

Федеральное агентство по образованию  
Государственное образовательное учреждение  
высшего профессионального образования  
Владимирский государственный университет  
Кафедра информационных систем  
и информационного менеджмента

# ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Методические указания к лабораторным работам

Составитель  
С.Ю. КИРИЛЛОВА

Владимир 2006

УДК 004  
ББК 32.97  
И 74

Рецензент

Кандидат технических наук, доцент кафедры  
информационных систем и информационного менеджмента  
Владимирского государственного университета  
*В.Ф. Романов*

Печатается по решению редакционно-издательского совета  
Владимирского государственного университета

**Информационные** технологии : метод. указания к лаб. ра-  
И74 ботам / сост. Владим. гос. ун-т ; С. Ю. Кириллова. – Владимир :  
Изд-во Владим. гос. ун-та., 2006. – 64 с.

Содержат краткое описание приемов и методов работы с базами данных, подготовки компьютерных презентаций, разработки web-страниц, решения оптимизационных задач. Приводятся примеры, снабженные необходимыми комментариями, задания для самостоятельного выполнения. Методические указания ориентированы на использование приложений пакета MS Office. Даны задания для лабораторных работ, содержание отчетов.

Предназначены для проведения лабораторных работ по дисциплине «Информационные технологии» в соответствии с действующими Государственными образовательными стандартами и учебными планами направления 230200 подготовки специалистов и бакалавров, а также для студентов других направлений, изучающих дисциплины соответствующего профиля.

Табл. 8. Ил. 19. Библиогр.: 11 назв.

УДК 004  
ББК 32.97

## **ВВЕДЕНИЕ**

*Информационные технологии (ИТ) являются составной частью научного направления «Информатика» и базируются на ее достижениях. Информатизация как процесс перехода к информационному обществу сопровождается возникновением новых и развитием существующих ИТ. Информация превращается в коммерческий ресурс, способствуя получению прибыли при внедрении ИТ во многие сферы человеческой деятельности. Сфера информатики становится доминирующей в деятельности человека, потребляя уже в настоящее время большую долю трудовых ресурсов, чем материальная. Поэтому понятно то пристальное внимание, которое уделяет компьютерному образованию мировое сообщество, стремление унифицировать и упорядочить знания, необходимые специалисту этого направления. Одним из результатов такой работы является Международный стандарт по компьютерному образованию Computing Curricula 2001 – Computer Science.*

*Внедрение ИТ требует подготовки как пользователей, так и разработчиков. В Российской высшей школе подготовка профессионалов в области ИТ осуществляется по ряду направлений, из которых наиболее глубоко реализуется по направлениям 230100 «Информатика и вычислительная техника» и 230200 «Информаци-*

*онные системы». Базовая дисциплина общепрофессиональной подготовки для этих направлений – «Информационные технологии», в результате ее изучения студенты должны:*

*знать базовые информационные процессы, структуру, модели, методы и средства базовых и прикладных ИТ; методику создания, проектирования и сопровождения систем на базе ИТ;*

*уметь применять ИТ при решении функциональных задач в различных предметных областях, а также при разработке информационных систем;*

*иметь представление об областях применения ИТ и их перспективах в условиях перехода к информационному обществу.*

*Данные методические указания не охватывают всю ширину и глубину ИТ, предназначены, в первую очередь, для студентов младших курсов, обеспечивая лабораторный практикум по таким базовым информационным процессам, как обработка, хранение и представление информации, поддержка принятия решений.*

## **Лабораторная работа № 1**

### **КОМПЬЮТЕРНЫЕ ПРЕЗЕНТАЦИИ**

#### **1. Цель работы**

Рассмотреть области применения компьютерных презентаций, изучить и на практике освоить технологию подготовки компьютерных презентаций с использованием мультимедиа технологий средствами приложения MS PowerPoint.

#### **2. Основные сведения**

Термин «*мультимедиа*» — калька с английского слова multimedia, что можно перевести как «многие среды» (от multi — много и media — среда). *Мультимедиа технология* позволяет одновременно использовать различные способы представления информации: числа, текст, графику, анимацию, видео и звук. Важной особенностью мультимедиа технологий является ее *интерактивность*, то есть то, что в диалоге с компьютером пользователю отводится активная роль. Графический интерфейс мультимедийных проектов обычно содержит различные управляющие элементы (кнопки, текстовые окна и так далее).

*Компьютерные презентации* являются одним из типов мультимедийных проектов. Компьютерные презентации часто применяются в рекламе, при выступлениях на конференциях и совещаниях, они могут также использоваться на учебных занятиях в процессе объяснения материала преподавателем или докладов учащихся. Автоматический режим презентации часто используют во время проведения различных выставок. Компьютерная презентация представляет собой последовательность *слайдов*, содержащих мультимедийные объекты. Переход между слайдами осуществляется с помощью управляющих объектов (кнопок) или гиперссылок.

Наиболее эффективным и универсальным средством для подготовки презентаций является приложение Microsoft Office — PowerPoint. Оно по-

зволяет создать качественную презентацию с использованием графической информации, слайдов, звука, видеоклипов, эффектов анимации и т. д. Для подготовки презентации можно использовать тексты, схемы и диаграммы из приложений Word или Excel, а также рисунки со сканера.

### 2.1. Разработка презентации

Создание презентации целесообразно начинать с разработки проекта, в котором необходимо определить примерное количество слайдов в презентации и их содержание. Создадим, например, проект учебной презентации «Знакомимся с компьютером», которая будет посвящена рассмотрению устройства компьютера. Последовательность слайдов этой презентации может быть, например, такой:

- слайд 1 – «Знакомимся с компьютером»;
- слайд 2 – «Структурная схема компьютера»;
- слайд 3 – «Долговременная память»;
- слайд 4 – «Устройства ввода».

Рассмотрим основные шаги создания презентации:

1. В окне приложения PowerPoint ввести команду **Файл/Создать/Новая презентация...**

Кроме этого возможно создание презентаций в следующих режимах: **Из шаблона оформления, Из мастера автосодержания..., Из имеющейся презентации..., Фотоальбом...**

2. При выполнении команды **Создать слайд** открывается панель *автомкета слайда*, содержащая различные варианты разметки слайда. Текстовая информация на слайде может быть расположена либо в виде маркированного списка, либо в две колонки. Слайд целиком может занимать таблица или диаграмма, на нем могут находиться текст и диаграмма, текст и графика и так далее. Большинство типов слайдов содержат также заголовки. Наконец, есть пустые заготовки слайдов с заголовком и без него.

Процедура заполнения слайда информацией одинакова для слайдов всех видов. Достаточно щелкнуть мышью в выбранной области и набрать свой текст или скопировать туда рисунок, диаграмму и пр.

3. Создадим титульный слайд презентации «Знакомимся с компьютером». Первый слайд презентации обычно содержит ее название и создается на основе *Титульного слайда*.

4. Второй слайд называется «Структурная схема компьютера». Здесь мы в дальнейшем поместим рисунок структурной схемы компьютера. Вве-

сти команду Вставка/ Новый слайд.... На диалоговой панели Создать слайд выбрать тип автомакета Только заголовок. Щелкнуть на поле заголовка и ввести текст «Структура компьютера».

5. Третий слайд «Долговременная память» будет содержать таблицу из трех столбцов и четырех строк. В таблице будут содержаться названия устройств долговременной (внешней) памяти и их информационная емкость. Создать новый слайд и выбрать тип автомакета Таблица. Выбрать количество столбцов и строк таблицы. Ввести заголовок и заполнить таблицу.

6. PowerPoint предоставляет возможность красиво оформить внешний вид таблицы с помощью команды Формат/Таблица....

7. Четвертый слайд «Устройства ввода» будет содержать названия устройств ввода и их изображения, которые будут размещены в две колонки. Создать новый слайд, для которого выбрать тип автомакета Текст в две колонки. Ввести заголовок и текст.

## *2.2. Рисунки и графические примитивы на слайдах*

Рисунок для слайда можно создать с помощью графического редактора, а затем поместить на слайд командой Вставка/Рисунок/Из файла.... Можно воспользоваться коллекцией рисунков, которая имеется в Microsoft Office. Рисунки из коллекции добавляются с помощью команды Вставка/Рисунок/Картинки.... Картинки можно изменять. Для этого используется панель Настройка изображения. Эта панель появляется на экране после выделения какого-либо графического объекта или вызывается командой Вид/Панели инструментов/Настройка изображения.

На слайдах можно разместить различные графические примитивы (линии, прямоугольники, фигурные стрелки и пр.). Для этого необходимо воспользоваться панелью Рисование, которая находится обычно в нижней части окна приложения. С помощью графических примитивов можно создавать различные схемы, например схему «Структурная схема компьютера» для слайда 2.

