, расположенные в строках тестовой таблицы. Первое, что нужно сделать, это перейти в режим редактирования диаграммы. Для этого нужно дважды нажать кнопку мыши на диаграмме. Изменится обрамление диаграммы, появится бордюр. Это свидетельствует о том, что вы находитесь в режиме редактирования диаграммы. В качестве альтернативного способа перехода в режим редактирования диаграммы нажмите правую кнопку мыши, когда ее указатель находится на диаграмме, и в появившемся списке команд выберите команду «Редактировать объект».

Одним из способов изменения стиля линий является изменение стиля обозначений в легенде. Чтобы войти в режим редактирования линий легенды и внести необходимые изменения, выполните следующие действия:

1. Нажмите кнопку мыши на изображении легенды. В углах и на серединах сторон легенды появятся маленькие черные квадратики. Вы вошли в режим редактирования легенды.

2. Нажмите кнопку мыши на изображении линии. Черные квадратики появятся вокруг линии.

3. Нажмите правую кнопку мыши. Появится меню из двух команд «Очистить» и «Формат обозначения легенды».

4. Выберите команду «Формат обозначения легенды». Появляется окно диалога "Форматирование условного обозначения" и вы можете установить в нем цвет линий, их стиль, толщину.

5. Нажатие кнопки ОК приводит к выходу из режима редактирования легенды в режим редактирования диаграммы.

Вы можете изменять размеры графика, перемещать текст, редактировать любые объекты диаграммы. Признаком режима редактирования являются черные квадратики внутри диаграммы. Для выхода из режима редактирования диаграммы достаточно нажать мышью вне диаграммы.

Перемещение объектов диаграммы. Перемещение объектов диаграммы выполняется в режиме редактирования диаграммы. Перейдите в него. Для перемещения объекта диаграммы выполните следующие действия:

1. Нажмите мышью на объекте, который вы хотите переместить. При этом вокруг объекта появляется окаймление из черных квадратиков.

2. Подведите курсор к границе объекта и нажмите кнопку мыши. Появится прерывистая рамка.

3. Переместите объект в нужное место (перемещение осуществляется курсором мыши), удерживая нажатой кнопку мыши, после чего отпустите кнопку мыши. Объект переместился.

Для изменения размеров поля, на котором находится какой-либо из объектов диаграммы, выполняются следующие действия:

1. Нажмите мышью на объекте, размеры которого вы хотите изменить. При этом вокруг объекта появляется окаймление из черных квадратиков.

2. Переместите указатель мыши в черный квадратик на той стороне объекта, которую вы собираетесь изменять, или в угол объекта. При этом белая стрелка переходит в двунаправленную черную стрелку.

3. Нажмите кнопку мыши и удерживайте ее нажатой. Появится прерывистая рамка.

4. Переместите границу объекта (перемещение осуществляется курсором мыши), удерживая нажатой кнопку мыши, в нужное место и отпустите кнопку мыши. Размеры объекта изменились. Если размеры объекта вас опять не устраивают, повторите операцию.

Размеры и положение диаграммы изменяются аналогично. Операция форматирования для любых объектов выполняется по следующей схеме:

1. Нажмите правую кнопку мыши на объекте, который нужно форматировать. Появляется список команд, который зависит от выбранного объекта.

2. Выберите команду для форматирования: -форматировать название диаграммы;

- форматировать легенду;

- форматировать ось; -форматировать область построения.

После выбора любой из этих команд появляется окно диалога для форматирования объекта, в котором, используя стандартную технику MS Excel, вы можете выбирать шрифты, размеры, стили, форматы, заполнения и цвета.

Для изменения заполнения области построения диаграммы нажмите на ней правую кнопку мыши и в появившемся списке выберите опцию «Формат» области построения. В открывшемся окне диалога выберите подходящее заполнение.

При работе с графикой MS Excel вы можете заменить один ряд данных другим в построенной диаграмме и, изменяя данные на диаграмме, соответственно корректировать исходные данные в таблице.

В окне диалога "Форматирование ряда данных" вы можете не только изменять значения данных, но и корректировать метки оси OX. Для этого нужно выбрать вкладку "Значение X" и внести соответствующие изменения. Во вкладке "Вид" можно изменить стиль, цвет и толщину линий, которыми изображаются на графике ряды данных. Вы можете ввести величину погрешности значений во вкладке "Y-погрешность". Кроме того, вы можете также отформатировать метки данных в соответствующей вкладке.

Изменение данных посредством изменения графика. В MS Excel вы можете подобрать внешний вид диаграммы и соответствующим образом откорректированную таблицу исходных данных.

Данные в таблице и линии графика связаны между собой. Если вы изменяете данные в таблице, то соответственно изменяются и линии графики. Это совер-

шенно естественно. Но, оказывается, можно поступать и наоборот: вы изменяете линии на графике и соответственно изменяются данные в таблице. Можно увеличивать и уменьшать значения данных ряда без всяких ограничений. MS Excel автоматически растягивает ось OY так, чтобы новое значение ряда не выходило за пределы графика.

Для восстановления информации в режиме редактирования воспользуйтесь Ctrl+Z, но это возможно только тогда, когда вы не успели выполнить еще какие-либо действия.

В MS Excel можно строить объемные и плоские диаграммы. Всего насчитывается 102 вида различных диаграмм и графиков. Существуют следующие типы плоских диаграмм: линейчатая, гистограмма, с областями, график, круговая, кольцевая, лепестковая, ху-точечная и смешанная. Объемные диаграммы можно строить следующих типов: линейчатая, гистограмма, с областями, график, круговая и поверхность. У каждого типа диаграммы, как у плоской, так и у объемной, существуют подтипы.

Требуемый тип выбирается на шагах 2 и 3 процесса построения. В режиме редактирования диаграмм нужный тип можно выбрать при помощи команды «Автоформат».

Линейчатые диаграммы и гистограммы похожи по внешнему виду и способам представления данных, отличаясь друг от друга в основном ориентацией: в линейчатых диаграммах ось OX, или ось меток, располагается вертикально, ось OY - горизонтально; в гистограммах расположение осей обратное. Поэтому из этих двух типов диаграмм более подробно рассмотрим только линейчатые.

Два важных подтипа линейчатой диаграммы заслуживают особого внимания (рис. 2.3). На левом варианте диаграммы четыре горизонтальных линейки соответствуют четырем столбцам исходных данных. Строкам исходных данных соответствуют цвета закраски отрезков линеек. Диаграмма данного типа строится так, что все линейки имеют одинаковую длину, поэтому диаграмма в целом показывает процентное распределение суммы столбца по строкам. На правом варианте диаграммы линейки также соответствуют столбцам, но все данные

изображены в одном масштабе и поэтому длина линеек показывает сумму данных столбца.

Линейчатая диаграмма имеет 10 подтипов, из которых вы практически всегда можете выбрать наиболее подходящий вид для графического отображения ваших данных.

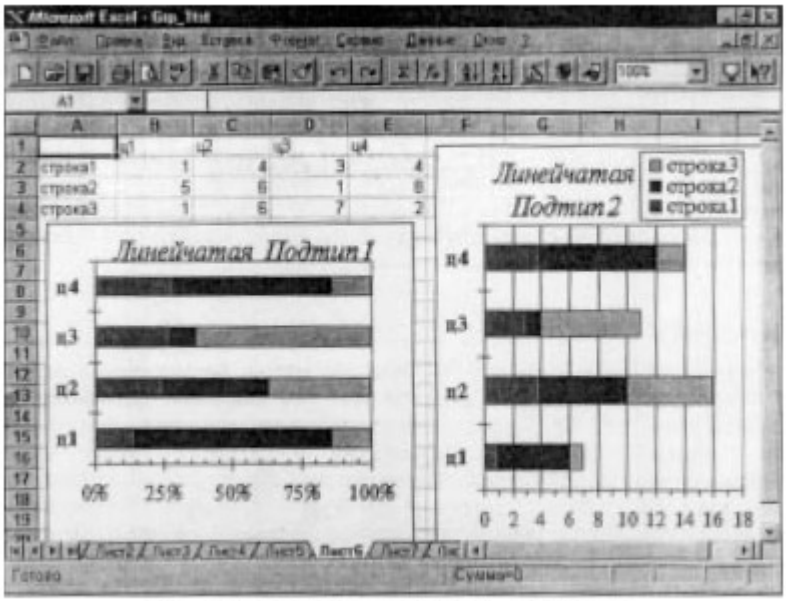


Рис. 2.3. Подтипы линейчатой диаграммы

Характерной особенностью диаграмм с областями является то, что области, ограниченные значениями в рядах данных, заполняются штриховкой. Величины следующего ряда данных не изменяются по величине, но откладываются от значений предыдущего ряда. Это иллюстрирует левый график на рис. 2.4. На правом графике рис. 2.4 показано долевое (в процентах) распределение по каждому столбцу с заполнением.

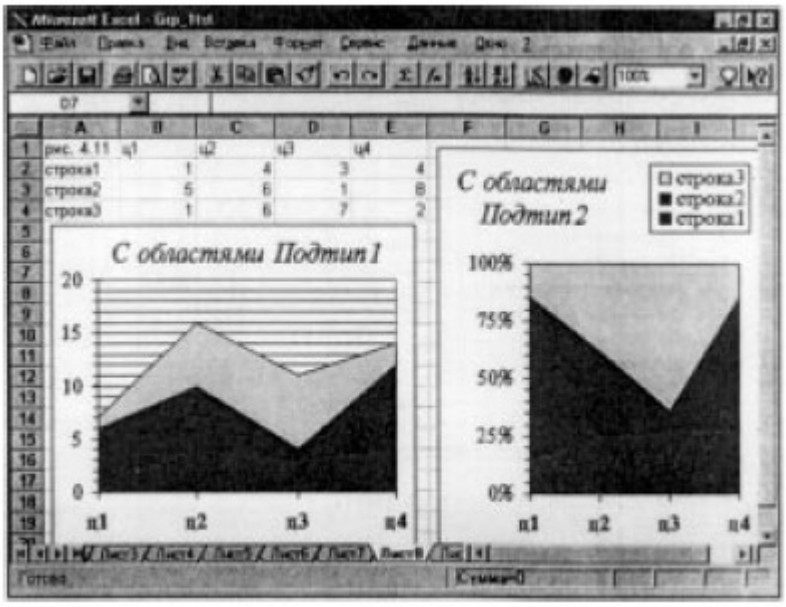


Рис. 2.4. Диаграммы с областями

Для определения, какой ряд данных оказывает самое сильное влияние на суммарный итог, нужно поставить поочередно на нижнее место все ряды данных. Для этой цели в окне диалога "Форматирование диаграмм с областями" перейдите на вкладку "Порядок рядов" и выберите кнопку Поместить вниз.

Кольцевые диаграммы отличаются от круговых только тем, чем отличается кольцо от круга - наличием в середине пустого пространства. Выбор нужного типа определяется соображениями целесообразности, наглядности и т.д. С точки зрения техники построения отличия отсутствуют. При помощи круговой диаграммы вы можете показать только один ряд данных. Каждому элементу ряда данных соответствует сектор круга.

Подтипы круговых диаграмм.На круговых диаграммах вы можете изменять величину отдельного сектора, также как изменяли ранее размеры столбца на гистограмме и расположение линии на графике.

Трехмерная графика. В MS Excel трехмерная графика представлена шестью типами трехмерных диаграмм: гистограмма, линейчатый, с областями, графика, круговая и поверхность. Для получения трехмерной диаграммы нужно на втором шаге построения диаграммы выбрать пространственный образец. После пятого шага мы получили трехмерную диаграмму, которая изображена на рис. 2.5.

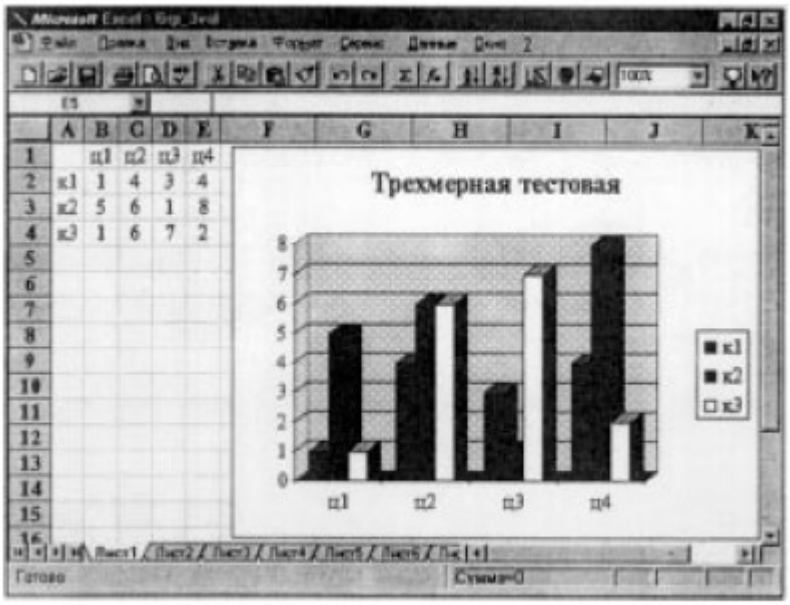


Рис. 2.5. Трехмерная гистограмма

К трехмерной диаграмме можно перейти и в режиме редактирования диаграммы. Для этого нужно установить флажок «Объемная» в тех режимах, где изменяется тип диаграммы.

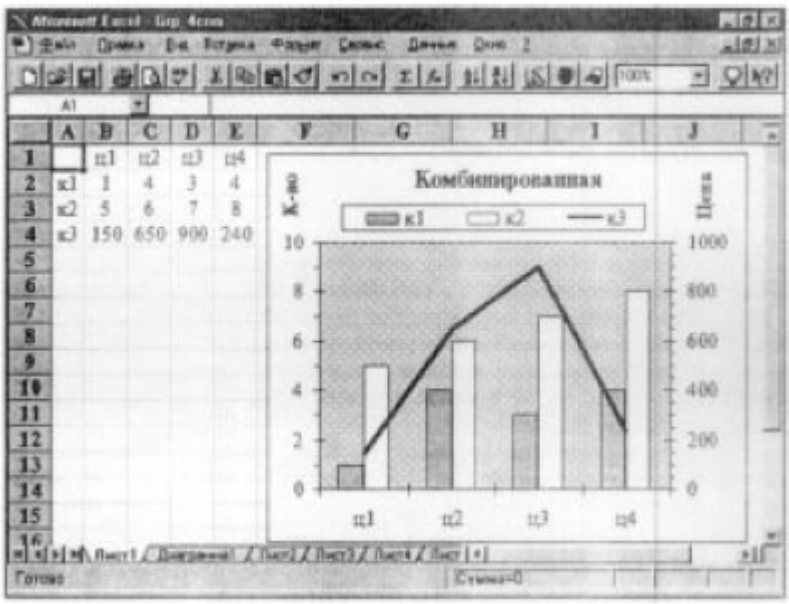


Рис. 2.6. Количество товара и его цена

Можно комбинировать различные типы только плоских диаграмм с помощью правой кнопки мыщи. При использовании объемных диаграмм комбинирование невозможно.

Работа с базами данных

В общем смысле термин "база данных" можно применить к любой совокупности связанной информации, объединенной вместе по определенному признаку. Например, в качестве базы данных можно рассматривать расписание движения поездов или книгу регистрации данных о заказах покупателей и выполнении заказов. Основным назначением баз данных является быстрый поиск содержащейся в них информации. Если у вас есть информация о продажах компьютеров, то с помощью инструментов MS Excel, представляемых в ваше распоряжение для обработки баз данных, вы можете не только выяснить, какая из моделей компьютеров имела наибольший спрос у покупателей за последнее время, но, что важнее, проследить измененииеспроса и определить тенденцию продаж каждой модели компьютера.

MS Excel содержит достаточно широкий набор средств для обеспечения эффективного управления базами данных:

- организация ввода и просмотра данных;
- сортировка, фильтрация и консолидация данных в таблицах;
- подведение итогов и сводная таблица.

В MS Excel базы данных размещаются в таблицах. Каждая таблица состоит из строк и столбцов, которые в базах данных называются записями и полями соответственно. Информация в базах данных имеет постоянную структуру. Каждую строку можно рассматривать как единичную запись. Информация в пределах каждой записи содержится в полях.

При работе с базами данных в MS Excel прежде всего нужно ввести заголовки столбцов. После этого вы можете ввести информацию в базу данных. Ввод данных и просмотр информации осуществляется с помощью команды Данные | Форма.

Команда "Форма" отображает на экране форму, которая представляет собой окно диалога, предназначенное для просмотра и редактирования записей в базе данных, а также для добавления новых и удаления существующих записей. Кроме того, с помощью формы вы можете осуществить поиск конкретных записей на основании сложных критериев.

Просмотр записей всегда начинается с текущей записи. Если окажется, что впереди нет записей, удовлетворяющих критериям, то обязательно нажмите клавишу "Далее", поскольку нужные записи могут оказаться выше текущей.

После ввода данных вам может потребоваться упорядочить их. Процесс упорядочения записей в базе данных называется сортировкой. Порядок сортировки записей определяется конкретной задачей. При сортировке изменяется порядок следования записей в базе данных или таблице. Таким образом происходит изменение базы данных. Вы должны иметь возможность восстановить исходный порядок следования записей. Универсальным средством для этого является введение порядковых номеров записей. В сочетании со средствами MS Excel по восстановлению данных это полностью защитит вашу базу от потерь при случайных сбоях в работе.

Задание

1) Создать таблицу из двух столбцов по 20 элементов. Ввести название таблицы и название столбцов. Элементам первого столбца назначить тип «Числовой», а элементам второго «денежный».

2) Добавить в таблицу третий столбец и с помощью команды Формат|Стиль определить его ячейкам тип «Процентный».

3) Отдельно от таблицы создать ряд чисел от 1 до 40 с шагом 3. Создать текстовый ряд с последовательностью a,b,c.

4) Создайте таблицу из двух столбцов по 20 числовых элементов и с помощью формул создайте рядом третий столбец каждый элемент которого является суммой элементов первых двух таблиц на соответствующей ему строке. С помощью копирования формул создать четвертый столбец, использовать формулу для третьего столбца.

5) Создать таблицу из двух столбцов по двадцать логических элементов и с помощью формул сделать операции ЕСЛИ, И, ИЛИ, НЕ.

6) Создать таблицу из двух столбцов по 15 элементов, первый столбец произвольно заполнить элементами числового типа от 1 до 10, второй столбец произвольно заполнить элементами числового типа от 1 до 100. Создать следующие диаграммы: гистограмму, график, круговую диаграмму, точечную диаграмму, диаграмму с областями, комбинированную диаграмму.

7) Изменить данные для 5 элементов в таблице с помощью изменения графика. На графике поменять цвет линий.

8) Используя второй столбец таблицы для графиков найдите сумму элементов, среднее, максимум, минимум. Выполните сортировку от большего к меньшему и от меньшего к большему.

Тема 3. Изучение СУБД MS Access.

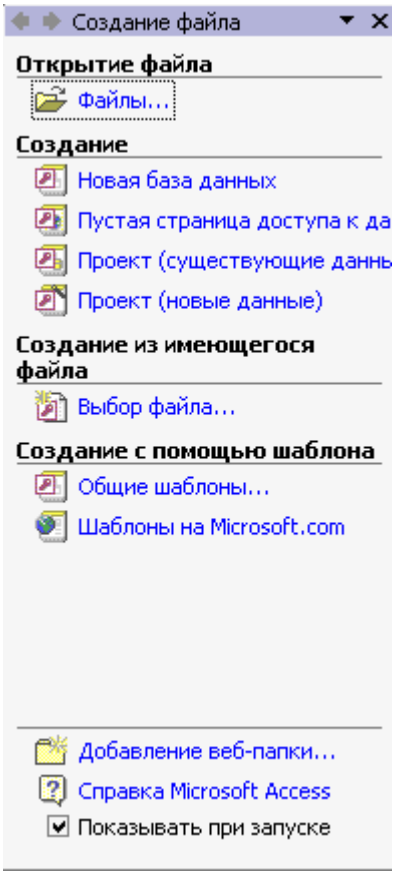
Цель лабораторной работы заключается в том, чтобы научиться:
-открывать базу данных
-создавать новую базу данных
-работать с панелями инструментов
-создавать таблицы при помощи Мастера таблиц
-использовать справочную систему Access 2000

Открытие базы данных при запуске Access 2000

Чтобы запустить Access 2000 и открыть в нем базу данных Борей, которая разработана для автоматизации деятельности вымышленной торговой компании Борей, специализирующейся на продаже продуктов питания:

1. Нажмите кнопку “Пуск” в системе Windows, выберите в нем пункт “Программы ”, в котором выберите ”Microsoft Access”.

В процессе запуска на экране появляется главное окно Access 2000 и диалоговое окно “Microsoft Access”, позволяющее создать базу данных или открыть существующую базу данных (рис.1.1).



В нижней части диалогового окна “Microsoft Access” выводится список ранее открывавшихся баз данных. Если Access 2000 был запущен первый раз, то в списке появится только элемент “Другие файлы”.

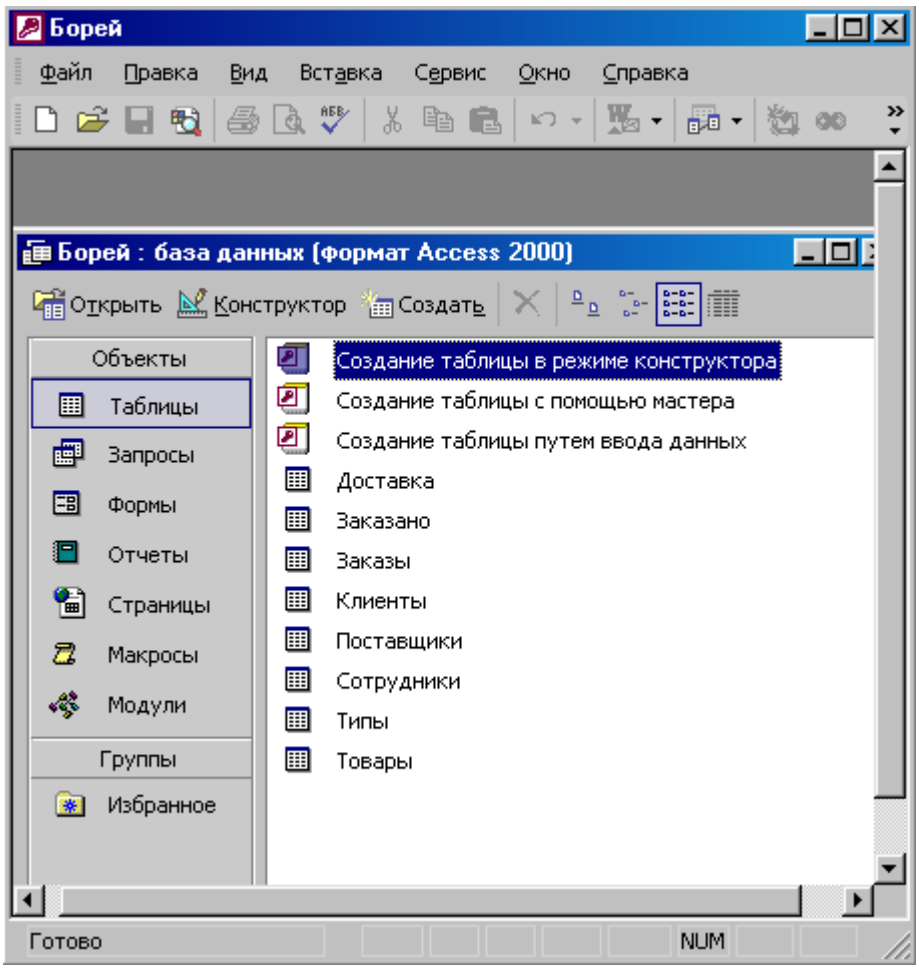
2. Выберите переключатель “Открыть базу данных”.

3. В списке баз данных выделите элемент “Другие файлы”.

Нажмите кнопку ОК. появится стандартное диалоговое окно “Открытие файла базы данных”.

4. Файл базы данных Борей находится по умолчанию в папке \PROGRAM FILES\MICROSOFT OFFICE\OFFICE\SAMPLES.

5. В главном окне Access 2000 появится окно базы данных Борей, в котором находятся все необходимые элементы: Таблицы, Запросы, Формы, Отчеты, Страницы, Макросы и Модули. (рис. 1.2).



Открытие базы данных после запуска Access 2000

Открытие новой базы данных влечет за собой автоматическое закрытие уже открытой.

Чтобы открыть базу данных:

1. Нажмите кнопку **“Открыть”** на панели инструментов или выберите команду “Файл, Открыть”. Появится диалоговое окно “Открытие файла базы данных”, где следует выбрать нужный вам файл базы данных.

2. Выбрать требуемый файл в меню <<Файл>> из списка ранее открывавшихся файлов.

Создание новой базы данных.

1 способ: если Access 2000 запущен, то закройте его и снова запустите.

1. В диалоговом окне <<Microsoft Access>> (рис. 1.1) выберите переключатель **<<Новая база данных>>** и нажмите кнопку **ОК**.

2. Появится диалоговое окно “Файл новой базы данных”. По умолчанию, Access 2000 присваивает новой базе данных имя dbl, а если база с таким именем уже существует, то db2 и т.д.

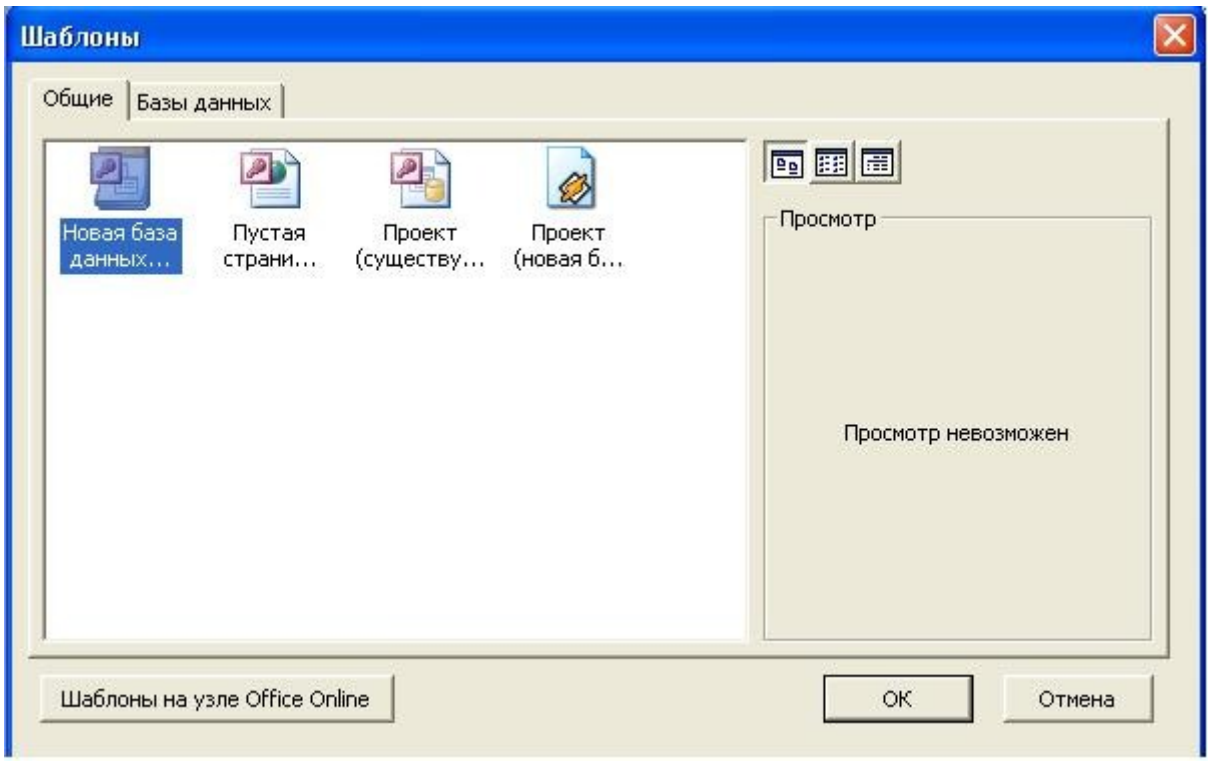
3. В раскрывающемся списке <<Папка>> откройте папку, в которой будет храниться создаваемая база данных. В поле ввода “Имя файла” укажите имя новой базы данных и не добавляйте расширение- Access 2000 автоматически добавит расширение .mdb к имени файла.

4. Нажмите кнопку **“Создать”** или клавишу Enter для создания базы данных. Если в этот момент была открыта другая база данных, то Access 2000 закроет ее и начнет создавать новую. После завершения процесса создания в главном окне Access 2000 появится окно новой базы данных.

2 способ: если Access 2000 уже был запущен:

1. В его главном окне нажмите кнопку “Создать” на панели инструментов или выберите команду <<Файл, Создать>>. Появится диалоговое окно “Создание” (рис. 1.4).раскройте вкладку “Общие”, выделите элемент <<База данных>> и нажмите кнопку ОК. продолжайте создание с шага 2 предыдущего способа.

2. Вкладка “База данных” позволяет использовать один из 10 шаблонов, на основе которых могут быть созданы наиболее популярные базы данных.