## *2.3. Дизайн презентации, редактирование и сортировка слайдов*

Теперь можно выбрать дизайн презентации из коллекции, которая имеется в PowerPoint. Для этого необходимо ввести команду Формат/Применить оформление.... На появившейся диалоговой панели Применить шаблон оформления в раскрывающемся списке можно выбирать

различные стили дизайна и просматривать их в окне просмотра. Выбрав подходящий, например Капсулы, надо нажать кнопку Применить. Все слайды разработанной презентации получают выбранный дизайн. Кроме того, можно изменить цветовую схему слайдов.



Рис. 1

приложения или панель кнопок (рис. 1), которая располагается в нижнем левом углу экрана.

PowerPoint позволяет редактировать каждый слайд по отдельности в режиме Слайды, а также просматривать все слайды одновременно и сортировать их в режиме Сортировщик слайдов. Для переключения режимов просмотра можно использовать пункт Вид меню

#### *2.4. Использование анимации в презентации*

PowerPoint позволяет «оживить» демонстрацию презентации с помощью анимации. Можно создать эффекты анимации при смене одного слайда следующим, настроить анимацию, установить способы показа слайдов через команды меню Показ слайдов. Можно указать, какой анимационный эффект будет использоваться при смене слайдов, какими звуками это будет сопровождаться, что будет вызывать смену кадров – щелчок мыши или истекший интервал времени, и так далее. Выбранные настройки можно применить как к одному текущему слайду, так и сразу ко всем слайдам презентации.

Любой объект, находящийся на слайде, можно заставить возникать на экране необычно: проявляться, вылетать сбоку; а также разворачиваться до заданного размера, уменьшаться, вспыхивать, вращаться и так далее. Текст может появляться целиком, по словам или даже по отдельным буквам. Для установки значений параметров анимации объекта его необходимо выделить, а затем в контекстном меню выбрать пункт Настройка анимации.

#### *2.5. Переходы между слайдами*

Мультимедийная презентация создана, слайды содержат красиво оформленный текст, иллюстрации, звуковые эффекты и даже анимацию.



Теперь нужно сделать презентацию интерактивной. Для этого необходимо в процессе демонстрации презентации иметь возможность изменять последовательность предъявления слайдов.

Существуют два различных способа создания переходов. Первый способ состоит в создании *гиперссылок* на другие слайды или, в общем случае, на другие объекты (документы на локальном компьютере и Web-страницы в Интернет). Второй способ состоит в размещении на слайдах *управляющих элементов* (например, *кнопок*). Если активизировать кнопку (щелкнуть мышью), то произойдет некоторое событие (в данном случае переход на другой слайд).

Гиперссылку можно закрепить за любым объектом на слайде. Щелчком мыши выделить объект и ввести команду Вставка/Гиперссылка... (или через контекстное меню). Появится диалоговая панель Добавить гиперссылку. В текстовом поле Связать с: можно указать полный адрес какого-либо файла на локальном компьютере или адрес Web-страницы в Интернете. В этом случае в процессе демонстрации презентации при активизации данной ссылки в презентацию добавлялся бы внешний файл или документ. Однако в нашем случае необходимо создать ссылку на слайд данной презентации. Для этого щелкнуть по кнопке местом в документе и выбрать слайд, на который будет осуществляться переход.

Возврат со слайдов на исходный слайд реализуется с помощью кнопок, которые должны быть размещены на соответствующих слайдах. Ввести команду Показ слайдов/Управляющие кнопки. На панели Кнопки действий выбрать кнопку Возврат и установить для нее местоположение, размер и цвет.

Для задания действий, которые будут производиться по нажатию на созданную кнопку Возврат, в контекстном меню кнопки выбрать пункт Настройка действия. На появившейся одноименной панели из раскрывающегося списка выбрать нужный слайд.

## *2.6. Демонстрация презентации*

Запуск демонстрации презентации может осуществляться либо командой Вид/Показ слайдов, либо нажатием кнопки Показ слайдов на панели кнопок. Если делать это с помощью кнопки, то предварительно надо вызвать на экран первый слайд презентации, так как кнопка запускает демонстрацию, начиная с текущего слайда.

Для перехода от одного слайда к другому, следующему за ним, нажимают клавишу <Enter> или щелкают левой кнопкой мыши. Для перемещения по слайдам презентации вперед или назад можно пользоваться клавишей <PageUp> или <PageDown>.

В процессе показа слайдов указатель мыши не виден на экране, но он сразу появляется, стоит только начать перемещение мыши. Одновременно с курсором в нижнем левом углу экрана появляется почти сливающееся с фоном изображение кнопки. Нажатие на нее вызывает раскрывающееся меню, с помощью которого также можно управлять ходом демонстрации.

В процессе демонстрации презентации для перехода на нужный слайд можно также пользоваться управляющими кнопками и гиперссылками.

### **3. Задание к работе**

Создать мультимедийную интерактивную презентацию по выбранной самостоятельно теме или в соответствии с предложенными вариантами.

*Вариант 1.* Разработать презентацию «Глобальная компьютерная сеть Интернет»

*Вариант 2.* Создать презентацию любимых певцов или групп с использованием фотографий, текстов песен, музыкальных вставок, видеоклипов, кнопок перехода, эффектов анимации.

*Вариант 3.* Создать видеофильм на тему «Великие путешественники». Используя географические карты, отобразить путь следования первооткрывателей при помощи средств рисования PowerPoint.

*Вариант 4.* Создать видеофильм на тему «Объяснение нового материала» по любой дисциплине. При подготовке желательно использовать сканированные картинки или картинки из электронных энциклопедий.

*Вариант 5.* Создать презентацию по достопримечательным местам. При выполнении работы можно применять разные формы подачи материала: прогулка, шуточное знакомство, история возникновения и т. д.

### **4. Содержание отчета**

1. Цель работы.
2. Вариант индивидуального задания.
3. Структура презентации.
4. Распечатка слайдов.

5. Выводы по работе.

### **5. Контрольные вопросы**

1. В каком базовом информационном процессе могут применяться компьютерные презентации?
2. Каково назначение приложения PowerPoint?
3. Что можно получить в результате создания презентации?
4. Как создать слайд?
5. Что применяется в оформлении презентации?
6. Какие существуют способы задания переходов между слайдами и чем они отличаются?
7. Как настроить анимацию текста и картинки?
8. Как запустить презентацию на исполнение?
9. Какие операции можно осуществить при помощи Сортировщика слайдов?
10. Как создать и настроить кнопку?

## **Лабораторная работа № 2 ОСНОВЫ ЯЗЫКА РАЗМЕТКИ ГИПЕРТЕКСТОВЫХ ДОКУМЕНТОВ**

### **1. Цель работы**

Рассмотреть области применения Web-сайтов, изучить и на практике освоить основные элементы языка разметки гипертекстовых документов HTML, разработать проект и практически реализовать Web-сайт.

### **2. Основные сведения**

#### *2.1. Web-сайты и Web-страницы*

Публикации в World Wide Web (Всемирной паутине) реализуются в форме Web-сайтов. Web-сайт состоит из компьютерных Web-страниц. Сайт является *интерактивным* и *мультимедийным* средством представления информации. Обычно сайт имеет титульную страницу (страницу с оглавлением), на которой имеются *гиперссылки* на его основные разделы (Web-страницы). Гиперссылки также имеются на других Web-страницах сайта, что обеспечивает возможность пользователю свободно перемещать-

ся по сайту. Web-страницы сайта могут содержать *динамические объекты* (исполнимые модули), созданные с использованием сценариев на языках JavaScript и VBScript или элементов управления ActiveX. Расположенные на сайте управляющие элементы (например, кнопки) позволяют пользователю запускать те или иные динамические объекты.

Создание Web-сайтов реализуется с использованием языка разметки гипертекстовых документов HTML (Hyper Text Markup Language). Язык HTML используется для задания логической структуры документа (заголовки, абзацы, графические изображения и прочие объекты) и является той средой, в которой размещаются остальные компоненты Web-страницы. Технология HTML состоит в том, что в обычный текстовый документ вставляются управляющие символы – *теги*, и в результате мы получаем Web-страницу. Для создания Web-страниц используются простейшие текстовые редакторы, которые не включают в создаваемый документ управляющие символы форматирования текста. В качестве такого редактора в Windows можно использовать стандартное приложение Блокнот.

## 2.2. Основные теги

Вид Web-страницы задается тэгами, которые заключаются в угловые скобки. Тэги могут быть одиночными или парными, для которых обязательно наличие открывающего и закрывающего тегов (такая пара тегов называется *контейнером*). Закрывающий тэг содержит прямой слэш (/) перед обозначением. Тэги могут записываться как прописными, так и строчными буквами.

HTML-код страницы помещается внутрь контейнера `<HTML>` `</HTML>`. Без этих тегов браузер не в состоянии определить формат документа и правильно его интерпретировать. Web-страница разделяется на две логические части: заголовок и содержание.

Заголовок Web-страницы заключается в контейнер `<HEAD>` `</HEAD>` и содержит название документа и справочную информацию о странице, которая используется браузером для ее правильного отображения (например, тип кодировки, ключевые слова документа, используемые поисковыми роботами). Ключевые слова и кодировка, используемая на странице, задаются соответственно следующими тэгами:

```
<META name="keywords" content="слово1, слово2, слово3">  
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;  
charset=windows-1251">
```

Название Web-страницы содержится в контейнере `<TITLE>` `</TITLE>` и отображается в строке заголовка браузера при просмотре страницы.

Основное содержание страницы помещается в контейнер `<BODY>``</BODY>` и может включать текст, таблицы, бегущие строки, ссылки на графические изображения и звуковые файлы и так далее.

Созданную Web-страницу необходимо сохранить в виде файла. Принято сохранять титульный файл сайта, то есть тот, который первый загружается в браузер, под именем `index.htm`. В качестве расширения файла Web-страницы можно также использовать `html`. Необходимо различать имя файла `index.htm`, то есть имя, под которым Web-страница хранится в файловой системе, и собственно имя Web-страницы, которая высвечивается в верхней строке окна браузера и в первую очередь анализируется поисковыми системами. Имя Web-страницы должно в максимальной степени соответствовать ее содержанию. Рекомендуется создать для размещения сайта специальную папку и сохранять все файлы разрабатываемого сайта в этой папке.

Тег `<BODY>` может содержать атрибуты:

- `b bgcolor="цвет"` – для заливки заднего фона документа цветом. Цвет можно указывать явно (`black`, `maroon`, `green`, `olive`, `navy`, `purple`, `teal`, `gray`, `silver`, `red`, `lime`, `yellow`, `blue`, `fuchsia`, `aqua`, `white`) или через код цвета (например, `#45BAB1`);
- `background="местоположение и название рисунка"` – для задания в качестве фона рисунка. Если рисунок находится в одной директории с документом, тогда достаточно указать лишь его название. Если рисунков много, то их удобнее выделить в отдельные директории и указывать их название с полным описанием пути к ним (при использовании HTML-документа только на локальном диске), например: `<BODY background="C:\html\pic\fon.gif">`. Если же сайт опубликован в Интернете, то и путь надо указывать с адреса Web-страницы – URL (Universal Resource Locator – универсальный указатель ресурсов), например: `<BODY background="http://www.abcd.ru/fon.gif">`.

Таким образом, простая Web-страница будет содержать:

```
<HTML>
<HEAD>
<TITLE>Заголовок</TITLE>
<META name="keywords" content="слово1, слово2, слово3">
```

```

    <META HTTP-EQUIV="Content-Type" CONTENT="text/html;
charset=windows-1251">
    </HEAD>
    <BODY background="pic\fon.gif">
Тело документа (основной HTML-текст)
    </BODY>
</HTML>

```

Загрузить этот файл в окно браузера для просмотра в виде Web-страницы и в виде HTML-кода (через меню Вид).

### 2.3. Форматирование текста

Основным элементом текста в HTML является абзац (параграф). Абзац заключается в контейнер `<P></P>`. При просмотре в браузере абзацы отделяются друг от друга интервалами. Каждый абзац можно выравнивать по левому краю, по правому краю или по центру. Для этого служит указываемое в начальном теге параграфа свойство `align` со значениями `left`, `center`, `right`. Например: `<p align="left">`.

Принудительный переход на новую строку выполняется с помощью тега `<BR>`. Например:

```
<p align="center">Строка1<br>Строка2<br>Строка3</p>
```

Текст можно выделять жирным и/или курсивом, как показано в табл. 1.

Таблица 1

Запись в HTML	Отображение в браузере
Простой текст	Простой текст
<code>&lt;B&gt;</code> Выделение жирным <code>&lt;/B&gt;</code>	<b>Выделение жирным</b>
<code>&lt;I&gt;</code> Выделение курсивом <code>&lt;/I&gt;</code>	<i>Выделение курсивом</i>
<code>&lt;B&gt;&lt;I&gt;</code> Выделение жирным и курсивом <code>&lt;/I&gt;&lt;/B&gt;</code>	<b><i>Выделение жирным и курсивом</i></b>

С помощью контейнера `<font></font>` и его атрибутов можно задать параметры форматирования шрифта любого фрагмента текста. Атрибут `FACE` позволяет задать гарнитуру шрифта (например, `FACE="Arial"`), атрибут `SIZE` — размер шрифта (например, `SIZE="4"`), атрибут `COLOR` — цвет шрифта (например, `COLOR="blue"`). Все эти атрибуты и тэги можно совмещать в любом порядке, например:

```

<p align="center">
<font color="green" size="3" face="Arial">
<b>Пример отформатированного текста</b>

```

`</font>`

Ещё одним элементом текста является заголовок. Заголовок заключается в тэги `<H></H>`. После буквы H в тэге заголовка следует указать размер заголовка – цифру от 1 до 6. Заголовок можно выровнять по левому, правому краю или по центру точно так же, как и абзац, т.е. при помощи атрибута `align=""`, например:

```
<h1 align="center">Заголовок </h1>
```

Нередко для разделения текста используется горизонтальная линия. Её можно задать с помощью одиночного тэга `<hr>`. На самом деле, тэг `<hr>` рисует не линию, а прямоугольник, просто по умолчанию этот прямоугольник имеет нулевую высоту и длину на весь экран (если использовать его в таблице, то длина будет равна ячейке таблицы). Увидеть, что это прямоугольник можно, если изменить его высоту с помощью атрибута `size="размер"`, где "размер" – это высота прямоугольника в пикселях. Чтобы сделать его непрозрачным, следует указать атрибут `noshade`. Для указания ширины прямоугольника используется атрибут `width="размер"`, который позволяет задать ширину прямоугольника либо в пикселях (как и высоту), либо в процентах от экрана браузера (`width="10%"`). Помимо этого, прямоугольник тоже можно выровнять по любому краю или по центру. Например:

```
<hr size="20" width="100" align="center" noshade >
```

#### *2.4. Таблицы*

Таблицы являются важнейшим элементом HTML-документов. В основном таблицы используются не для представления табличных данных, а для формирования структуры HTML-документов. Практически всегда для построения страницы используются таблицы с невидимыми рамками. Внутри ячеек таблицы может располагаться что угодно – текст, картинки, гиперссылки, другие таблицы. Причём каждой ячейке таблицы можно задать свои свойства – заливка, выравнивание и т.п.

Таблица располагается между тэгами `<table>` и `</table>`. Каждая новая строка таблицы задаётся тэгом `<tr>`, а каждая новая ячейка в строке – тэгом `<td>`. Вот как задается простейшая таблица:

```
<table border="1">
<tr><td>Ячейка 1-1<td>Ячейка 1-2<td>Ячейка 1-3<td>
<tr><td>Ячейка 2-1<td>Ячейка 2-2<td>Ячейка 2-3<td>
<tr><td>Ячейка 3-1<td>Ячейка 3-2<td>Ячейка 3-3<td>
</table>
```

В результате в браузере получится:

Ячейка 1-1	Ячейка 1-2	Ячейка 1-3
Ячейка 2-1	Ячейка 2-2	Ячейка 2-3
Ячейка 3-1	Ячейка 3-2	Ячейка 3-3

В этом примере обратите внимание на тэг `<table border="1">`, в котором указан атрибут таблицы `border="1"`. Этот атрибут задаёт толщину рамки для таблицы. Если указать нулевую толщину, то рамка будет невидимой.

Для выравнивания таблицы служит атрибут `align=""`, который работает точно так же, как при выравнивании абзаца. Для этого используется конструкция `<table align="...">`. Подобным образом можно выровнять содержимое каждой ячейки `<td align="...">`. Данные в ячейках можно выравнивать и по вертикали с помощью атрибута `valign=""`, в кавычках указывается, как это сделать (`bottom` – по нижнему краю ячейки, `middle` – по центру, `top` – по верхнему).