3 способ: если Access 2000 запущен, то закройте его и снова запустите.

При запуске Access 2000 в диалоговом окне <<Microsoft Access>> (рис. 1.1) выберите переключатель **<<Мастера, страницы и проекты базы данных>>** и нажмите кнопку **ОК.** появится диалоговое окно Создание (рис. 1.4) с раскрытой вкладкой Базы данных. Выберите один из шаблонов и нажмите кнопку **ОК**, а дальше пошагово выполняйте все инструкции для создания типичной базы данных.

Создание экспериментальной копии базы данных Борей.

Для выполнения примеров следует создать копию базы данных Борей. Это желательно сделать потому, что при работе с копией базы данных не нужно будет беспокоится о том, как повлияют внесенные изменения на работу приложения в целом. При создании копии базы данных можно воспользоваться операцией сжатия базы данных:

1. Закройте все окна объектов базы данных, для этого выберите команду <<Файл, Закрыть>> или закройте окно базы данных. Окно Microsoft Access 2000станет пустым.

2. Выберите команду <<Сервис, Служебные программы, Сжать и восстановить базу данных>>. Появится диалоговое окно <<База данных для сжатия>>. Найдите и выберите файл базы данных Борей, нажмите кнопку <<Сжать>>.

3. Появится диалоговое окно <<Сжатие базы данных под именем>>.

4. Если надо, то перейдите в папку, в которой должна быть сохранена сжатая копия базы данных Борей. В поле ввода Имя файла введите свою фамилию. Нажмите кнопку Сохранить. На диске будет создана сжатая копия базы данных, с которой вы будете дальше работать и которую можно использовать для тестирования.

5. Выберите команду Файл, Открыть и щелкните по значку созданной базы данных.

Теперь везде, где будет говориться о базе данных Борей-используйте созданную базу данных.

Элементы базы данных Access 2000

В состав любой базы данных Access 2000 входят следующие элементы:

1. Таблицы.

В базах данных Access 2000 информация хранится в виде двумерных таблиц, строки которых называются записями, а столбцы полями, таблицы должны быть связаны между собой по значениям ключевых полей. База данных Access 2000 может содержать до 32768 объектов (комбинаций таблиц, форм, отчетов и т.д.), причем одновременно могут быть открыты 1024 таблицы (при наличии достаточного количества системных ресурсов). Можно импортировать таблицы из других систем управления базами данных (Paradox), баз данных типа клиент/сервер (таких как Microsoft SQR Server) и систем управления электронными таблицами (Microsoft Excel, Lotus 1-2-3 и др.).Более того, можно связать базу данных Access 2000 с таблицами баз данных других приложений (dBASE, FoxPro, Paradox), файлами электронных таблиц, форматированными текстовыми файлами и таблицами, хранящимися в другой базе данных Access 2000.

2. Запросы.

При помощи запросов можно произвести выборку данных, соответствующих некоторому критерию. Выборка данных может производится одновремен-

но из 16 таблиц. При создании запроса можно указать таблицы, в которых будет производится выборка данных, и указать те поля таблиц, которые должны быть включены в результат запроса, и указать условие отбора данных. В запрос можно включить до 255 полей (столбцов). Условие отбора определяет о, какие данные будут выбраны из таблицы в результате запроса.

3. Формы.

Формы позволяют отображать данные, содержащиеся в таблицах или запросах в более удобном для восприятия виде. При помощи форм можно добавлять в таблицы новые данные, а также изменять и удалять существующие. Форма может содержать рисунки, графики и другие внедренные объекты. Формы Access 2000 могут также содержать процедуры, написанные на языке Access VBA, для обеспечения обработки различных событий.

4. Отчеты.

Отчеты предназначены для печати данных, содержащихся в таблицах и запросах, в красиво оформленном виде. Access 2000 позволяет использовать различные форматы оформления отчетов и включать в отчеты иллюстрации, что делает вид отчета еще более привлекательным. Отчеты, как и формы, позволяют включать процедуры обработки событий, написанные на языке Access VBA.

5. Макросы.

Использование макросов в приложениях Access 2000 позволяет автоматизировать повторяющиеся операции. Чтобы создать макрос, нужно просто выбрать последовательность макрокоманд (из списка доступных), в том порядке, в котором они должны быть выполнены.

6. Модули.

Модули содержат Access VBA код, используемый для написания процедур обработки событий, таких как: нажатие на кнопку в форме или отчете, для создания функций настройки, используемых в формах, отчетах и запросах, для автоматического выполнения операций над объектами базы данных и программного управления операциями, которыми нельзя управлять с помощью макросов. Эти процедуры и функции можно использовать для сложных вычислений, которые не могут быть представлены последовательностью простых математических выражений, или вычислений, требующих принятия решений. Другими словами, добавление VBA кода дает возможность создавать полные базы данных с настраиваемыми меню, панелями инструментов и многими другими возможностями.

Единственный файл (.MDB) базы данных Access 2000 может содержать не только объекты данных(таблицы, индексы и запросы), но и объекты приложения(формы, отчеты, макросы и модули, написанные на языке Access VBA). Таким образом, можно создать завершенное приложение Access 2000, все объекты которого содержаться в одном .MDV файле. Однако многие разработчики приложений Access обычно создают два файла: в одном хранятся все объекты данных, а в другом – все объекты приложения.

Работа с панелями инструментов Access 2000

В Access 2000 команды “Вид, Панели инструментов” делают возможным выбрать текущую видимую панель инструментов. Эти команды показывают список тех панелей инструментов, которые доступны в текущем режиме работы.

Элемент “Настройка” в меню “Вид, Панели инструментов” открывает диалоговое окно “Настройка”, которое позволяет вам отображать панели инструментов или скрывать их. Чтобы показать или убрать панель инструмента щелкните по вкладке “Панели инструментов” и поставьте (снимите) флажок в окошке слева от имени. Для получения справки нажмите на кнопку “Помощник” в левом нижнем углу диалогового окна Настройка.

Диалоговое окно “Настройка” можно также использовать для изменения параметров отображения панелей инструментов. Щелкните по вкладке “Параметры”. Эта страница дает возможность выбрать различные параметры отображения панелей инструментов. Можно выбрать параметр “Крупные значки”, который вызывает изображение кнопок панели инструментов примерно в 2 раза крупнее, что делает их лучше для зрительного восприятия и проще для нажатия мышью.

Параметр “Отображать подсказки для кнопок” на панели инструментов управляет появлением всплывающих подсказок.

От установки опций “Включить в подсказки сочетания клавиш” зависит, показывает ли Access горячие клавиши (если они есть) как часть текста всплывающих подсказок.

Раскрывающийся список “Эффект при выводе меню” позволяет выбрать режим рисования меню. Можно выбрать “Нет” для отсутствия спецэффектов, “Случайный выбор” для случайного выбора анимационного эффекта при каждом новом открытии меню, “Развертывание” (меню разворачивается как веер), “Соскальзывание” (открывается, как катящаяся тень).

Можно добавить или удалить кнопки из панели инструментов:

1. Выберите вкладку <<Команды>> и в списке <<Категории>> выберите <<Элементы управления>>. В списке <<Команды>> появятся дополнительные кнопки.

2. Поместите указатель мыши на кнопку <<Диаграмма>>, нажмите левую кнопку мыши и не отпуская ее перетащите кнопку на панель инструментов.

Диалоговое окно <<Настройка>> имеет следующие дополнительные возможности:

1. Для удаления кнопок с панели инструментов надо открыть диалоговое окно <<Настройка>> и перетащить ненужные кнопки в любое место за панелью инструментов.

2. Для того чтобы вернуть панель инструментов к первоначальному виду по умолчанию, следует открыть вкладку <<Панели инструментов>>, выбрать в

списке нужную панель и нажать кнопку <<Сброс>>. Появится сообщение, запрашивающее подтверждения на отмену всех изменений.

3. Для создания кнопок, открывающих или запускающих на выполнение объект базы данных, выберите вкладку <<Команды>>, в списке <<Категории>> выберите, например, <<Все таблицы>>. Рядом в списке <<Команды>> появятся все таблицы текущей базы данных. Выберите название нужной таблицы и перетащите на панель инструментов. Проверьте работу кнопки.

4. Для изменения текста или значка на кнопках, щелкните правой кнопкой мыши по кнопке. В появившемся контекстном меню выберите команду <<Выбрать значок для кнопки>> – для смены значка на кнопке; выберите команду <<Изменить значок на кнопке>> –для редактирования изображения кнопки; выберите <<Имя>> –для ввода в текстовое поле нужной подписи для отображения ее на кнопке (при включенном пункте <<Значок и текст>>).

5. Чтобы создать новую пустую панель инструментов для настройки ее пользователем и добавления любого набора нужных кнопок, выберите вкладку <<Панели инструментов>> и в списке найдите элемент <<Служебная программа>> 1 или <<Служебная программа 2>> (для появления панели установите галочку слева).

6. Для создания пользовательской панели инструментов, которая станет частью базы данных, откройте вкладку <<Панели инструментов>> и в ней нажмите кнопку <<Создать>>. Появится диалоговое окно <<Создание панели инструментов>>, запрашивающее название новой панели. Access создаст плавающую панель инструментов, кнопки в которую добавляются из вкладки <<Команды>>.

7. Для удаления пользовательской панели инструментов во вкладке <<Панели инструментов>> выберите нужную панель и нажмите кнопку <<Удалить>>.

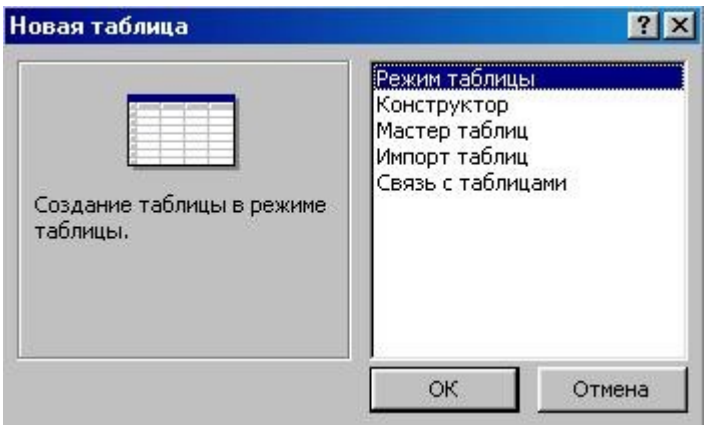
Создание новых таблиц при помощи Мастера таблиц.

Мастер таблиц позволяет создать новую таблицу на основе 25 образцов таблиц делового применения и 20 образцов таблиц личного применения.

Чтобы создать таблицу “Каталог видеозаписей” при помощи Мастера таблиц:

1. Откройте БД Борей. Сделайте активным окно базы данных.
Для этого закройте все окна объектов БД или нажмите кнопку <<Окно базы данных>> на панели инструментов.

2. Раскройте объект <<Таблицы>> окна базы данных и дважды щелкните на ярлыке <<Создание таблицы>> с помощью мастера (или нажмите кнопку “Создать” и в появившемся диалоговом окне (рис.2.1) выделите элемент “Мастер таблиц” и нажмите кнопку ОК).



3. В первом диалоговом окне выберите переключатель “Личное применение”. В списке “Образцы таблиц” выделите элемент “Каталог видеозаписей”. В списке “Образцы полей” появятся поля, которые могут быть включены в новую таблицу.

4. Чтобы включить все поля нажмите кнопку “>>”. При помощи кнопки “>” в список <<Поля новой таблицы>> можно добавить выделенный элемент списка “Образцы полей”. Кнопка”<” удаляет выделенный элемент списка “Поля новой таблицы”, а кнопка”<< “ удаляет все поля из этого списка. Нажмите кнопку “Далее”.

5. Появится второе диалоговое окно, в котором можно ввести имя новой таблицы и выбрать способ определения ключевого поля. Выберите переключатель “Пользователь определяет ключ самостоятельно”. Нажмите кнопку “Далее”.

6. Появится следующее диалоговое окно, в котором можно выбрать ключевое поле и указать, какие данные должны содержаться в ключевом поле.

Выбор поля “КодКаталогВидео” в качестве ключевого поля и автоматическое присвоение последовательных чисел каждой записи является в этом случае наиболее подходящим, поэтому выбранные параметры можно оставить без изменений. Нажмите кнопку “Далее”.

7. Следующее диалоговое окно появляется только в том случае, если в базе данных уже содержатся другие таблицы. Мастер таблиц позволяет связать новую таблицу с остальными. По умолчанию новая таблица не связывается ни с какой другой. В этом примере нет необходимости создавать связь с другими таблицами, поэтому все параметры этого диалогового окна можно оставить без изменения. Нажмите кнопку “Далее”.

8. Появится последнее диалоговое окно. Выберите переключатель “Изменить структуры таблицы” и нажмите кнопку “Готово”. Новая таблица будет открыта в режиме конструктора.

9. Чтобы просмотреть содержимое таблицы, нажмите на кнопке “Вид” или выберите команду “Вид, Режим таблицы”.

10. Нажмите кнопку “Закрыть”

Задание

1. Самостоятельно изучить контекстно-зависимую справку:

-показать справочную информацию о кнопке <<Запросы>>, о кнопке ”Вырезать”.

2. Разобраться со справочной системой Access 2000:

-показать работу вкладок <<Содержание>>, <<Предметный указатель>>, <<Поиск>>.

3. Разобраться с возможностями <<Помощника>>:

-способы вызова, параметры.

4. Создайте кнопки открытия таблиц <<Товары>> и <<Заказы>>. Переименуйте их.

5. Создайте пользовательскую панель инструментов.

6. Создайте **таблицу путем ввода данных**. В качестве примера наберите почтовые адреса 5 человек: Фамилия, Имя, адрес, город, индекс, страна. Таблицу назовите <<Адреса>>.

7. Создайте с помощью Мастера таблицу <<**Список рассылки**>> и свяжите ее с таблицей <<Клиенты>>, установив переключатель на: **Одной записи таблицы <<Клиенты>>будет соответствовать много записей таблицы <<Список рассылки>>**.

Контрольные вопросы

1. Назвать основные и дополнительные функции Access 2000
2. Перечислить характеристики приложения Access 2000
3. Перечислить и дать характеристику каждого элемента базы данных Access 2000
4. Перечислить способы создания новой базы данных.

Лабораторная работа 2

Цель лабораторной работы заключается в том, чтобы научится:

- создавать таблицы в режиме конструктора.
- определять значения по умолчанию для полей.
- добавлять условия на значение полей.
- добавлять условия на значение записей.

Создание таблицы в режиме конструктора.

1. Нажмите на объекте “Таблицы”, а затем нажмите кнопку “Создать”. Выберите в списке элемент “Конструктор” и нажмите ОК или откройте ярлык <<**Создание таблицы в режиме конструктора>>**. Новая пустая таблица будет открыта в режиме конструктора.