## 2.5. Списки

В HTML есть возможность создания списков. Простейший список – нумерованный. Он заключается между тэгами `<UL>` и `</UL>`, причём каждый пункт списка задаётся тэгом `<LI>`:

```
<UL>
<LI>Пункт 1
<LI>Пункт 2
<LI>Пункт 3
</UL>
```

Существует возможность создания любого числа списков, вложенных в другой список. Для этого в HTML-коде стоит лишь написать внутри одного списка другой, например так:

```
<UL>
<LI>Пункт 1
<LI>Пункт 2
<UL>
<LI>Пункт 2-1
<LI>Пункт 2-2
<LI>Пункт 2-3
</UL>
<LI>Пункт 3
</UL>
```



Также в HTML есть возможность создания списков с различной нумерацией. Нумерованные списки заключаются в тэги <OL> и </OL>. Пункты списка обозначаются <LI>. Для указания типа нумерации к тэгу <OL> добавляется атрибут `Type=""`, где в кавычках может указываться любое из следующих значений: 1, i, I, a, A, которое и указывает на тип нумерации. Тип "1" используется по умолчанию и указывать его необязательно, т.е. тэги <OL> и <OL Type="1"> идентичны. В табл. 2 приводятся примеры всех типов нумерации.

Таблица 2

Написание в HTML	Отображение в браузере
<pre>&lt;OL Type="1"&gt; &lt;LI&gt;Пункт 1 &lt;LI&gt;Пункт 2 &lt;LI&gt;Пункт 3 &lt;/OL&gt;</pre>	1. Пункт 1 2. Пункт 2 3. Пункт 3
<pre>&lt;OL Type="i"&gt; &lt;LI&gt;Пункт 1 &lt;LI&gt;Пункт 2 &lt;LI&gt;Пункт 3 &lt;/OL&gt;</pre>	i. Пункт 1 ii. Пункт 2 iii. Пункт 3
<pre>&lt;OL Type="I"&gt; &lt;LI&gt;Пункт 1 &lt;LI&gt;Пункт 2 &lt;LI&gt;Пункт 3 &lt;/OL&gt;</pre>	I. Пункт 1 II. Пункт 2 III. Пункт 3
<pre>&lt;OL Type="A"&gt; &lt;LI&gt;Пункт 1 &lt;LI&gt;Пункт 2 &lt;LI&gt;Пункт 3 &lt;/OL&gt;</pre>	A. Пункт 1 B. Пункт 2 C. Пункт 3
<pre>&lt;OL Type="a"&gt; &lt;LI&gt;Пункт 1 &lt;LI&gt;Пункт 2 &lt;LI&gt;Пункт 3 &lt;/OL&gt;</pre>	a. Пункт 1 b. Пункт 2 c. Пункт 3

### 2.6. Вставка изображений.

На Web-страницах могут размещаться графические файлы трех форматов – GIF, JPG и PNG. Если рисунок сохранен в другом формате, то его необходимо предварительно преобразовать в один из выше-

указанных форматов с помощью графического редактора. Для этих целей можно использовать редактор Photo Editor, входящий в пакет Microsoft Office.

Для вставки изображения используется тэг <IMG> с атрибутом SRC="name", значение которого указывает на место хранения файла на локальном компьютере или в Интернете. Если графический файл находится на локальном компьютере в том же каталоге, что и файл Web-страницы, то в качестве значения атрибута SRC достаточно указать только имя файла. Если файл находится в другом каталоге на данном локальном компьютере, то значением атрибута должно быть полное имя файла. Если файл находится на удаленном сервере в Интернете, то должен быть указан URL-адрес этого файла.

Расположить рисунок относительно текста различными способами позволяет атрибут ALIGN тэга <IMG>, который может принимать пять различных значений: TOP (верх), MIDDLE (середина), BOTTOM (низ), LEFT (слева) и RIGHT (справа).

Иллюстрации на Web-страницах стали неотъемлемым элементом дизайна. Однако иногда пользователи в целях экономии времени отключают в браузере загрузку графических изображений и читают только тексты. Поэтому, чтобы не терялись смысл и функциональность страницы, вместо рисунка с помощью атрибута ALT выводится текст, поясняющий, что должен был бы увидеть пользователь на рисунке. Например: <IMG SRC="computer.gif" ALT="Компьютер" ALIGN="right >.

## 2.7. Гиперссылки на Web-страницах

Первая титульная страница должна предоставлять посетителю Web-сайта возможность начать путешествие по сайту. Для этого на титульную страницу должны быть помещены гиперссылки на другие страницы сайта.

Гиперссылка состоит из двух частей: *указателя ссылки* и *адресной части ссылки*. Указатель ссылки — это то, что мы видим на Web-странице (текст или рисунок), обычно выделенный синим цветом и подчеркиванием. Активизация гиперссылки вызывает переход на другую страницу. Адресная часть гиперссылки представляет собой URL-адрес документа, на который указывает ссылка. URL-адрес может быть абсолютным и относительным. Абсолютный URL-адрес документа полностью определяет компьютер, каталог и файл, в котором располагается документ.

Абсолютный адрес документа, находящегося на локальном компьютере, будет включать в себя путь к файлу и имя файла, например: `C:/Web-сайт/filename.htm`. Абсолютный адрес документа, находящегося на удаленном компьютере в Интернете, будет включать имя сервера Интернета, путь к файлу и имя файла, например: `http://www.host.ru/Web-сайт/filename.htm`.

Относительный URL-адрес указывает на местоположение документа относительно того, в котором находится указатель ссылки. При разработке сайта рекомендуется входящие в него Web-страницы связывать относительными ссылками. Это позволит не изменять адресную часть ссылок при перемещении Web-сайта в другой каталог локального компьютера или при его публикации в Интернете.

Для определения адреса перехода по каждой гиперссылке используется контейнер `<A></A>` с атрибутом `HREF`, значением которого является URL-адрес документа на локальном компьютере или в Интернете. Контейнер должен содержать указатель гиперссылки:

```
<A HREF="URL">Указатель гиперссылки</A>.
```

Создать на титульной странице *панель навигации* в виде абзаца, выровненного по центру, в котором указатели гиперссылок разделены пробелами (`&nbsp;`), можно с помощью следующего кода:

```
<P ALIGN="center">
<A HREF="URL 1">Указатель 1</A> &nbsp;
<A HREF="URL 2">Указатель 2</A> &nbsp;
<A HREF="URL 3">Указатель 3</A>
</P>
```

Полезно на титульной странице сайта создать ссылку на адрес электронной почты, по которому посетители могут связаться с администрацией сайта. Для этого необходимо атрибуту ссылки `HREF` присвоить URL-адрес электронной почты и вставить ее в контейнер `<ADDRESS></ADDRESS>`, который задает стиль абзаца, принятый для указания адреса:

```
<ADDRESS>
<A HREF="mailto:mailbox@provaider.ru">E-mail:
mailbox@provaider.ru</A>
</ADDRESS>
```

По щелчку мыши по ссылке на адрес электронной почты будет открываться почтовая программа Outlook Express (или другая используемая по умолчанию почтовая программа), где в строке «Кому» будет указан заданный в ссылке адрес.

## 2.8. Формы на Web-страницах

Для того чтобы посетители сайта могли не только просматривать Web-страницы, но и отправлять информацию администраторам сайта, на его страницах размещают формы. Формы включают в себя управляющие элементы различных типов: текстовые поля, раскрывающиеся списки, флажки, переключатели и так далее. Вся форма заключается в контейнер `<FORM></FORM>`.

Для получения данных о посетителях разместим на форме два однострочных *текстовых поля* для ввода информации. Текстовые поля создаются с помощью тэга `<INPUT>` со значением атрибута `TYPE="text"`. Атрибут `NAME` является обязательным и служит для идентификации полученной информации. Значением атрибута `SIZE` является число, задающее длину поля ввода в символах. Для того чтобы анкета «читалась», необходимо разделить строки с помощью тэга перевода строки `<BR>`.

Создадим в Блокноте файл `anketa.htm` и добавим HTML-код, создающий текстовые поля для ввода данных:

```
<FORM>
Пожалуйста, введите ваше имя: <BR>
<INPUT TYPE="text"
NAME="name" SIZE=30> <BR>
E-mail: <BR>
<INPUT TYPE="text"
NAME="e-mail" SIZE=30> <BR>
</FORM>
```

Для создания группы *переключателей* («*радиокнопок*») используется тэг `<INPUT>` со значением атрибута `TYPE="radio"`. Все элементы в группе должны иметь одинаковые значения атрибута `NAME`. Например, `NAME="group"`. Еще одним обязательным атрибутом является `VALUE`, которому присвоим значения "schoolboy", "student" и "teacher". Значение атрибута `VALUE` должно быть уникальным для каждой «радиокнопки», так как при ее выборе именно они передаются серверу. Атрибут `CHECKED`, который задает выбор кнопки по умолчанию, в данном случае не используется.

Добавим HTML-код, создающий группу переключателей для выбора одного варианта:

```
<BR>
<INPUT TYPE="radio"
```

```
NAME="group" VALUE="schoolboy">учащийся<BR>
<INPUT TYPE="radio"
NAME="group" VALUE="student">студент<BR>
<INPUT TYPE="radio"
NAME="group" VALUE="teacher">преподаватель<BR>
```

Пусть, например, мы хотим узнать, какими сервисами Интернета посетитель сайта пользуется наиболее часто. Здесь из предложенного перечня он может выбрать одновременно несколько вариантов, пометив их *флажками*. Флажки создаются в тэге <INPUT> со значением атрибута TYPE="checkbox". Флажки, объединенные в группу, могут иметь одинаковые значения атрибута NAME. Например, NAME="group".

Еще одним обязательным атрибутом является VALUE, которому присвоим значения "www", "e-mail" и "ftp". Значение атрибута VALUE должно быть уникальным для каждого флажка, так как при его выборе именно они передаются серверу.

Добавим HTML-код, создающий флажки для выбора нескольких вариантов:

```
Какие из сервисов Интернета вы используете наиболее часто:
<BR>
<INPUT TYPE="checkbox" NAME="group" VALUE="www"> WWW <BR>
<INPUT TYPE="checkbox" NAME="group" VALUE="e-mail"> e-mail <BR>
<INPUT TYPE="checkbox" NAME="group" VALUE="ftp"> FTP <BR>
```

Теперь выясним, какой из браузеров предпочитает посетитель сайта. Перечень браузеров представим в виде раскрывающегося *списка*, из которого можно выбрать только один вариант. Для реализации такого списка используется контейнер <SELECT></SELECT>, в котором каждый элемент списка определяется тэгом <OPTION>. Выбираемый по умолчанию элемент задается с помощью атрибута SELECTED:

```
Какой браузер вы предпочитаете:
<BR>
<SELECT NAME="browsers">
<OPTION SELECTED> Internet Explorer
<OPTION> Netscape Navigator
<OPTION> Opera
<OPTION> Neo Planet
</SELECT>
```

В заключение поинтересуемся, что хотел бы видеть посетитель на наших страницах, какую информацию следовало бы в них добавить. Так как мы не можем знать заранее, насколько обширным будет ответ читателя, отведем для него *текстовую область* с линейкой прокрутки. Создается такая область с помощью тэга <TEXTAREA> со следующими обязательными атрибутами: NAME, задающим имя области; ROWS, определяющим число строк; COLS – число столбцов области. Добавим HTML-код, создающий текстовую область для ввода комментариев:

```
Какую еще информацию вы хотели бы видеть на нашем сайте?  
<BR>  
<TEXTAREA NAME="resume" ROWS=4 COLS=30>  
</TEXTAREA>  
<BR>
```

Отправка введенной в форму информации или очистка полей от уже введенной информации осуществляется с помощью кнопок. Кнопки создаются с помощью тэга <INPUT>. Для создания кнопки, которая отправляет информацию, атрибуту TYPE необходимо присвоить значение "submit", а атрибуту VALUE, который задает надпись на кнопке, — значение "Отправить". Для создания кнопки, которая производит очистку формы, атрибуту TYPE необходимо присвоить значение "reset", а атрибуту VALUE – значение "Очистить". Добавим HTML-код, создающий кнопки:

```
<INPUT TYPE="submit" VALUE="Отправить">  
<INPUT TYPE="reset" VALUE="Очистить">
```

Заполненная форма отправляется на сервер, где обрабатывается специальной программой – CGI-скриптом, или по электронной почте автору сайта, где он уже самостоятельно обрабатывает полученные данные.

Для того чтобы при щелчке по кнопке «Отправить» данные из формы передавались на сервер и там обрабатывались, необходимо указать, куда их отправить и какая программа будет их обрабатывать. Эти параметры задаются с помощью атрибута ACTION контейнера <FORM>:

```
<FORM ACTION="URL"></FORM>
```

Атрибут ACTION определяет URL-адрес программы, расположенной на Web-сервере, которая обрабатывает полученные данные из формы. Пусть программой, которая заносит данные из формы в базу данных, будет программа bd.exe. Обычно такие программы хранятся в каталоге cgi-bin. Тогда атрибут ACTION будет выглядеть следующим образом:

```
ACTION="http://www.mycompany.ru/cgi-bin/bd.exe"
```

## 2.9. Описание основных тэгов HTML

В табл. 3 приведено краткое описание основных тэгов HTML.

Таблица 3

Тег	Функция
<p>Структура Web-страницы:</p> <pre>&lt;html&gt; &lt;/html&gt;</pre> <pre>&lt;head&gt; &lt;/head&gt;</pre> <pre>&lt;title&gt; &lt;/title&gt;</pre> <pre>&lt;body&gt; &lt;/body&gt;</pre>	<p>Указывает программе просмотра страниц, что это HTML-документ</p> <p>Определяет место, где помещается различная информация, не отображаемая в теле документа. Здесь располагается тег названия документа и теги для поисковых машин</p> <p>Помещает название документа в оглавление программы просмотра страниц</p> <p>Определяет видимую часть документа</p>
<p>Атрибуты тела документа:</p> <pre>&lt;body background="name"&gt;</pre> <pre>&lt;body bgcolor="*"&gt;*</pre> <pre>&lt;body text="*"&gt;</pre> <pre>&lt;body link="*"&gt;</pre> <pre>&lt;body vlink="*"&gt;</pre> <pre>&lt;body alink="*"&gt;</pre>	<p>Устанавливает фоновое изображение</p> <p>Устанавливает цвет фона документа с помощью названия цвета или его 16-ричного кода RRGGBB (пример: black или #000000 – черный; red или #FF0000 – красный; green или #00FF00 – зеленый; blue или #0000FF – синий; grey или #333333 – серый)</p> <p>Устанавливает цвет текста документа</p> <p>Устанавливает цвет гиперссылок</p> <p>Устанавливает цвет пройденной гиперссылки</p> <p>Устанавливает цвет активной гиперссылки</p>
<p>Форматирование текста:</p> <pre>&lt;pre&gt; &lt;/pre&gt;</pre> <pre>&lt;h* &gt; &lt;/h* &gt;</pre> <pre>&lt;h* align="*" &gt; &lt;/h* &gt;</pre> <pre>&lt;p &gt; &lt;/p &gt;</pre> <pre>&lt;p align="*" &gt; &lt;/p &gt;</pre> <pre>&lt;br &gt;</pre> <pre>&lt;blockquote &gt; &lt;/blockquote &gt;</pre> <pre>&lt;b &gt; &lt;/b &gt;</pre> <pre>&lt;i &gt; &lt;/i &gt;</pre> <pre>&lt;tt &gt; &lt;/tt &gt;</pre> <pre>&lt;cite &gt; &lt;/cite &gt;</pre> <pre>&lt;em &gt; &lt;/em &gt;</pre>	<p>Обрамляет предварительно отформатированный текст</p> <p>Заголовок определенного размера (уровни от 1 до 6 по убыванию)</p> <p>Заголовок определенного размера с выравниванием относительно сторон документа (left – влево; center – по центру; right – вправо)</p> <p>Создает новый абзац</p> <p>Абзац с выравниванием left, right, или center</p> <p>Вставляет перевод строки</p> <p>Создает отступы с обеих сторон текста</p> <p>Создает жирный текст</p> <p>Создает наклонный текст</p> <p>Создает текст, имитирующий стиль печатной машинки</p> <p>Используется для цитат, обычно наклонный текст</p> <p>Используется для выделения из текста слова (наклонный или жирный текст)</p>

\* Здесь и далее звездочка обозначает одно из допустимых значений атрибута.