2. Каждое поле таблицы имеет набор свойств. Первые три свойства полей выводятся в бланке структуры таблицы, который расположен6 в верхней части окна таблицы, открытой в режиме конструктора. Значения оставшихся свойств полей вводятся в нижней части окна конструктора таблицы на вкладках <<Общие>> и <<Подстановка>>. О свойствах таблиц и полей смотрите в приложе-

нии 1. более подробно о свойствах можно узнать, установив текстовый курсор в это поле и нажав на клавишу <F1>. Введите данные из таблицы 2.1.

Таблица 2.1

Имя_ поля	Тип данных	Описание
Код _сотрудника	Числовой	Код сотрудника, с которым связано мероприятие. Значение присваивается на основе значения поля <<Код сотрудника>> таблицы <<Сотрудники>>
Тип мероприятия	Текстовый	Сокращение от названия мероприятия. <<Д>>-анализ деятельности, <<М>>-мероприятия по продвижению, <<Р>>-рекламная компания, <<С>>-изучение запроса, <<А>>-анкетирование, <<П>>-прогнозирование, <<К>>-анализ конкурентов.
Код_ ответственного	Числовой	Код сотрудника, инициировавшего или рекомендовавшего проведение мероприятия.
Дата _план	Дата/Время	Запланированная дата проведения мероприятия.
Код _начальника	Числовой	Код сотрудника, утвердившего проведение мероприятия.
Дата _факт	Дата/Время	Дата проведения мероприятия. Поле остается незаполненным, если мероприятие не проведено.
Рейтинг	Числовой	Рейтинг сотрудника по 10-бальной шкале. Незаполненное поле означает отсутствие рейтинга
Сумма	Денежный	Предполагаемый рост дохода от проведенного мероприятия.
Примечание	МЕМО	Полное описание мероприятия. В этом поле ответственный за проведение мероприятия и начальник могут оставлять свои замечания.

3. Заполните свойства полей в соответствии с таблицей 2.2. О выборе типа размера и формата данных для полей смотрите в приложении 1.

Таблица 2.2.

Имя _поля	подпись	Размер поля формат поля	Формат поля
Код _сотрудника	Сотрудник	Длинное целое	Основной
Тип _мероприятия	Мероприятие	1	@>
Код _ответственного	Ответственный	Длинное целое	Основной
Дата_план	Дата-план	-	Краткий формат даты
Код _начальника	Начальник	Длинное целое	Основной
Дата _факт	Дата-факт	-	Краткий формат даты
рейтинг	рейтинг	Целое	Основной

Сумма	Сумма	-	######0,00р

6. Переключитесь в режимы таблицы для просмотра результата выполненной работы. Появится диалоговое окно, запрашивающее подтверждение сохранения структуры таблицы. Нажмите кнопку ДА. Появится диалоговое окно “Сохранение”. Введите имя таблицы: **”Задания”** и нажмите кнопку **ОК** или нажмите клавишу <Enter>. После сохранения структуры таблицы появится окно, сообщающее о том, что в таблице не задано ключевое поле и предлагающее создать его. Сейчас откажитесь от определения ключевых полей, нажав кнопку **НЕТ**.

Таблица ”Задания” откроется в режиме таблицы. Пока в ней нет данных. Та запись, что появляется в режиме таблицы, на самом деле в ней не хранится. Она используется для добавления новых записей и имеет специальное название- хвостовая запись. Поскольку пока определены не все свойства полей, не добавляйте новые записи в таблицу.

Определение значений по умолчанию для полей.

Обратите внимание на то, что в единственной записи таблицы <<Задания>> числовым полям присвоены нулевые значения. При создании таблицы, полями с типами данных <<Числовой>> и <<Денежный>> автоматически присваивается значение 0, а поля других типов по умолчанию остаются пустыми. Иногда нулевое значение для числовых полей недопустимо. В этом случае значение по умолчанию должно быть изменено. Чтобы увеличить производительность работы при вводе данных в таблицу, необходимо задать значения по умолчанию для ее полей.

1. В таблице 2.3 приведены значения по умолчанию для полей таблицы <<Задания>>, введите их.

Таблица 2.3- Значения по умолчанию для полей таблицы

Поле	Значение по умолчанию	Пояснение
Код_сотрудника		Число 0 не может быть кодом сотрудника, поэтому значение по умолчанию должно быть удалено.
Тип_мероприятия	с	Изучение спроса выполняется чаще всего.
Код_ответственного		Число 0 не может быть кодом ответственного.
Дата_план	Date()	Выражение возвращает текущую дату (см. прил. 4)
Код_начальника		Число 0 не может быть кодом начальника
Дата_факт	Date()+28	Выражение возвращает текущую дату плюс 28 дней.

Рейтинг		Многие мероприятия влияют на рейтинг сотрудника.
Сумма		Поле <<Сумма>> не должно содержать нулевого значения.
Примечание		Изменений не требуется. Для текстовых полей b полей типа MEMO Access 2000 автоматически устанавливает значение по умолчанию “пустая строка”.

Если в свойстве “Значение по умолчанию” ничего не введено, то полю присваивается значение по умолчанию “Пустое значение”(Null)(см. прил.1).

Перед выражением, задающим значение по умолчанию для поля, должен стоять знак неравенства, например: =Date().

2. После того как значения по умолчанию присвоены, возвратитесь в режим таблицы. Появится диалоговое окно, запрашивающее подтверждения изменения структуры таблицы. Нажмите кнопку **ДА**.

Добавление условий на значение полей.

Условия на значения проверяют корректность данных только в одном поле, независимо от значений других полей.

1. Перейдите в режим конструктора.

2. Добавьте в структуру таблицы Задания условия на значение, приведенные в таблице 2.4 (об операторах смотрите приложение 3).

Таблица 2.4

Поле	Условие на значение	Текст сообщения об ошибке
Код_сотрудника	>0	Введите правильный код сотрудника
Тип_мероприятия	“П” Or ”Д” Or ”К” Or ”Р” Or ”М” Or ”С” Or ”А”	Недопустимый код. Допустимы только коды<<П>>,<<Д>>,<<М>>,<<К>>,<<Р>>,<<С>>и<<А>>
Код_ответственного	>0	Введите правильный код сотрудника, ответственного за мероприятие.
Дата_план	Between Date()-365 And Date()+365	Мероприятие не может быть запланировано более чем на один год вперед.
Код_начальника	>0 Or Is Null	Введите правильный код начальника или оставьте поле пустым.
Рейтинг	Between 0	Введите правильный рейтинг сотрудника

	And 10 Or Is Null	(число от 0 до 10) или оставьте поле пустым

3. Установите для свойства “**Обязательное поле**” полей ”Код_Сотрудника”, ”Тип_Мероприятия”, ”Код_Ответственного” и ”Дата_План” значение ”**Да**”.

4. Сохраните структуру таблицы.

Добавление условий на значение записей.

При нормальной организации работы фирмы значение поля “Дата_Факт” не должно превышать значение поля “Дата_План’. Можно задать условие на значение записи таблицы. В таких условиях могут сравниваться одновременно значения нескольких полей. Само условие и текст сообщения об ошибке, появляющийся при его нарушении, можно задать в диалоговом окне “Свойства таблицы”.

Чтобы определить условие на значение записи для поля “Дата_Факт”:

1. Откройте таблицу “Задания” в режиме конструктора и нажмите кнопку ”Свойства” на панели инструментов. Появится диалоговое окно ”Свойства таблицы” (о свойствах таблицы смотрите приложение 1).

2. В поле ввода “Описание” введите: **Персональные мероприятия по отделу.**

3. Поместите текстовый курсор в поле “Условие на значение” и нажмите кнопку “…” справа от него. Появится диалоговое окно “Построитель выражений” (рис. 2.2). В левом списке этого диалогового окна выделена таблица “Задания”, а в центральном списке выводится список ее полей.

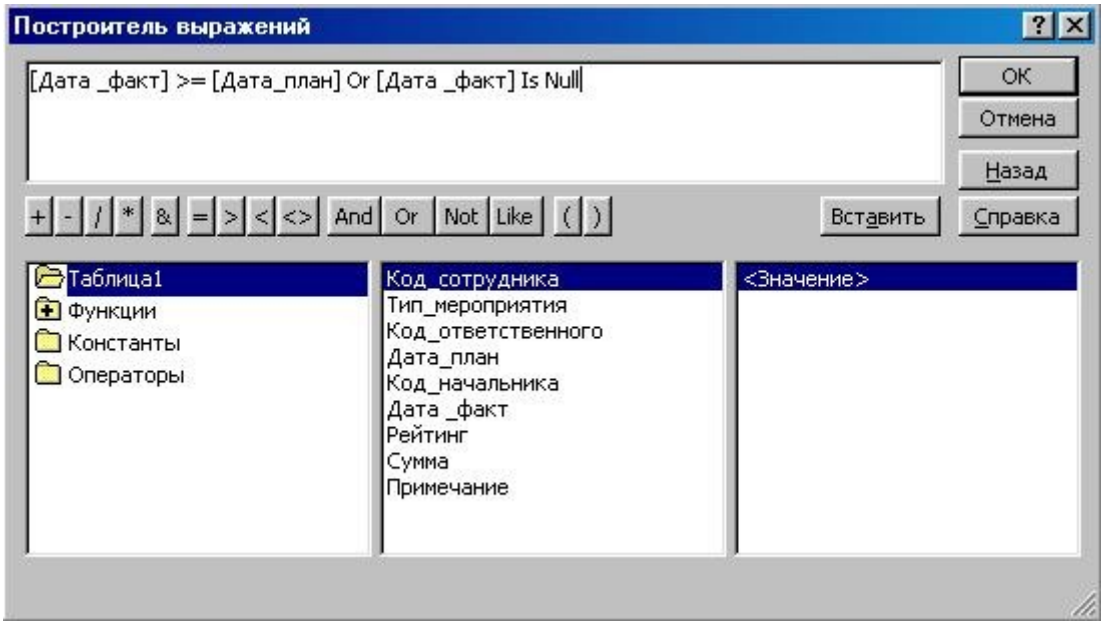


Рис. 2.2

4. Дважды щелкните левой кнопкой мыши по элементу “Дата_Факт” в списке полей таблицы. В окне выражения появится строка **[Дата_Факт].**

5. Введите: >= в окне выражения и дважды щелкните левой кнопкой мыши по элементу **“Дата_План”** в списке полей таблицы, чтобы добавить его в выражение.

6. При создании таблицы “Задания” в поле “Дата_Факт” планировалось хранить пустые значения для мероприятий, которые еще не произошли. Поэтому добавьте к выражению, задающему условие на значение, строку **Or [Дата_Факт] Is Null** (рис. 2.2).

7. Чтобы закрыть окно “Построитель выражений” и добавить условие на значение записей таблицы, нажмите кнопку ОК.

8. Поместите текстовый курсор в поле <<Сообщение об ошибке>> в диалоговом окне <<Свойства таблицы>> и введите текст сообщения: **Дата проведения мероприятия не может предшествовать запланированной дате мероприятия.**

9. Закройте окно “Свойства таблицы”, нажав кнопку “Закрыть” или нажав кнопку “Свойства” на панели инструментов.

Примечание.

В таблице может быть задано более одного условия на значение записей. Эти условия нужно соединить операторами And и поместить в свойство “Условие на значение” таблицы. Однако при нарушении любого из условий на значение записей будет выводится один и тот же текст сообщения об ошибке.

10. Сохраните структуру таблицы.

Удаление таблиц.

Если вы хотите удалить таблицу из базы данных, нажмите кнопку “Окно базы данных”, вкладку “Таблица” (если список таблиц не открыт), и щелкните по имени таблицы, чтобы выделить ее. Нажмите <Delete> и затем ОК, когда Access в окне сообщения запросит подтверждение удаления . Перед удалением таблица должна быть закрыта.

Помните о том , что удаленную таблицу нельзя восстановить, выбрав команду “Правка, Отменить”.

Копирование таблиц.

Чтобы скопировать таблицу, база данных, в которую происходит копирование, должна быть создана.

1. Установите курсор на таблицу, которую вы хотите копировать.

2. Выберите команду “Правка, Копировать” или нажмите кнопку “Копировать” на панели инструментов.

3. Откройте БД куда будет копироваться таблица, раскройте вкладку “Таблицы” и выберите команду “Правка, Вставить” или нажмите кнопку “Вставить” на панели инструментов. Появится диалоговое окно “Вставка таблицы”.

В диалоговом окне “Вставка” таблицы можно ввести имя таблицы и выбрать параметры вставки таблицы. Чтобы создать новую таблицу и скопировать в нее данные из таблицы-источника или заменить данные в таблице, имя которой указано в поле “Имя таблицы”, на данные, содержащиеся в Буфере обмена, выберете переключатель “Структура и данные”. Чтобы добавить данные в таблицу, имя которой указано в поле ввода “Имя таблицы”, выберите переключатель “Добавление данных в таблицу”.

Задание

1. Показать результаты по лабораторной работе (таблицу <<Задание>>).
2. В таблице “Заказы” установить, чтобы значение поля “ДатаИсполнения” было меньше значения поля “ДатаНазначения” и выводилось сообщение об ошибке.
3. В таблице “Сотрудники” для поля “ДатаНайма” установить, чтобы дата выглядела следующим образом: Август 4, 1998-4:33
4. В таблице “Сотрудники” для поля “Домашний телефон” создайте маску ввода (например, для ввода номера телефона 567-77-22).
5. Скопируйте из базы данных Борей таблицы” Товары” и “Поставщики” и поместите их в свою базу данных под другими именами.
6. Удалите скопированные таблицы в присутствии преподавателя.

Контрольные вопросы

1. Перечислить типы данных.
2. Описать типы данных.
3. Описать свойства таблиц.
4. Описать свойства полей.

Тема 4 - 9. Создание Web-страницы.

HTML начало

HyperText Markup Language (HTML) является стандартным языком, предназначенным для создания гипертекстовых документов в среде WEB. HTML-документы могут просматриваться различными типами WEB-броузеров. Когда документ создан с использованием HTML, WEB-броузер может интерпретировать HTML для выделения различных элементов документа и первичной их обработки. Использование HTML позволяет форматировать документы для их представления с использованием шрифтов, линий и других графических элементов на любой системе, их просматривающей.