Тег	Функция
<p><code>&lt;strong&gt; &lt;/strong&gt;</code></p> <p><code>&lt;font size=* &lt;/font&gt;</code>  <code>&lt;font color="*" &lt;/font&gt;</code>  <code>&lt;font face="*" &lt;/font&gt;</code></p> <p><code>&lt;sub&gt; &lt;/sub&gt;</code>  <code>&lt;sup&gt; &lt;/sup&gt;</code></p>	<p>Используется для выделения наиболее важных частей текста (наклонный или жирный текст)</p> <p>Устанавливает размер шрифта в пределах от 1 до 7</p> <p>Устанавливает цвет шрифта</p> <p>Устанавливает гарнитуру шрифта (Arial, Times New Roman и др.)</p> <p>Верхний индекс</p> <p>Нижний индекс</p>
<p>Графические элементы:</p> <p><code>&lt;img src="name"&gt;</code>  <code>&lt;img src="name" align="*"&gt;</code></p> <p><code>&lt;img src="name" alt="*"&gt;</code>  <code>&lt;img src="name" border=* &gt;</code>  <code>&lt;hr&gt;</code>  <code>&lt;hr size=* &gt;</code>  <code>&lt;hr width=? &gt;</code></p> <p><code>&lt;hr noshade &gt;</code>  <code>&lt;hr color="*" &gt;</code></p>	<p>Добавляет изображение в HTML-документ</p> <p>Выравнивает изображение относительно сторон документа, принимает значения: left, right, center; bottom, top, middle</p> <p>Вывод текста вместо изображения</p> <p>Устанавливает толщину рамки вокруг изображения</p> <p>Добавляет горизонтальную линию</p> <p>Устанавливает высоту (толщину) линии</p> <p>Устанавливает ширину линии, можно указать ширину в пикселях или процентах</p> <p>Создает линию без тени</p> <p>Задаст линии определенный цвет</p>
<p>Гиперссылки:</p> <p><code>&lt;a name="*" &gt; &lt;/a&gt;</code></p> <p><code>&lt;a href=#* &gt; &lt;/a&gt;</code> указатель ссылки</p> <p><code>&lt;a href="URL#*" &gt; &lt;/a&gt;</code> указатель ссылки</p> <p><code>&lt;a href="URL" &gt; &lt;/a&gt;</code> указатель ссылки</p> <p><code>&lt;a href="mailto:EMAIL" &gt; &lt;/a&gt;</code></p>	<p>Определяет закладку (отмечает часть текста как цель для гиперссылок в документе)</p> <p>Гиперссылка на закладку в текущем документе</p> <p>Гиперссылка на закладку в другом документе</p> <p>Гиперссылка на другую страницу</p> <p>Создает гиперссылку вызова почтовой программы для написания письма автору документа</p>
<p>Списки:</p> <p><code>&lt;dl&gt; &lt;/dl&gt;</code></p> <p><code>&lt;dt&gt;</code> термин</p> <p><code>&lt;dd&gt;</code> определение</p> <p><code>&lt;ol&gt; &lt;li&gt; &lt;/ol&gt;</code>  <code>&lt;li&gt;</code></p> <p><code>&lt;ol type="*" &gt;</code></p> <p><code>&lt;ol start=* &gt;</code></p> <p><code>&lt;ul&gt; &lt;li&gt; &lt;/ul&gt;</code>  <code>&lt;li&gt;</code></p> <p><code>&lt;ul type="*" &gt;</code></p>	<p>Создает список определений</p> <p>Определяет каждый из терминов списка</p> <p>Описывает каждое определение</p> <p>Создает нумерованный список</p> <p>Определяет каждый элемент списка</p> <p>Определяет тип нумерации (A, a, I, i, 1)</p> <p>Определяет первый номер списка (1, 2, ...)</p> <p>Создает ненумерованный список</p> <p>Определяет тип метки (disk, circle, square)</p>



Тег	Функция
<p><code>&lt;div align=*&gt;</code></p> <p><code>&lt;menu&gt; &lt;li&gt; &lt;/menu&gt;</code></p> <p><code>&lt;dir&gt; &lt;li&gt; &lt;/dir&gt;</code></p>	<p>Важный тег, используемый для форматирования больших блоков текста HTML-документа, также используется в таблицах стилей</p> <p>Создает меню</p> <p>Создает каталог</p>
<p>Таблицы:</p> <p><code>&lt;table&gt; &lt;/table&gt;</code></p> <p><code>&lt;tr&gt; &lt;/tr&gt;</code></p> <p><code>&lt;td&gt; &lt;/td&gt;</code></p> <p><code>&lt;th&gt; &lt;/th&gt;</code></p>	<p>Создает таблицу</p> <p>Определяет строку в таблице</p> <p>Определяет отдельную ячейку в таблице</p> <p>Определяет заголовок таблицы (нормальная ячейка с отцентрованным жирным текстом)</p>
<p>Атрибуты таблицы:</p> <p><code>&lt;table border=*&gt;</code></p> <p><code>&lt;table cellspacing=*&gt;</code></p> <p><code>&lt;table cellpadding=*&gt;</code></p> <p><code>&lt;table width=*&gt;</code></p> <p><code>&lt;tr align=*&gt;</code> или</p> <p><code>&lt;td align=*&gt;</code></p> <p><code>&lt;tr valign=*&gt;</code> или</p> <p><code>&lt;td valign=*&gt;</code></p> <p><code>&lt;td colspan=*&gt;</code></p> <p><code>&lt;td rowspan=*&gt;</code></p> <p><code>&lt;td nowrap&gt;</code></p>	<p>Задаёт толщину рамки таблицы</p> <p>Задаёт расстояние между ячейками таблицы</p> <p>Задаёт расстояние между содержимым ячейки и ее рамкой</p> <p>Устанавливает ширину таблицы в пикселах или процентах от ширины документа</p> <p>Устанавливает выравнивание ячеек в таблице, принимает значения: left, center или right</p> <p>Устанавливает вертикальное выравнивание для ячеек таблицы, принимает значения: top, middle или bottom</p> <p>Указывает количество столбцов, которое объединено в одной ячейке (по умолчанию=1)</p> <p>Указывает количество строк, которое объединено в одной ячейке (по умолчанию=1)</p> <p>Не позволяет программе просмотра делать перевод строки в ячейке таблицы</p>
<p>Кадры:</p> <p><code>&lt;frameset&gt; &lt;/frameset&gt;</code></p> <p><code>&lt;frameset rows="value,value"&gt;</code></p> <p><code>&lt;frameset cols="value,value"&gt;</code></p> <p><code>&lt;frame&gt;</code></p> <p><code>&lt;noframes&gt; &lt;/noframes&gt;</code></p>	<p>Предваряет тег <code>&lt;body&gt;</code> в документе, содержащем кадры</p> <p>Определяет строки в таблице кадров, высота которых определена количеством пикселей или в процентном соотношении к высоте таблицы кадров</p> <p>Определяет столбцы в таблице кадров, ширина которых определена количеством пикселей или в процентном соотношении к ширине таблицы кадров</p> <p>Определяет единичный кадр или область в таблице кадров</p> <p>Определяет, что будет показано в окне браузера, если он не поддерживает кадры</p>

Тег	Функция
<p>Атрибуты кадров:</p> <p><code>&lt;frame src="URL"&gt;</code></p> <p><code>&lt;frame name="name"&gt;</code></p> <p><code>&lt;frame marginwidth=*&gt;</code></p> <p><code>&lt;frame marginheight=*&gt;</code></p> <p><code>&lt;frame scrolling=VALUE&gt;</code></p> <p><code>&lt;frame noresize&gt;</code></p>	<p>Определяет, какой из HTML-документов будет показан в кадре</p> <p>Указывает имя кадра или области, что позволяет перенаправлять информацию в этот кадр или область из других кадров</p> <p>Определяет величину отступов по левому и правому краям кадра; должно быть равно или больше 1</p> <p>Определяет величину отступов по верхнему и нижнему краям кадра; должно быть равно или больше 1</p> <p>Указывает, будет ли выводиться линейка прокрутки в кадре; значение value может быть "yes", "no" или "auto". Значение по умолчанию для обычных документов – auto</p> <p>Препятствует изменению размеров кадра</p>
<p>Формы<sup>2</sup>:</p> <p><code>&lt;form&gt; &lt;/form&gt;</code></p> <p><code>&lt;select multiple name="*" size=*&gt;&lt;/select&gt;</code></p> <p><code>&lt;option&gt;</code></p> <p><code>&lt;select name="*"&gt;</code></p> <p><code>&lt;/select&gt;</code></p> <p><code>&lt;textarea name="*" cols=40 rows=8&gt; &lt;/textarea&gt;</code></p> <p><code>&lt;input type="checkbox" name="*"&gt;</code></p> <p><code>&lt;input type="radio" name="*" value="x"&gt;</code></p> <p><code>&lt;input type="text" name="foo" size=20&gt;</code></p> <p><code>&lt;input type="submit" value="*"&gt;</code></p> <p><code>&lt;input type="image" border=0 name="*" src="*.gif"&gt;</code></p> <p><code>&lt;input type="reset"&gt;</code></p>	<p>Создает формы</p> <p>Создает прокручиваемое меню. Size устанавливает количество пунктов меню, которое будет показано на экране, остальные будут доступны при использовании прокрутки</p> <p>Указывает каждый отдельный элемент меню</p> <p>Создает ниспадающее меню</p> <p>Создает окно для ввода текста. Columns указывает ширину окна, rows – его высоту</p> <p>Создает checkbox. За тегом следует текст</p> <p>Создает radio-кнопку. За тегом следует текст</p> <p>Создает строку для ввода текста. Параметром Size указывается длина в символах</p> <p>Создает кнопку "Принять"</p> <p>Создает кнопку "Принять" – для этого используется изображение</p> <p>Создает кнопку "Отмена"</p>

<sup>2</sup> Для форм, выполняющих какие-то функции, должны быть запущены соответствующие CGI-скрипты на сервере. HTML только создает внешний интерфейс формы.

### **3. Задание к работе**

Разработать Web-сайт, состоящий из нескольких страниц на самостоятельно выбранную тему или в соответствии с предложенными вариантами.

*Вариант 1.* Личный сайт.

*Вариант 2.* Сайт по современным информационным технологиям.

*Вариант 3.* Сайт по истории вычислительной техники.

*Вариант 4.* Сайт, посвященный писателю, исполнителю, музыкальной группе, спортсмену и т.д.

*Вариант 5.* Учебный сайт, электронный учебник.

*Вариант 6.* Сайт предприятия, организации.

### **4. Содержание отчета**

1. Цель работы.
2. Вариант индивидуального задания.
3. Описание предметной области.
4. Структура сайта.
5. Тексты HTML-кода.
6. Выводы по работе.

### **5. Контрольные вопросы**

1. Какие базовые информационные процессы реализуются в Web-технологии?
2. В каких прикладных информационных технологиях могут применяться Web-технологии?
3. Что является основными достоинствами HTML-документов?
4. Назовите инструментальные средства создания Web-страниц.
5. Какие объекты могут размещаться на Web-страницах?
6. Перечислите основные тэги.
7. Перечислите теги для форматирования текста.
8. Поясните создание таблиц, списков.
9. Как вставить графические элементы?
10. Что обеспечивает интерактивность сайтов?

## Лабораторная работа № 3

### СОЗДАНИЕ И РАБОТА С РЕЛЯЦИОННОЙ БАЗОЙ ДАННЫХ

#### 1. Цель работы

Рассмотреть области применения компьютерных баз данных; изучить и на практике освоить функции и возможности СУБД MS Access; разработать проект реляционной базы данных, выполнить ее нормализацию; создать средствами СУБД MS Access основные объекты базы данных в соответствии с разработанным проектом.

#### 2. Основные сведения

Развитие информационных технологий привело к созданию компьютерных баз данных (БД). Создание баз данных, а также операции поиска и сортировки данных выполняются специальными программами – *системами управления базами данных* (СУБД). Настольной системой управления базами данных является приложение Access, входящее в пакет Microsoft Office.

В Access используется стандартный для среды Windows и Office многооконный интерфейс, но в отличие от других приложений, не многодокументный. Единственно может быть открыта только одна база данных, содержащая обязательное *окно базы данных* и *окна для работы с объектами базы данных*. В каждый момент времени одно из окон является активным, и в нем курсором отмечается активный объект.

Окно базы данных – один из главных элементов интерфейса Access. Здесь систематизированы все объекты БД: таблицы, запросы, формы, отчеты, макросы и модули. В базах данных вся информация хранится в *двумерных таблицах*. Это основной объект БД, все остальные объекты создаются на основе существующих таблиц. Главное предназначение *запросов* — отбор данных на основании заданных условий. *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 редактировать или удалять существующие. Форма может содержать рисунки, графики и другие внедренные объекты. *Отчеты* предназначены для печати данных, содержащихся в таблицах и запросах, в красиво оформленном виде. *Макросы* служат для автоматизации повторяющихся операций. *Модули* также служат для автоматизации работы с БД.

Модули еще называются *процедурами обработки событий* и пишутся на языке *Visual Basic for Application (VBA)*.

## 2.1. Создание базы данных

Создать в приложении Access новую базу данных с помощью команды *Файл/Создать базу данных...* и присвоить ей имя, например, *Контакты*.

### 2.1.1. Создание таблиц

Прежде всего необходимо определить *структуру базы данных*, т.е. количество полей, их названия и тип данных, в них хранящихся. После чего в окне *Контакты: база данных* выбрать группу объектов *Таблицы* и создать основной объект БД – таблицу. Сделать это можно одним из трех способов:

1) *Создание таблицы в режиме мастера* на основе стандартных шаблонов;

2) *Создание таблицы путем ввода данных*. Access откроет в режиме таблицы пустой бланк, в котором необходимо переименовать нужное количество столбцов и выполнить ввод данных хотя бы в одну *запись* (строку). Access анализирует записи и на основании анализа определяет типы полей для новой таблицы;

3) *Создание таблицы в режиме конструктора*. Для каждого поля нужно задать имя, тип данных и определить его свойства. Кроме того, необходимо задать ключевое поле.

С помощью мастера, используя в качестве образца таблицу из категории *Деловые*, создадим одноименную с БД таблицу *Контакты*, включив в нее поля *КодКонтакта* (содержанием этого поля может быть порядковый, не повторяющийся ни для какой пары записей, номер делового партнера), *Фамилия*, *Имя*, *Адрес*, *НазваниеКомпании*, *РабочийТелефон*.

С помощью конструктора добавим в базу данных еще одну таблицу – *Список*, предназначенную для записи всех случаев контакта с людьми из первой таблицы. Щелчком на кнопке *Окно базы данных* переключитесь на список таблиц базы данных и выберите команду *Создание таблицы в режиме конструктора*. Окно конструктора состоит из двух частей. В верхней части отображается список полей таблицы, а в нижней – свойства выделенного поля. Определите структуру таблицы *Список* в соответствии с рис. 2.

Поле с именем КодКонтакта является ссылкой на одну из записей таблицы Контакты. Эта ссылка будет иметь числовой тип и содержать число, равное значению поля КодКонтакта соответствующей записи таблицы Контакты, поэтому в новой таблице не нужно заново вводить имя и фамилию.

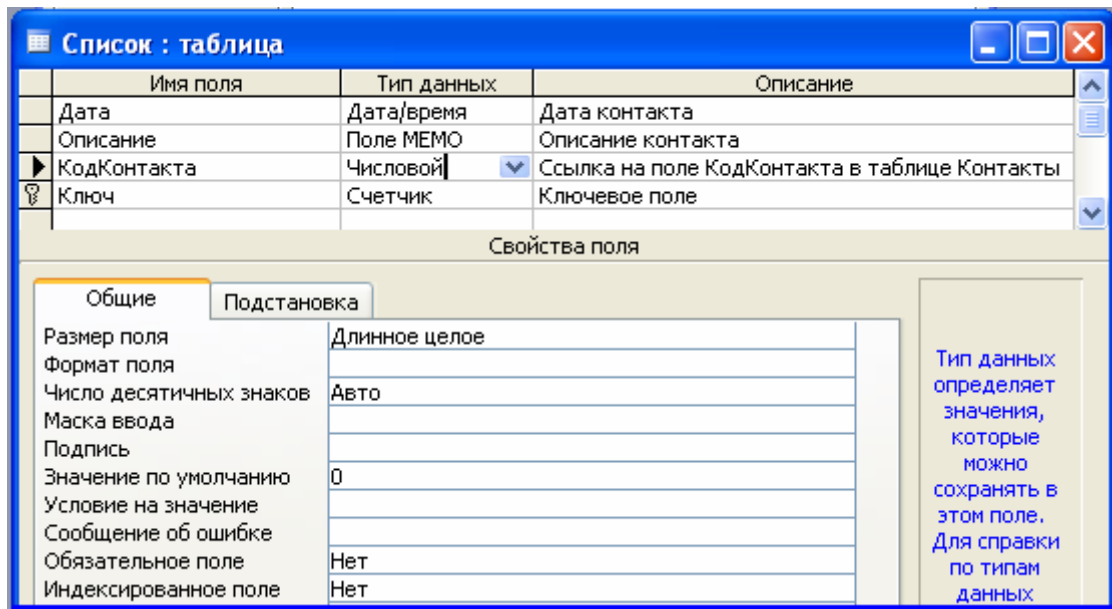


Рис. 2.

Поле Ключ – это индексированное поле, или *индекс*. Такие поля отличаются от обычных тем, что для них Access создает специальные списки, позволяющие выполнять быструю сортировку и поиск по содержимому индексированного поля. Таблица может содержать несколько индексов. Сохраняя активной строку Ключ, щелкните на кнопке Ключевое поле (Primary Key). Напротив этой строки появится значок ключа, показывающий особый статус поля Ключ – это специальный индекс, идентифицирующий записи. Значения такого поля должны быть уникальными. Его наличие необязательно, но оно так часто используется, что при отсутствии ключа Access напоминает об этом и предлагает автоматически добавить ключевое поле. Тип Счетчик избавляет пользователя от необходимости ввода значений ключевого поля и инициирует автоматическую генерацию уникальных значений.

Закройте окно конструктора, сохранив структуру таблицы Список.

### 2.1.2. Параметры поля

Чтобы обеспечить возможность хранения в базе данных разнообразной информации, Access предлагает большой набор типов данных (табл. 4).

Таблица 4.