Большинство документов имеют стандартные элементы, такие, как заголовок, параграфы или списки. Используя **тэги** HTML вы можете обозначать данные элементы, обеспечивая WEB-броузеры минимальной информацией для отображения данных элементов, сохраняя вцелом общую структуру и информационную полноту документов. Все что необходимо, чтобы прочитать HTML-доку-

мент - это WEB-броузер, который интерпретирует тэги HTML и воспроизводит на экране документ в виде, который ему придает автор.
В большинстве случаев автор документа строго определяет внешний вид документа. В случае HTML читатель (основываясь на возможностях WEB-броузера может, в определенной степени, управлять внешним видом документа (но не его содержимым). HTML позволяет отметить, где в документе должен быть заголовок или абзац при помощи тэга HTML, а затем предоставляет WEB-броузеру интерпретировать эти тэги. Например, один WEB-броузер может распознавать тэг начала абзаца и представлять документ в нужном виде, а другой не имеет такой возможности и представляет документ в одну строку. Пользователи некоторых WEB-броузеров имеют, также, возможность настраивать размер и вид шрифта, цвет и другие параметры, влияющие на отображение документа.
HTML-тэги могут быть условно разделены на две категории:

- тэги, определяющие, как будет отображаться WEB-броузером тело документа вцелом
- тэги, описывающие общие свойства документа, такие как заголовок или автор документа

Запомните, что основное преимущество HTML заключается в том, что ваш документ может быть просмотрен на WEB-броузерах различных типов и на различных платформах.

Как создаются HTML документы?

HTML-документы могут быть созданы при помощи любого текстового редактора или специализированных HTML-редакторов и конвертеров. Выбор редактора, который будет использоваться для создания HTML-документов, зависит исключительно от понятия удобства и личных пристрастий каждого автора. Например, HTML редакторы, такие, как "Netscape Navigator Gold" компании Netscape позволяют
создавать документы графически с использованием технологии WYSIWYG (What You See Is What You Get). С другой стороны, большинство традиционных средств для создания документов имеют конвертеры, позволяющие преобразовывать документы к формату HTML.

Основные положения

Все тэги HTML начинаются с "<" (левой угловой скобки) и заканчиваются символом ">" (правой угловой скобки). Как правило, существует стартовый тэг и завершающий тэг. Для примера приведем тэги заголовка, определяющие текст, находящийся внутри стартового и завершающего тэга и описывающий заголовок документа:
<TITLE> Заголовок документа </TITLE>
Завершающий тэг выглядит также, как стартовый, и отличается от него прямым слэшем перед текстом внутри угловых скобок. В данном примере тэг <TITLE> говорит WEB-броузеру об использовании формата заголовка, а тэг </TITLE> - о завершении текста заголовка.
Некоторые тэги, такие, как <P> (тэг, определяющий абзац), не требуют завер-

шающего тэга, но его использование придает исходному тексту документа улучшенную читаемость и структурируемость.
HTML не реагирует на регистр символов, описывающих тэг, и приведенный ранее пример может выглядеть следующим образом:
<title> Заголовок документа </title>

> **Внимание!** Дополнительные пробелы, символы табуляции и возврата каретки, добавленные в исходный текст HTML-документа для его лучшей читаемости, будут проигнорированы WEB-броузером при интерпретации документа. HTML-документ может включать вышеописанные элементы только если они помещены внутрь тэгов <PRE> и </PRE>. Более подробно о тэгах <PRE> будет написано ниже.

Структура документа

Когда WEB-броузер получает документ, он определяет, как документ должен быть интерпретирован. Самый первый тэг, который встречается в документе, должен быть тэгом <HTML>. Данный тэг сообщает WEB-броузеру, что ваш документ написан с использованием HTML. Минимальный HTML-документ будет выглядеть так:
<HTML> ...тело документа... </HTML>
Заголовочная часть документа <HEAD>
Тэг заголовочной части документа должен быть использован сразу после тэга <HTML> и более нигде в теле документа. Данный тэг представляет из себя общее описание документа. Избегайте размещать какой-либо текст внутри тэга <HEAD>. Стартовый тэг <HEAD> помещается непосредственно перед тэгом <TITLE> и другими тэгами, описывающими документ, а завершающий тэг </HEAD> размещается сразу после окончания описания документа. Например:
<HTML>
<HEAD>
<TITLE> Список сотрудников </TITLE>
</HEAD>
...

> **Внимание!** Технически, стартовые и завершающие тэги типа <HTML>, <HEAD> и <BODY> необязательны. Но настоятельно рекомендуется их использовать, поскольку использование данных тэгов позволяет WEB-броузеру уверенно разделить заголовочную часть документа и непосредственно смысловую часть.

Заголовок документа <TITLE>
Большинство WEB-броузеров отображают содержимое тэга <TITLE> в заголовке окна, содержащего документ и в файле закладок, если он поддерживается WEB-броузером. Заголовок, ограниченный тэгами <TITLE> и </TITLE>, размещается внутри <HEAD>-тэгов, как показано выше на примере. Заголовок документа не появляется при отображении самого документа в окне.

Комментарии
Как любой язык, HTML позволяет вставлять в тело документа комментарии, которые сохраняются при передаче документа по сети, но не отображаются

броузером. Синтаксис комментария:
<!-- Это комментарий -->
Комментарии могут встречаться в документе где угодно и в любом количестве.

Тэги тела документа

Тэги тела документа идентифицируют отображаемые в окне компоненты HTML-документа. Тело документа может содержать ссылки на другие документы, текст и другую форматированную информацию.

Тело документа <BODY>
Тело документа должно находиться между тэгами <BODY> и </BODY>. Это та часть документа, которая отображается как текстовая и графическая (смысловая) информация вашего документа.
Уровни заголовков <Hx>
Когда пишется HTML-документ, текст структурно делится на просто текст, заголовки частей текста, заголовки более высокого уровня и т.д. Первый уровень заголовков (самый большой) обозначается цифрой 1, следующий - 2, и т.д. Большинство броузеров поддерживает интерпретацию шести уровней заголовков, определяя каждому из них собственный стиль. Заголовки выше шестого уровня не являются стандартом и могут не поддерживаться броузером. Заголовок самого верхнего уровня имеет признак "1". Синтакс заголовка уровня 1 следующий:
<H1> Заголовок первого уровня </H1>
Заголовки другого уровня могут быть представлены в общем случае так:
<Hx> Заголовок x-го уровня </Hx>
где x - цифра от 1 до 6, определяющая уровень заголовка.
Тэг абзаца <P>
В отличии от большинства текстовых процессоров, в HTML-документе обычно игнорируются символы возврата каретки. Физический разрыв абзаца может находиться в любом месте исходного текста документа (для удобства его читаемости). Однако броузер разделяет абзацы только при наличии тэга <P>. Если вы не разделите абзацы тэгом <P>, ваш документ будет выглядеть как один большой абзац.
Дополнительные параметры тэга <P>:
<P **ALIGN=left|center|right**>
позволяют выравнивать абзац по левому краю, центру и правому краю соответственно.

Центрирование элементов документа
Вы можете центрировать все элементы документа в окне броузера. Для этого можно использовать тэг <CENTER>.
Все элементы между тэгами <CENTER> и </CENTER>
будут находиться в центре окна

Тэг преформатирования <PRE>

Тэг преформатирования, <PRE>, позволяет представлять текст со специфическим форматированием на экране. Предварительно сформатированный текст заканчивается завершающим тэгом </PRE>. Внутри предварительно сформатированного текста разрешается использовать:

- перевод строки
- символы табуляции (сдвиг на 8 символов вправо)
- непропорциональный шрифт, устанавливаемый броузером

Использование тэгов, определяющих формат абзаца, таких как <Hx> или <ADDRESS>, будет игнорироваться броузером при помещении их между тэгами <PRE> и </PRE>.

Далее идет несколько более подробный пример, собранный из предыдущих:

```
<HTML>
<HEAD>
<TITLE> Список сотрудников </TITLE>
</HEAD>
<BODY>
<H2> Список сотрудников нашей фирмы </H2>
<H3> Составлено : 30 июля 1996 года </H3>
Данный список содержит фамилии, имена и отчества
всех сотрудников нашей компании. <P>
Список может быть использован только в служебных целях. <P>
</BODY>
</HTML>
```

Вот, что вы увидите на экране броузера:

Список сотрудников нашей фирмы

Составлено : 30 июля 1996 года

Данный список содержит фамилии, имена и отчества всех сотрудников нашей компании.

Список может быть использован только в служебных целях.

> **Внимание!** Заголовок "Список сотрудников" не отображен броузером как часть документа. Он появится в заголовке окна броузера.

Разрыв строки

Тэг
 извещает броузер о разрыве строки. Наилучший пример использования данного тэга - форматированный адрес или любая другая последовательность строк, где броузер должен отображать их одну под другой. Например:

```
Алексей Ярцев <BR>
Дмитровское шоссе, <BR>
д.9Б, офис 326 <BR>
```

Дополнительный параметр позволяет расширить вохможности тэга
.

<BR **CLEAR=left|right|all**>

Данный параметр позволяет выполнить не просто перевод строки, а разместить следующую строку, начиная с чистой левой (left), правой (right) или обоих (all) границ окна броузера. Например, если рядом с текстом слева встречается рисунок, то можно использовать тэг
 для смещения текста ниже рисунка:

```
<p> Как вы можете видеть, данная схема демонстрирует связь<BR
```

CLEAR=left>
<img src="http://www.softexpress.com/images/schema1.gif" align=baseline>
Мастер/Деталь </p>

Неразрывная строка <NOBR>
Если вы не хотите, чтобы броузер автоматически переносил строку, то вы можете обозначить ее тэгами <NOBR> и </NOBR>. В этом случае броузер не будет переносить строку даже если она выходит за нраницы экрана; вместо этого броузер позволит горизонтально прокручивать окно. Например:
<NOBR> Данная строка является самой длинной строкой документа, которая не допускает какого-либо разбиения где бы то ни было </NOBR>
Если же вы хотите все же позволить разбиение данной строки на две, но в строго запланированном месте, то вставьте тэг <WBR> в это место. Например:
<NOBR> Данная строка является самой длинной строкой документа,<WBR>
которая не допускает какого-либо разбиения где бы то ни было </NOBR>
Данная строка является самой длинной строкой документа,
которая не допускает какого-либо разбиения где бы то ни было

Цитата <BLOCKQUOTE>
Данный тэг предназначен для обозначения в документе цитаты из другого источника. Текст, обозначенный тэгом <BLOCKQUOTE>, отступает от левого края документа на 8 пробелов. Например:
На открытии данной конференции глава представительства произнес: <P>
<BLOCKQUOTE>
Сегодня один из величайших дней для нашей компании.
Мы открыли новую технологию, позволяющую нашим клиентам повысить производительность их настольных систем в несколько раз.
</BLOCKQUOTE>
При отображении броузером данный текст будет выглядеть так:
На открытии данной конференции глава представительства произнес:
Сегодня один из величайших дней для нашей компании.
Мы открыли новую технологию, позволяющую нашим клиентам повысить производительность их настольных систем в несколько раз.

Список базовых тэгов HTML

Стартовый	Завершающий	Описание
<HTML>	</HTML>	Обозначение HTML-документа
<HEAD>	</HEAD>	Заголовочная часть документа
<TITLE>	</TITLE>	Заголовок документа
<BODY>	</BODY>	Тело документа
<H1>	</H1>	Заголовок абзаца первого уровня
<H2>	</H2>	Заголовок абзаца второго уровня
<H3>	</H3>	Заголовок абзаца третьего уровня
<H4>	</H4>	Заголовок абзаца четвертого уровня
<H5>	</H5>	Заголовок абзаца пятого уровня
<H6>	</H6>	Заголовок абзаца шестого уровня
<P>	</P>	Абзац
<PRE>	</PRE>	Форматированный текст
 		Перевод строки без конца абзаца
<BLOCKQUOTE>	</BLOCKQUOTE>	Цитата

Описанные ранее тэги - это все, что необходимо вам для того, чтобы начать работать с HTML.

С использованием описанных тэгов вы можете создать простой HTML-документ. Однако, следующие разделы позволят вам существенно улучшить внешний вид ваших документов и опишут новые возможности HTML.

Тэги списков

Существует три основных вида списков в HTML-документе:

- **пронуменрованный**
- **непронуменрованный**
- **список описаний**

Вы можете создавать вложенные списки, используя различные тэги списков или повторяя одни внутри других. Для этого просто необходимо разместить одну пару тэгов (стартовый и завершающий) внутри другой. Будут ли элементы вложенного списка иметь те же маркеры, обозначающие элемент списка - зависит от броузера. Более подробно смотри в разделе "Вложенные списки".

Пронумерованные списки

В пронумерованном списке броузер автоматически вставляет номера элементов по порядку. Это означает, что если вы удалите один или несколько элементов пронумерованного списка, то остальные номера автоматически будут пересчитаны.

Пронумерованный список начинается стартовым тэгом <OL> и завершается тэгом </OL>. Каждый элемент списка начинается с тэга <LI>. Например:

<OL>
<LI> Программирование
<LI> Алгоритмизация

<LI> Проектирование
</OL>

1. Программирование
2. Алгоритмизация
3. Проектирование

Тэг <OL> может иметь параметры:
<**OL** TYPE=A|a|I|i|1 START=n>
где:
TYPE

Вид счетчика:

- A - большие латинские буквы (A,B,C...)
- a - маленькие латинские буквы (a,b,c...)
- I - большие римские цифры (I,II,III...)
- i - маленькие римские цифры (i,ii,iii...)
- 1 - обычные цифры (1,2,3...)

START=n

Число, с которого начинается отсчет

Например:
<OL TYPE=I START=15>
<LI> Программирование
<LI> Алгоритмизация
<LI> Проектирование
</OL>

XV. Программирование
XVI. Алгоритмизация
XVII. Проектирование

Непронумерованные списки.
Для непронумерованных списков броузер обычно использует маркеры для пометки элемента списка. Вид маркера, как правило, настраивает пользователь броузера.
Пронумерованный список начинается стартовым тэгом <UL> и завершается тэгом </UL>. Каждый элемент списка начинается с тэга <LI>. Например:
<UL>
<LI> Программирование
<LI> Алгоритмизация
<LI> Проектирование
</UL>

- Программирование
- Алгоритмизация
- Проектирование

Тэг <UL> может иметь параметр:
<**UL** TYPE=disc|circle|square>
Тип тэга <UL> определяет внешний вид маркера как вид по умолчанию (disc), круглый (circle) или квадратный (square). Например:

<UL TYPE=square>
<LI> Программирование
<LI> Алгоритмизация
<LI> Проектирование
</UL>

- Программирование
- Алгоритмизация
- Проектирование

Вложенные списки
Дадим пример вложенных списков:

```
<HTML>
<HEAD>
<TITLE> Список сотрудников </TITLE>
</HEAD>
<BODY>
<H2> Список сотрудников нашей фирмы </H2>
<H3> Составлено : 30 июля 1996 года </H3>
Данный список содержит фамилии, имена и отчества всех сотрудников нашей компании. <P>
Список может быть использован только в служебных целях. <P>
<OL>
<LI> Дирекция
<UL>
<LI> Иванов И.И.
<LI> Петров К.В.
</UL>
<LI> Отдел маркетинга
<UL>
<LI> Варшавская Е.Л.
<LI> Самсонов Д.М.
</UL>
</OL>
</BODY>
</HTML>
```

Вот, что вы увидите на экране броузера:

Список сотрудников нашей фирмы

Составлено : 30 июля 1996 года

Данный список содержит фамилии, имена и отчества всех сотрудников нашей компании.

Список может быть использован только в служебных целях.

1. Дирекция
    - Иванов И.И.
    - Петров К.В.
2. Отдел маркетинга
    - Варшавская Е.Л.
    - Самсонов Д.М.

Элемент списка <LI>

Тэг <LI> может иметь параметры:

<**OL** TYPE=disc|circle|squade> или <**OL** TYPE=A|a|I|i|1 VALUE=n>

в зависимости от того, в списке какого вида находится данный элемент.

TYPE

Вид маркера (см. <UL>) или счетчика (см.OL)

VALUE=n

Значение для элемента пронумерованного списка (его номер). Все дальнейшие номера элементов списка будут отсчитываться от этого номера.

Например:

```
<OL TYPE=I START=15>
<LI> Программирование
<LI TYPE=i VALUE=25> Алгоритмизация
<LI> Проектирование
</OL>
```

XV. Программирование
XVI. Алгоритмизация
XVII. Проектирование

Список определений

Список определений начинается с тэга <DL> и завершается тэгом </DL>. Данный список служит для создание списков типа "термин"-"описание". Каждый термин начинается тэгом <DT> , а описание - тэгом <DD>. Например:

<DL>
<DT> <B> Отдел маркетинга </B>
<DD> Данный отдел занимается продвижением продуктов и услуг
<DT> <B> Финансовый отдел </B>
<DD> Данный отдел занимается всеми финансовыми операциями
<DT> <B> Отдел кадров </B>
<DD> Данный отдел занимается учетом и набором новых сотрудников фирмы, распределением отпусков, отслеживанием больничных листов и т.д.
</DL>

Отдел маркетинга
Данный отдел занимается продвижением продуктов и услуг
Финансовый отдел
Данный отдел занимается всеми финансовыми операциями
Отдел кадров
Данный отдел занимается учетом и набором новых сотрудников фирмы, распределением отпусков, отслеживанием больничных листов и т.д.

Гипертекстовые ссылки

Гипертекстовые ссылки являются ключевым компонентом, делающим WEB привлекательным для пользователей. Добавляя гипертекстовые ссылки (далее - ссылки), вы делаете набор документов связанным и структурированным, что позволяет пользователю получать необходимую ему информацию максимально быстро и удобно.

Ссылки имеют стандартный формат, что позволяет броузеру интерпретировать их и выполнять необходимые функции (вызывать методы) в зависимости от типа ссылки. Ссылки могут указывать на другой документ, специальное место данного документа или выполнять другие функции, например запрашивать файл по FTP-протоколу для отображения его броузером. URL может указывать на специальное место по абсолютному пути доступа, или указывать на документ в текущем пути доступа, что часто используется при организации больших структурированных WEB-сайтов.

> **Внимание!** Вы можете использовать ссылки как для перемещения по документу, так и для перемещения от одного документа к другому. Однако, HTML не поддерживает возврат на предыдущую ссылку, если перемещение происходило внутри документа. Если вы используете ссылки внутри документа, а затем нажимаете на клавишу Back, то вы не перейдете на предыдущую ссылку, а вернетесь на ту часть документа, которую вы просматривали до этого.

URL

HTML использует URL (Uniform Resource Locator) для представления гипертекстовых ссылок и ссылок на сетевые сервисы внутри HTML-документа. Первая часть URL (до двоеточия) описывает метод доступа или сетевой сервис. Другая часть URL (после двоеточия) интерпретируется в зависимости от метода доступа. Обычно, два прямых слэша после двоеточия обозначают имя машины:

method://**machine-name**/path/foo.html

Следующий пример представляет собой вызов HTML-документа index.html с сервера www.softexpress.com с использованием HTTP протокола:

http://www.softexpress.com/index.html

Uniform Resource Locator имеет следующий формат:

method://servername:port/pathname#anchor

Опишем каждый из компонентов URL:

METHOD

Имя операции, которая будет выполняться при интерпретации данного URL. Наиболее часто используемые методы:

file:

чтение файла с локального диска. Имя файла интерпретируется для локальной машины пользователя. Данный метод используется для отображения какого-либо файла, находящегося на машине пользователя. Например:
file:/home/alex/index.html - отобажает файл index.html из каталога /home/alex на пользовательской машине

http:

доступ к WEB-странице в сети с использованием HTTP-протокола. (Это наиболее часто используемый метод доступа к какому-либо HTML-документу в сети). Например:
http://www.softexpress.com/ - доступ к Home-странице компании SoftExpress

ftp:

запрос файла с анонимного FTP-сервера. Например:
ftp://hostname/directory/filename

mailto:

активизирует почтовую сессию с указанным пользователем и хостом. Например:
mailto:info@softexpress.com - активизирует сессию посылки сообщения пользователю info на машине softexpress.com, если броузер поддерживает запуск электронной почты. Заметьте, что метод mailto: не требует указание слэшей после двоеточия (как правило, после двоеточия сразу идет электронный адрес абонента)

telnet:

обращение к службе telnet

news:

вызов службы новостей, если броузер ее поддерживает. Например: news:relcom.www.support

SERVERNAME

Необязательный параметр, описывающий полное сетевое имя машины. Например:

www.softexpress.com - полное сетевое имя сервера фирмы СофтСервис.

Если имя сервера не указано, то ссылка считается локальной, и полный путь, указанный далее в URL вычисляется на той машине, с которой взят HTML-документ, содержащий данную ссылку. Вместо символьного имени машины может быть использован IP-адрес, однако это не рекомендуется из-за возможного пересечения с фиксированными локальными адресами внутренней сети.

PORT

Номер порта TCP на котором функционирует WEB-сервер. Если порт не указан, то "по умолчанию" используется порт 80. Данный параметр (port) не используется в подавляющем большинстве URL.

PATHNAME

Частичный или полный путь к документу, который должен вызваться в результате интерпретации URL. Различные WEB-сервера сконфигурированы по разному для интерпретации пути доступа к документу. Например, при использовании CGI скриптов (исполняемых программ), они обычно собираются в одном или нескольких выделенных каталогах, путь

к которым записан в специальных параметрах WEB-сервера. Для данных каталогов WEB-сервером выделяется специальный логический путь, который и используется в URL. Если WEB-сервер видит данный путь, то запрашиваемый файл интерпретируется как исполняемый модуль. В противном случае, запрашиваемый файл интерпретируется просто как файл данных, даже если он является исполняемым модулем. Например:

http://www.softexpress.com/cgi-win/handle.exe

В данном примере HTTP-сервер должен вызвать CGI-скрипт с именем handle.exe, который находится на машине с сетевым именем www.softexpress.com. Путь к данному скрипту - /cgi-win/ - в действительности является виртуальным путем (выделенным сервером для исполняемых модулей). Заметьте, что при описании пути используется UNIX-подобный синтаксис, где, в отличии от DOS и Windows используются прямые слэши вместо обратных. Если после сетевого имени машины сразу идет имя документа, то он должен находиться в корневом каталоге на удаленной машине или (что чаще) в каталоге, выделенном WEB-сервером в качестве корневого. Если же URL закагчивается сетевым именем машины, то в качестве документа запрашивается документ из корневого каталога удаленной машины с именем, установленным в настройках WEB-сервера (как правило, это index.html).

#ANCHOR

Данный элемент является ссылкой на строку (точку) внутри HTML-документа. Большинство броузеров, встречая после имени документа данный элемент, размещают документ на экране таким образом, что указанная строка документа помещается в верхнюю строку рабочего окна броузера. Точки, на которые ссылается #anchor, указываются в документе при помощи тэга NAME, как это будет описано далее.

Структура ссылок в HTML-документе

Пока что мы рассмотрели только внешний вид URL. Для того, чтобы броузер отобразил ссылку на URL, необходимо отчетить URL специальными тэгами в HTML-документе. Синтаксис HTML, позволяющий жто сделать - следующий:

<A HREF="*URL*"> текст-который-будет-подсвечен-как-ссылка **</A>**

Тэг **<A HREF="*URL*">**открывает описание ссылки, а тэг </A> - закрывает его. Любой текст, находящийся между данными двумя тэгами подсвечивается специальным образом Web-броузером. Обычно этот текст отображается подчеркнутым и выделенным синим (или другим заданным пользователем) цветом.

Текст, обозначающий URL, не отображается броузером, а используется только для выполнения предписанных им действий при активизации ссылки (обычно при щелчке мыши на подсвеченном или подчеркнутом тексте). Вот пример сегмента HTML-документа:

Для получения примера смотри
<A HREF="http:/www.ruswebmasters.com/index.htm> страницу </A>

Данная строка будет выглядеть на экране следующим образом:

Для получения примера смотри страницу

Ссылки на точки внутри документа

Вы можете делать ссылки на различные участки или разделы одного и того же документа, используя специальных скрытый маркер для этих разделов. Это позволяет быстро переходить от раздела к разделу внутри документа, не используя скроллирование экрана. Как только вы щелкнете на ссылке, броузер переместит вас на указанный раздел документа, а строка, в которой стоит маркер данного раздела (обычно, первая строка раздела или заголовок раздела) будет ращзмещена на первой строке окна броузера (если данная строка не присутствует уже на экране броузера).

Для создания такой ссылки необходимо выполнить следующие шаги:

1. Создайте маркер раздела. Синтаксис данного маркера следующий:

<A NAME="named_anchor"> Текст-который-отобразится-в-первой-строке-броузера </A>

2. Создайте ссылку на данный маркер:

<A HREF="#named_anchor"> Текст </A>

Например:

```
<p><b>Список разделов</b></p>
<ul> <li><a href="#ex1">Раздел 1</a></li>
<li><a href="#ex2">Раздел 2</a></li> </ul>
<p><a name="ex1"></a>Раздел 1</p>
<ul> <p>Текст раздела 1</p> </ul>
<p><a name="ex2"></a>Раздел 2</p>
<ul> <p>Текст раздела 2 <br></p>
```

Список разделов

- Раздел 1
- Раздел 2

Раздел 1

Текст раздела 1

Раздел 2

Текст раздела 2

Символы "#ex1" сообщает вашему броузеру, что необходимо найти в данном HTML-документе маркер с именем "ex1".

Когда пользователь щелкнет мышью на строке "Раздел 1", броузер перейдет сразу к разделу 1.

> **Внимание!** Как ранее было показано в синтаксисе URL, маркер раздела может быть поставлен как в том же документе, который просматривается в текущий момент, так и в другом документе. Во втором случае броузер осуществит подгрузку другого документа и перейдет к указанному для него разделу.

Графика внутри HTML-документа

Одна из наиболее привлекательных черт Web - возможность включения ссылок на графические и иные типы данных в HTML-документ. Делается это при помощи тэга <IMG...ISMAP>. Использование данного тэга позволяет значительно улучшить внешний вид и функциональность документов.

Существует два способа использования графики в HTML-документах. Первый - это внедрение графических образов в документ, что позволяет пользователю видеть изображения непосредственно в контексте других элементов документа. Это наиболее используемая техника при проектировании документов, называемая иногда "inline image". Синтаксис тэга:

<IMG SRC="*URL*" ALT="*text*" HEIGHT=n1 WIDTH=n2 ALIGN=*top*|*middle*|*bottom*|*texttop* ISMAP>

Опишем элементы синтаксиса тэга:

URL

Обязательный параметр, имеющий такой же синтаксис, как и стандартный URL. Данный URL указывает броузеру где находится рисунок. Рисунок должен храниться в графическом формате, поддерживаемом броузе-

ром. На сегодняшний день форматы GIF и JPG поддерживаются большинством броузеров.

ALT="text"

Данный необязательный элемент задает текст, который будет отображен броузером, не поддерживающим отображение графики или с отключенной подкачкой изображений. Обычно, это короткое описание изображения, которое пользователь мог бы или сможет увидеть на экране. Если данный параметр отсутствует, то на месте рисунка большинство броузеров выводит пиктограмму (иконку), активизировав которую, пользователь может увидеть изображение. Тэг ALT рекомендуется, если ваши пользователи используют броузер, не поддерживающий графический режим, например Lynx.

HEIGTH=n1

Данный необязательный параметр используется для указания высоты рисунка в пикселах. Если данный параметр не указан, то используется оригинальная высота рисунка. Это параметр позволяет сжимать или растягивать изображения по вертикали, что позволяет более четко определять внешний вид документа. Однако, некоторые броузеры не поддерживают данный параметр. С другой стороны, экранное разрешение у вашего клиента может отличаться от вашего, поэтому будьте внимательны при задании абсолютной величины графического объекта.

WIDTH=n2

Параметр также необязателен, как и предыдущий. Позволяет задать абсолютную ширину рисунка в пикселах.

ALIGN

Данный параметр используется, чтобы сообщить броузеру, куда поместить следующий блок текста. Это позволяет более строго задать расположение элементов на экране. Если данный параметр не используется, то большинство броузеров располагает изображение в левой части экрана, а текст справа от него.

ISMAP

Этот параметр сообщает броузеру, что данное изображение позволяет пользователю выполнять какие-либо действия, щелкая мышью на определенном месте изображения. Данная возможность является расширением HTML и будет обсуждена нами позже.

Приведем пример использования данного тэга:

<**IMG SRC="http://www.softexpress.com/images/nekton.jpg"** ALT="СофтСервис лого" ALIGN="top" ISMAP>

С версии HTML 2.0 у тэга <IMG> появились дополнительные параметры:

<IMG SRC="*URL*" ALT="*text*" HEIGHT=n1 WIDTH=n2 ALIGN=*top*|*middle*|*bottom*|*texttop*|***absmiddle***|***baseline***|***absbottom*** **BORDER=n3 VSPACE=n4 HSPACE=n5** ISMAP>

Новые параметры:

BORDER

Данный параметр позволяет автору определить ширину рамки вокруг рисунка.

VSPACE

Позволяет установить размер в пикселах пустого пространства над и под рисунком, чтобы текст не наезжал на рисунок. Особенно это важно для динамически формируемых изображений, когда нельзя заранее увидеть документ.

HSPACE

То же самое, что и VSPACE, но только по горизантали.

Фоновые рисунки

Большинство броузеров позволяет включать в документ фоновый рисунок, который будет матрицироваться и отображаться на фоне всего документа. Некоторые пользователи любят фоновую графику, некоторые нет. Ненавязчивый полупрозрачный рисунок (обои) обычно хорошо выглядит в качестве фона для большинства документов.

Описание фонового рисунка включается в тэг BODY и выглядит следующим образом:

<**BODY** BACKGROUND="*picture.gif*">

Задание стандартных цветов

Многие HTML-авторы любят использовать заранее предопределенные цвета фона документа, обычного текста и ссылок. Чтобы задать эти цвета, необходимо включить в тэг <BODY> дополнительные параметры:

<**BODY** BGCOLOR="#XXXXXX" TEXT="#XXXXXX" LINK="#XXXXXX">

где каждый из параметров определяет цвет того или иного элемента. Опишем эти параметры:

BGCOLOR

Цвет фона документа

TEXT

Цвет простого текста документа

LINK

Цвет ссылки

Цвет задается шестизначным числом в шестнадцатиричном формате по схеме RGB (Red, Green, Blue). Цвет #000000 соответствует черному, а цвет #FFFFFF - белому. Например:

<BODY BGCOLOR="#000000" TEXT="#FFFFFF" LINK="#9690CC">

Данная строка определяет белый цвет фона документа, черный текст и серебристые ссылки.

Горизонтальная линия

Используя тэг <HR> вы можете разделить текст горизонтальной чертой.

Формат тэга:

<HR SIZE=number WIDTH=*number*|*percent* ALIGN=*left*|*right*|*center* NOSHADE**>**

Параметры тэга:

SIZE

Толщина линии в пикселах.

WIDTH

Ширина линии в пикселах или процентах от ширина окна броузера.

ALIGN

Расположение на экране (слева | по центру | справа).

NOSHADE

По умолчанию линия представлена в 3D виде с тенью. NOSHADE позволяет представить линию просто однотонной темной полоской.

Добавление стилей в ваш HTML-документ

HTML позволяет использовать различные стили шрифтов для выделения текстовой информации в ваших документах. Вот короткий список стилей, поддерживаемых большинством броузеров:

- bold (жирный)
- italic (наклонный)
- mono spaced (type writer - с использованием фиксированных шрифтов)

Вы можете комбинировать различные виды стилей, например жирный и наклонный.

Стиль	***Элемент или тэг***	***Результат***
Bold	<B> Этот текст жирный </B>	**Этот текст жирный**
Italic	<I> Этот текст наклонный </I>	*Этот текст наклонный*
Mono spaced	<TT> Этот текст с непроп. шрифтом </TT>	Этот текст с непроп. шрифтом

Комбинирование стилей позволяет вам отображать в одной строке несколько элементов различными стилями, например:

Жизнь - это песня!

Жизнь - *это **песня!***

Внимание! Добавление большого количества стилей и их комбинаций приводит к затруднению чтения текста!

Дополнительные стили:

- big (юольшой)

- small (маленький)
- sub (подстрочник)
- sup (надстрочник)

Стиль	***Элемент или тэг***	***Результат***
Big	Этот текст <BIG> большой </BIG>	Этот текст большой
Small	Этот текст <SMALL> маленький </SMALL>	Этот текст маленький
Sub	Этот текст _{подстрочник}	Этот текст подстрочник
Sup	Этот текст ^{надстрочник}	Этот текст надстрочник

Размер шрифта <FONT SIZE>

Вы можете изменять размер шрифта при помощи тэга:

<FONT SIZE=+|- n>

Шрифт может иметь размер от 1 до 7. Вы можете прямо указать размер шрифта цифрой, или указать смещение относительно базового значения (по умолчанию - 3) в положительную или отрицательную сторону. Базовое значение можно изменить при помощи тэга:

<BASEFONT SIZE=n>

Например:

```
<p>и
<font SIZE=+1>з</font><font SIZE=+2>м</font>
<font SIZE=+3>е</font><font SIZE=+4>н</font>
<font SIZE=+3>е</font><font SIZE=+2>н</font>
<font SIZE=+1>и</font>
е</p>
```

изменение

Цвет шрифта <FONT SIZE>

Вы можете изменить цвет шрифта при помощи тэга:

<FONT COLOR="#xxxxxx>

Цвет указывается в RGB-формате (Red-Green-Blue) посредством указания размерности каждой компоненты цвета в шестнадцатиричном формате. Например, белый цвет обозначается "000000", черный - "FFFFFF", синий - "0000FF" и т.п.

<FONT COLOR="#FF0000">
Красный </FONT>
<FONT COLOR="#00FF00">
Зеленый </FONT>
<FONT COLOR="#0000FF">
Синий </FONT>

Красный Зеленый Синий

Специальные тэги HTML

Следующие тэги позволят вам сделать ваш HTML-документ более функциональным.

Тэг адреса <ADDRESS>

Тэг <ADDRESS> используется для выделения автора документа и его дреса (например, e-mail). Синтаксис:

<ADDRESS> Адрес-автора </ADDRESS>

Escape-последовательности

Некоторые символы являются управляющими символами в HTML и немогут напрямую использоваться в документе:

- левая угловая скобка "<"
- правая угловая скобка ">"
- амперсанд "&"
- двойные кавычки """

Чтобы использовать данные символы в документе, необходимо заменить их escape-последовательностями:

<	**<**
>	**>**
&	**&**
"	**"**

Существует большое количество escape-последовательностей для обозначения специальных символов, например "©" для обозначения знака © и ® для значка ®, появившихся в HTML 2.0. Одной из особенностей является замена символов во 2-ой части символьной таблицы (после 127-ого символа) на

escape-последовательности для передачи текстовых файлов с национальными языками по 7-битным каналам.

HTML формы

Некоторые WWW browser позволяют пользователю, заполнив специальную форму, возвращающую полученное значение, выполнять некоторые действия на вашем WWW-сервере. Когда форма интерпретируется WEB-броузером, создается специальные экранные элементы GUI, такие, как поля ввода, checkboxes, radiobuttons, выпадающие меню, скроллируемые списки, кнопки и т.д. Когда пользователь заполняет форму и нажимает кнопку "Подтверждение" (SUBMIT - специальный тип кнопки, который задается при описании документа), информация, введенна пользователем в форму, посылается HTTP-серверу для обработки и передаче другим программам, работающим под сервером, в соответствии с CGI (Common Gateway Interface) интерфейсом.

Когда вы описываете форму, каждый элемент ввода данных имеет тэг <INPUT>. Когда пользователь помещает данные в элемент формы, инфоромация размещается в разделе VALUE данного элемента.

Синтаксис

Все формы начинаются тэгом <FORM> и звершаются тэгом </FORM>.

```
<FORM METHOD="get|post" ACTION="URL">
Элементы_формы_и_другие_элементы_HTML
</FORM>
```

METHOD

Метод посылки сообщения с данными из формы. В зависимости от используемого метода вы можете посылать результаты ввода данных в форму двумя путями:

- **GET**: Информация из формы добавляется в конец URL, который был указан в описании заголовка формы. Ваша CGI-программа (CGI-скрипт) получает данные из формы в виде параметра переменной среды QUERY_STRING. Использование метода GET не рекомендуется.
- **POST**: Данный метод передает всю информацию о форме немедленно после обращения к указанному URL. Ваша CGI-программа получает данные из формы в стандартный поток ввода. Сервер не будет пересылать вам сообщение об окончании пересылки данных в стандартный поток ввода; вместо этого используется переменная среды CONTENT_LENGTH для определения, какое количество данных вам необходимо считать из стандартного потока ввода. Данный метод реко-

мендуется к использованию.

ACTION

ACTION описывает URL, который будет вызываться для обработки формы. Данный URL почти всегда указывает на CGI-программу, обрабатывающую данную форму.

Тэги Формы

TEXTAREA

Тэг <TEXTAREA> используется для того, чтобы позволить пользователю вводить более одной строки информации (свободный текст). Вот пример использовани тэга <TEXTAREA>:

```
<TEXTAREA NAME="address" ROWS=10 COLS=50>
Москва,
Дмитровкое шоссе,
д.9Б, офис 448
</TEXTAREA>
```

Атрибуты, используемые внутри тэга <TEXTAREA> описывают внешний вид и имя вводимого значения. Тэг </TEXTAREA> необходим даже тогда, когда поле ввода изначально пустое. Описание атрибутов:

- **NAME** - имя поля ввода
- **ROWS** - высота поля ввода в символах
- **COLS** - ширина поля ввода в символах

Если вы хотите, чтобы в поле ввода по умолчанию выдавался какой-либо текст, то необходимо вставить его внутри тэгов <TEXTAREA> и </TEXTAREA>.

INPUT

Тэг <INPUT> используется для ввода одной строки текста или одного слова. Атрибуты тэга:

- **CHECKED -** означает, что CHECKBOX или RADIOBUTTON будет выбран.
- **MAXLENGTH** - определяет количество символов, которое пользователи могут ввести в поле ввода. При превышении количества допустимых символов броузер реагирует на попытку ввода нового символа звуковым

сигналом и не дает его ввести. Не путать с атрибутом SIZE. Если MAXLENGTH больше чем SIZE, то в поле осуществляется скроллинг. По умолчанию значение MAXLENGTH равно бесконечности.

- **NAME** - имя поля ввода. Данное имя используется как уникальный идентификатор поля, по которому, впоследствии, вы сможете получить данные, помещенные пользователем в это поле.
- **SIZE** - определяет визуальный размер поля ввода на экране в символах.
- **SRC** - URL,. указывающий на картинку (используется совместно с атрибутом IMAGE).
- **TYPE** - определяет тип поля ввода. По умолчанию это простое поле ввода для одной строки текста. Остальные типы должны быть явно указаны:

CHECKBOX
Используется для простых логических (BOOLEAN) значений. Значение, ассоциированное с именем данного поля, которое будет передаваться в вызываемую CGI-программу, может принимать значение ON или OFF.

HIDDEN
Поля данного типа не отображаются броузером и не дают пользователю изменять присвоенные данному полю по умолчанию значение. Это поле используетс для передачи в CGI-программу статической информации, как то ID прользователя, пароля или другой информации.

IMAGE
Данный тип поля ввода позволяет вам связывать графический рисунок с именем поля. При нажатии мышью на какую-либо часть рисунка будет немедленно вызвана ассоциированная форме CGI-программа. Значения, присвоенные переменной NAME будут выглядеть так - создается две новых переменных: первая имеет имя, обозначенное в поле NAME с добавлением .x в конце имени. В эту переменную будет помещена X-координата точки в пикселах (считая началом координат левый верхний угол рисунка), на которую указывал курсор мыши в момент нажатия, а переменная с именем, содержащимся в NAME и добавленным .y, будет содержать Y-координату. Все значения атрибута VALUE игнорируются. Само описание картинки осуществляется через атрибут SRC и по синтаксису совпадает с тэгом <IMG>.

PASSWORD
То же самое, что и атрибут TEXT, но вводимое пользователем значение не отображается броузером на экране.
RADIO
Данный атрибут позволяет вводить одно значение из нескольких

альтернатив. Для создания набора альтернатив вам необходимо создать несколько полей ввода с атрибутом TYPE="RADIO" с разными значениями атрибута VALUE, но с одинаковыми значениями атрибута NAME. В CGI-программу будет передано значение типа NAME=VALUE, причем VALUE примет значение атрибута VALUE того поля ввода, которое в данный момент будет выбрано (будет активным). При выборе одного из полей ввода типа RADIO все остальные поля данного типа с тем же именем (атрибут NAME) автоматически станут невыбранными на экране.

RESET

Данный тип обозначает кнопку, при нажатии которой все поля формы примут значения, описанные для них по умолчанию.

SUBMIT

Данный тип обозначает кнопку, при нажатии которой будет вызвана CGI-программа (или URL), описанная в заголовке формы. Атрибут VALUE может содержать строку, которая будет высвечена на кнопке.

TEXT

Данный тип поля ввода описывает однострочное поле ввода. Используйте атрибуты MAXLENGTH и SIZE для определения максимальной длинны вводимого значения в символах и размера отображаемого поля ввода на экране (по умолчанию принимается 20 символов).

- **VALUE** - присваивает полю значение по умолчанию или значение, которое будет выбрано при использовании типа RADIO (для типа RADIO данный атрибут обязателен)

Меню выбора в формах

Под меню выбора в формах понимают такой элемент интерфейса, как LISTBOX. Существует три типа тэгов меню выбора для форм:

- **Select** - пользователь выбирает одно значение из фиксированного списка значений, представленных тэгами OPTION. Данй вид представляется как выпадающий LISTBOX.
- **Select single** - то же самое, что и Select, но на экране пользователь видит одновременно три элемента выбора. Если их больше, то предоставляется автоматический вертикальный скроллинг.
- **Select multiple** - позволяет выбрать несколько элементов из LISTBOX.

SELECT

Тэг SELECT позволяет пользователю выбрать значение из фиксированного

списка значений. Обычно это представлено выпадающим меню.

Тэг SELECT имеет один или более параметр пежду стартовым тэгом <SELECT> и завершающим </SELECT>. По умолчанию, первый элемент отображается в строке выбора. Вот пример тэга <SELECT>:

```
<FORM>
<SELECT NAME=group>
<OPTION> AT 386
<OPTION> AT 486
<OPTION> AT 586
</SELECT>
</FORM>
```

SELECT SINGLE

Тэг SELECT SINGLE - это то же самое, что и Select, но на экране пользователь видит одновременно несколько элементов выбора (три по умолчанию). Если их больше, то предоставляется автоматический вертикальный скроллинг. Количество одновременно отображаемых элементов определяется атрибутом SIZE. Пример:

```
<FORM>
<SELECT SINGLE NAME=group SIZE=4>
<OPTION> AT 386
<OPTION> AT 486
<OPTION> AT 586
<OPTIONS> Pentium PRO
</SELECT>
</FORM>
```

SELECT MULTIPLE

Тэг SELECT MULTIPLE похож на тэг SELECT SINGLE, но пользователь может одновременно выбрать более чем один элемент списка. Атрибут SIZE определяет количество одновременно видимых на экране элементов, атрибут MULTIPLE - максимальное количество одновременно выбранных элементов. Пример:

```
<FORM>
<SELECT SINGLE NAME=group SIZE=4 MULTIPLE=2>
<OPTION> AT 386
<OPTION> AT 486
<OPTION> AT 586
<OPTIONS> Pentium PRO
```

```
</SELECT>
</FORM>
```

Если выбрано одновременно несколько значений, то серверу передаютс соответствующее выбранному количество параметров NAME=VALUE с одинаковыми значениями NAME, но разными VALUE.

Отправление файлов при помощи форм

Формы можно использовать для отправки не только небольших информационных сообщений ввиде параметров, а также и для отправки файлов.

> **Внимание!** Поскольку данная возможность требует поддержки получения файлов WEB-сервером, то, соответственно, необходимо, чтобы сервер поддерживал получение файлов!

Например:

```
<FORM ENCTYPE="multipart/form-data" ACTION="url" METHOD=POST>
Отправить данный файл: <INPUT NAME="userfile" TYPE="file">
<P>
<INPUT TYPE="submit" VALUE="Отправить файл">
</FORM>
```

HTML фреймы

Испольльзуя фрэймы, позволяющие разбивать Web-страницы на множественные скроллируемые подокна, вы можете значительно улучшить внешний вид и функциональность информационных систем и Web-приложений. Каждое подокно, или фрэйм, может иметь следующие свойства:

- Каждый фрэйм имеет свой URL, что позволяет загружать его независимо от других фрэймов
- Каждый фрэйм имеет собственое имя (параметр NAME), позволяющее переходить к нему из другого фрэйма
- Размер фрэйма может быть изменен пользователем прямо на экране при помощи мыши (если это не запрешено указанием специального параметра)

Данные свойства фрэймов позволяют создавать продвинутые интерфейсные решения, такие как:

- Размещение статической информации, которую автор считает необходимым постоянно показывать пользователю, в одном статическом фрэйме. Это может быть графический логотип фирмы, copyright, набор управляю-

щих кнопок
- Помещение в статическом фрэйме оглавления всех или части WEB-документов, содержащихся на WEB-сервере, что позволяет пользователю быстро находить интересующую его информацию
- Создавать окна результатов запросов, когда в одном фрэйме находится собственно запрос, а в другом результаты запроса
- Создавать формы типа "мастер-деталь" для WEB-приложений, обслуживающих базы данных

Синтаксис фрэймов

Формат документа, использующего фрэймы, внешне очень напоминает формат обычного документа, только вместо тэга BODY используется контейнер FRAMESET, содержащий описание внутренних HTML-документов, содержащий собственно информацию, размещаемую во фрэймах.

```
<HTML>
<HEAD>...</HEAD>
<FRAMESET>...</FRAMESET>
</HTML>
```

Однако, фрэйм-документ является специфичным видом HTML-документа, поскольку не содержит элемента BODY и какой-либо информационной нагрузки соответственно. Он описывает только фрэймы, которые будут содержать информацию (кроме случая двойного документа, который мы рассмотрим позже).

+Представим общий синтаксис фрэймов:

```
<FRAMESET COLS="value" | ROWS="value">

    <FRAME SRC="url1">
    <FRAME ...>
    ...

</FRAMESET>
```

Общий контэйнер FRAMESET описывает все фрэймы, на которые делится экран. Вы можете разделить экран на несколько вертикальных или несколько горизонтальных фрэймов. Тэг FRAME описывает каждый фрэйм в отдельности. Рассмотрим более детально каждый компонент.

FRAMESET

<FRAMESET [COLS="value" | ROWS="value"]>

Тэг <FRAMESET> имеет завершающий тэг </FRAMESET>. Все, что может находиться между этими двумя тэгами, это тэг <FRAME>, вложенные тэги <FRAMESET> и </FRAMESET>, а также контейнер из тэгов <NOFRAME> и </NOFRAME>, который позволяет строить двойные документы для броузеров, поддерживающих фрэймы и не поддерживающих фрэймы.

Данный тэг имеет два взаимоисключающих параметра: ROWS и COLS.

ROWS="список-определений-горизонтальных-подокон"

Данный тэг содержит описания некоторого количества подокон, разделенные запятыми. Каждое описание представляет собой числовое значение размера подокна в пикселах, процентах от всего размера окна или связанное масштабное значение. Количество подокон определяется количеством значений в списке. Общая сумма высот подокон должна составлять высоту всего окна (в любых измеряемых величинах). Отсутствие атрибута ROWS определяет один фрэйм, величиной во все окно броузера.

Синтаксис используемых видов описания величин подокон:

value

Простое числовое значение определяет фиксированную высоту подокна в пикселах. Это далеко не самый лучший способ описания высоты подокна, поскольку различные броузеры имеют различный размер рабочего поля, не говоря уже о различных экранных разрешениях у пользователя. Если вы, все же, используете данный способ описания размера, то настоятельно рекомендуется сочетать его с каким-либо другим, чтобы в результате вы точно получили 100%-ное заполнение окна броузера вашего пользователя.

value%

Значение величины подокна в процентах от 1 до 100. Если общая сумма процентов описываемых подокон превышает 100, то размеры всех фрэймов пропорционально уменьшаются до суммы 100%. Если, соответственно, сумма меньше 100, то размеры пропорционально учеличиваются.

value*

Вообще говоря, значение value в данном описании является необязательным. Символ "*" указывает на то, что все оставшееся место будет при-

надлежать данному фрэйму. Если указывается два или более фрэйма с описанием "*" (например "*,*"), то оставшееся пространство делится поровну между этими фрэймами. Если перед звездочкой стоит цифра, то она указывает пропорцию для данного фрэйма (во сколько раз од будет больше аналогично описанного чистой звездочкой). Например, описание "3*,*,*", говорит, что будет создано три фрэйма с размерами 3/5 свободного пространства для первого фрэйма и по 1/5 для двух других.

COLS="список-определений-горизонтальных-подокон"

То же самое, что и ROWS, но делит окно по вертикали, а не по горизонтали.

Внимание! Совместное использование данных параметров может привести к непредствазуемым результатам. Например, строка: <FRAMESET ROWS="50%,50%" COLS "50%,50%"> может привести к ошибочной ситуации.

Примеры:

<FRAMESET COLS="50,*,50"> - описывает три фрэйма, два по 50 точек справа и слева, и один внутри этих полосок.

<FRAMESET ROWS="20%,3*,*"> - описывает три фрэйма, первый из которых занимает 20% площади сверху экрана, второй 3/4 оставшегося от первого фрэйма места (т.е. 60% всей площади окна), а последний 1/4 (т.е. 20% всей площади окна.

<FRAMESET ROWS="*,60%,*"> - аналогично предыдущему примеру.

Тэги <FRAMESET> могут быть вложенными, т.е. например:

```
<FRAMESET ROWS="50%,50%">
    <FRAMESET COLS="*,*"
    </FRAMESET>
</FRAMESET>
```

Результат данного примера мы рассмотрим позже.

FRAME

<FRAME SRC="*url*" [NAME="frame_name"] [MARGINWIDTH="nw"] [MARGINHEIGHT="nh"] [SCROLLING=yes|no|auto] [NORESIZE]>

Данный тэг определяет фрэйм внутри контейнера FRAMESET.

SRC="url"

Описывает URL документа, который будет отображен внутри ланного фрэйма. Если он отсутствует, то будет отображен пустой фрэйм.

NAME="frame_name"

Данный параметр описывает имя фрэйма. Имя фрэйма может быть использовано для определения действия с данным фрэймом из другого HTML-документа или фрэйма (как правило, из соседнего фрэйма этого же документа). Имя обязательно должно начинаться с символа. Содержимое поименованных фрэймов может быть задействовано из других документов при помощи специального атрибута TARGET, описываемого ниже.

MARGINWIDTH="value"

Это атрибут может быть использован, если автор документа хочет указать величину разделительных полос между фрэймами сбоку. Значение *value* указывается в пикселах и не может быть меньше единицы. По умолчанию данное значение зависит от реализации поддержки фрэймов используемым клиентом броузером.

MARGINHEIGHT="value"

То же самое, что и MARGINWIDTH, но для верхних и нижних величин разделительных полос.

SCROLLING="yes | no | auto"

Этот атрибут позволяет задавать наличие полос прокрутки у фрэйма. Параметр *yes* указывает, что полосы прокрутки будут в любом случае присутствовать у фрэйма, параметр *no* наоборот, что полос прокрутки не будет. *Auto* определяет наличие полос прокрутки только при их необходимости (значение по умолчанию).

NORESIZE

Данный атрибут позволяет создавать фрэймы без возможности измене-

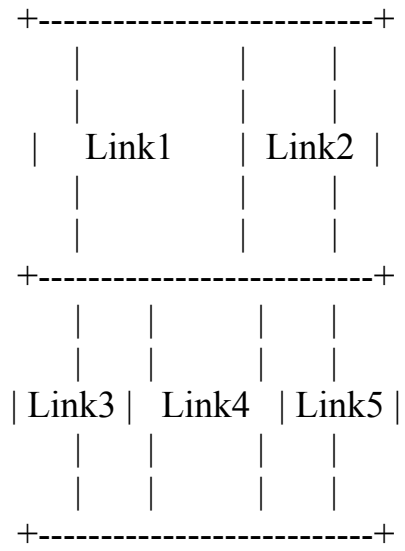
ния размеров. По умолчанию, размер фрэйма можно изменить при помощи мыши так же просто, как и размер окна Windows. NORESIZE отменяет данную возможность. Если у одного фрэйма установлен атрибут NORESIZE, то у соседних фрэймов тоже не может быть изменен размер со стороны данного.

NOFRAMES

Данный тэг используется в случае, если вы создаете документ, который может просматриваться как броузерами, поддерживающими фрэймы, так и броузерами, их не поддерживающими. Данный тэг помещяется внутри контейнера FRAMESET, а все, что находится внутри тэгов <NOFRAMES> и </NOFRAMES> игнорируется броузерами, поддерживающими фрэймы.

Примеры

Рассмотрим реализацию фрэймов для подобного разбиения окна:




```
<FRAMESET ROWS="*,*">
<NOFRAMES>
<H1>Ваша версия WEB-броузера не поддерживает фрэймы!</H1>
</NOFRAMES>
    <FRAMESET COLS="65%,35%">
    <FRAME SRC="link1.html">
    <FRAME SRC="link2.html">
    </FRAMESET>
    <FRAMESET COLS="*,40%,*">
    <FRAME SRC="link3.html">
    <FRAME SRC="link4.html">
    <FRAME SRC="link5.html">
```

```
    </FRAMESET>
</FRAMESET>
```

Планирование фрэймов и взаимодействия между фрэймами

С появлением фрэймов сразу возникает вопрос: "*А как сделать так, чтобы нажимая на ссылку в одном фрэйме инициировать появление информации в другом?*"

Ответом на данный вопрос является **планирование взаимодействия фрэймов** (далее - планирование). Каждый фрэйм может иметь собственное имя, определяемое параметром NAME при описании данного фрэйма. Существует, также, специальный атрибут - TARGET, позволяющий определять, к какому фрэйму относится та или иная операция. Формат данного атрибута следующий:

TARGET="windows_name"

Данный атрибут может встречаться внутри различных тэгов:

TARGET в тэге **A**

Это самое прямое использование TARGET. Обычно, при активизации пользователем ссылки соответствующий документ появляется в том же окне (или фрэйме), что и исходный, в котором была ссылка. Добавление атрибута TARGET позволяет произвести вывод документа в другой фрэйм. Например:

```
<A HREF="mydoc.html" TARGET="Frame1"> Переход в фрэйм № 1 </A>
```

TARGET в тэге **BASE**

Размещение TARGET в тэге BASE позволит вам не указывать при описании каждой ссылки фрэйм-приемник документов, вызываемых по ссылкам. Это очень удобно, если в одном фрэйме у вас находится меню, а в другой - выводится информация. Например:

Документ № 1.

```
<FRAMESET ROWS="20,*">
<FRAME SRC="doc2.htm" NAME="Frame1">
<FRAME SRC="doc3.htm" NAME="Frame2">
</FRAMESET>
```

Документ № 2 (doc2.htm).

```
<HTML>
<HEAD>
<BASE TARGET="Frame2">
</HEAD>
<BODY>
<A HREF="url1"> Первая часть</A> |
<A HREF="url2"> Вторая часть</A>
</BODY>
</HTML>
```

TARGET в тэге **AREA**

Таже можно включать тэг TARGET в описание ссылки при создании карты изображения. Например:
<AREA SHAPE="circle" COORDS="100,100,50" HREF="http://www.soft-express.com" TARGET="Frame1">

TARGET в тэге FORM

То же относится и к определению формы. В данном случае, после обработки переданных параметров формы результирующий документ появится в указанном фрэйме.
<FORM ACTION="url" TARGET="window_name">

Внимание! Имя окна (фрэйма) в параметре TARGET должно начинаться с латинской буквы или цифры. Также необходимо помнить, что существуют зарезервированные имена для разрешения специальных ситуаций.

Зарезервированные имена фрэймов

Зарезервированные имена фрэймов служат для разрешения специальных ситуаций. Все они начинаются со знака подчеруивания. Любые другие имена фрэймов, начинающиеся с подчеркивания будут игнорироваться броузером.

TARGET="_blank"

Данное значение определяет, что документ, полученный по ссылке будет отображаться в новом окне броузера.

TARGET="_self"

Данное значение определяет, что документ, полученный по ссылке будет отображаться в том же фрэйме, в котором находится ссылка. Это имя удобно для переопределения окна назначения, указанного ранее в тэге **BASE**.

TARGET="_parent"

Данное значение определяет, что документ, полученный по ссылке будет отображаться в родительском окне, вне зависимости от параметров **FRAMESET**. Если родительского окна нет, то данное имя аналогично "_self".

TARGET="_top"

Данное значение определяет, что документ, полученный по ссылке будет отображаться на всей поверхности окна, вне зависимости от наличия фрэймов. Использование данного параметра удобно в случае вложенных фрэймов.

7. Перечень программных продуктов, реально используемых в практике деятельности выпускников.

Студенты в специализированной аудитории по подготовке курсовых и дипломных работ имеют доступ на информационные программы Гарант, Труд-эксперт, Консультант плюс, Зеркало, Облако, Stalker, Вода, MathCAD, пакет MS Office.

8. Комплекты заданий для лабораторных работ изложены в материалах данного УМКД.

9. Комплекты билетов для зачета по дисциплине «Программное обеспечение задач БЖД».

Билет 1

1. Программные продукты и их основные характеристики.
2. СУБД MS Access. Связи между таблицами. Ключи, макросы модули и объекты.

Билет 2

1. Защита программных продуктов. Правовые методы защиты программных продуктов.
2. Инструментарий технологии программирования.

Билет 3

1. Классификация программных продуктов.
2. Проблемно-ориентированные ППП.

Билет 4

1. Базовое программное обеспечение.
2. ППП автоматизированного проектирования.

Билет 5

1. Основные понятия теории баз данных. Классификация баз данных.
2. ППП общего назначения.

Билет 6

1. Модели данных.
2. Офисные ППП.

Билет 7

1. Структурные элементы базы данных.
2. Методо-ориентированные ППП.

Билет 8

1. Системы управления базами данных.
2. Настольные издательские системы.

Билет 9

1. СУБД в многопользовательских системах.
2. Работа с текстом: минимальный набор типовых операций.

Билет 10

1. Технология использования СУБД.
2. Расширенный набор типовых операций.

Александр Александрович Дрюков,
ассистент кафедры БЖД АмГУ

Программное обеспечение задач БЖД: УМКД

_____
Изд-во АмГУ. Подписано к печати _____ Формат _____. Усл. печ. л. _____, уч. изд. л. ____. Тираж 100. Заказ _____.
Отпечатано в типографии АмГУ.