Тип	Назначение
Текстовый (Text)	Текст длиной до 255 символов
Поле МЕМО (Memo)	Текст без ограничения длины
Числовой (Number)	Числа различных форматов
Дата/время (Date/Time)	Дата и/или время
Денежный (Currency)	Денежные значения различных форматов
Счетчик (AutoNumber)	Счетчик, который автоматически увеличивается на единицу с добавлением каждой новой записи
Логический (Yes/No)	Величины, способные принимать только два значения да/нет или 1/0
Поле объекта OLE (OLE Object)	Поля, позволяющие вставлять рисунки, звуки и данные других типов
Гиперссылка (Hyperlink)	Ссылки, дающие возможность открывать объект Access (таблицу, форму, запрос и т. п.), файл другого приложения или Web-страницу

Откройте таблицу Список в режиме конструктора, щелкнув на кнопке Конструктор. Кроме типа данных и формата отображения вкладка Общие окна конструктора позволяет настроить некоторые дополнительные параметры поля, влияющие на способ ввода данных. Параметр Значение по умолчанию задает содержимое поля, которое записывается в новую запись в том случае, если пользователь не ввел в это поле никаких данных. Рассмотрим поле Дата. Если регулярно обновлять список контактов, то дата контакта чаще всего будет совпадать с текущей датой. Чтобы упростить ввод информации, можно задать текущую дату в качестве значения по умолчанию для поля Дата. Введите в поле Значение по умолчанию функцию Date(), которая возвращает текущую дату.

В некоторых случаях требуется не только задать тип данных поля, но и ограничить его возможные значения некоторым интервалом или более сложным правилом отбора. Такое правило вводится в поле Условие на значение. Предположим, что все контакты, которые будут заноситься в список, состоятся после 1 января 2006 г. Чтобы избежать ошибок, можно заставить Access сообщать о вводе неверной даты. Для этого в поле Усло-

вие на значение щелкните на кнопке построителя выражений. Откроется соответствующее окно диалога. Для задания условий или значений определенных параметров Построитель выражений Access предлагает большой набор математических операций и функций. Верхнее прокручивающееся поле построителя содержит конструируемое выражение. Его можно изменять как путем непосредственного редактирования текста, так и с помощью других кнопок и списков построителя. Необходимо, чтобы дата записи была позднее, то есть больше даты 1 января 2006 г. Поэтому щелкните на кнопке >, чтобы добавить соответствующее условие. Выполните в нижней части диалогового окна выбор нужной функции: **Функции / Встроенные функции / Дата/время / DateValue** и добавьте ее в поле выражения. Заполните местозаполнитель аргумента «stringexpr» конкретной датой в строковом формате, например, “1.1.2006”.

В поле Сообщение об ошибке введите текст **Неверная дата**, который будет появляться при вводе даты, более ранней, чем 1 января 2006 г. (рис. 3).

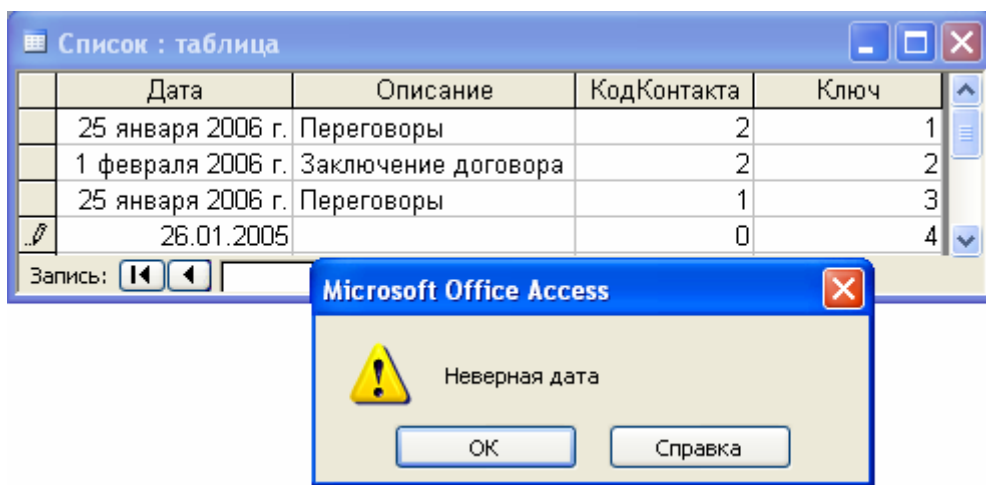


Рис. 3

### 2.1.3. Индекс

Для ускорения поиска данных о конкретном человеке в таблице Контакты удобно рассортировать записи по алфавиту фамилий. Открыв таблицу Контакты в режиме редактирования и выделив столбец Фамилия, щелчком по кнопке **Сортировка по возрастанию** упорядочите записи. Чтобы выполнить сортировку сразу по нескольким полям, следует воспользовать-



ся сложным индексом. Переключитесь в режим конструктора, откройте окно индексов таблицы Контакты, щелкнув на кнопке Индексы панели инструментов. Мастер таблиц при создании таблицы Контакты индексировал ее по трем полям: НазваниеКомпании, Фамилия, КодКонтакта, добавив индексы с именами, соответственно: НазваниеКомпании, Фамилия и PrimaryKey. Последний из трех индексов является ключом таблицы, однозначно идентифицирующим ее записи. Все имеющиеся индексы простые, то есть назначают сортировку только по одному полю. Чтобы записи с одинаковыми фамилиями располагались по алфавиту имен, следует модифицировать индекс Фамилия, добавив в него условие сортировки по полю Имя. Вставив при необходимости пустую строку после строки индекса Фамилия, в новой строке щелкните на кнопке раскрывающегося списка ячейки Имя поля и выберите пункт Имя. В столбце Порядок сортировки автоматически появится вариант По возрастанию. Результат изменения показан на рис. 4.

	Код контакта	Фамилия	Имя	Адрес	Название	Рабочий тел
+	4	Андреев	Николай	Гагарина, 3	ООО "Сапфир"	337523
+	6	Иванов	Антон	Токарева, 3-25	ИП Иванов	336387
+	1	Иванов	Иван	Горького, 47	ООО "Спутник"	325742
+	7	Петров	Андрей	Семашко, 21	ООО "Селенга"	346321
+	5	Петров	Василий	Зеленая, 17	Виконт	
+	3	Петров	Игорь	Студенческая, 4б	Тритон	326597
▶	2	Петров	Николай	Юбилейная, 53-32	ИП Николаев А.П.	315923

Запись: 7 из 7

Рис. 4

## 2.2. Ввод и редактирование данных

### 2.2.1. Ввод и редактирование данных в режиме Таблица

Ввод данных в таблицу базы данных и их редактирование в этом случае мало чем отличается от аналогичных действий в других офисных приложениях. При вводе данных в режиме Таблица в поле маркера записи, которое расположено слева от полей таблицы, может отображаться один из следующих символов:

- \* (звездочка) – отмечает пустую запись в конце таблицы;
- ▶ (стрелка) – отмечает выделенную (активную) запись;

..  (карандаш) – отмечает редактируемую запись.

Перемещение между записями можно осуществлять с помощью мыши, клавиш управления курсором или полосы прокрутки. Для быстрого перемещения между записями в базе данных можно использовать кнопки перемещения на панели Запись, которая находится в нижней части окна таблицы.

### 2.2.2. Список подстановки

При заполнении таблицы Список в режиме просмотра данных не забывайте, что числа в столбце КодКонтакта должны соответствовать номерам записей таблицы Контакты, описывающих людей, с которыми вы контактировали. Искать соответствующие номера очень неудобно. Access позволяет автоматизировать эту операцию с помощью списка подстановки.

Откройте окно конструктора таблицы Контакты. В списке типа данных для поля КодКонтакта выберите пункт Мастер подстановок. В первом окне Мастера подстановок оставьте выбранным положение переключателя Объект «столбец подстановки» будет использовать значения из таблицы или запроса и щелкните на кнопке Далее. Во втором окне Мастера подстановок нужно выбрать таблицу или запрос на роль источника подстановки – таблицу Контакты. Третье окно мастера предлагает выбрать из таблицы Контакты поля, участвующие в подстановке. Этот список обязан содержать то поле, значения которого должны помещаться в поле КодКонтакта, то есть одноименное поле таблицы Контакты. В список источника подстановки следует также добавить те поля, содержимое которых должно отображаться вместо численного значения, помещаемого из поля КодКонтакта таблицы Контакты в поле КодКонтакта таблицы Список. Будет удобно выбирать людей по их именам и фамилиям. Добавьте в список Выбранные поля пункты Фамилия и Имя. В следующем окне мастер продемонстрирует два столбца таблицы-источника, скрыв столбец КодКонтакта. Щелкните на кнопке Далее. В поле последнего столбца введите название Контакт, которое заменит имя столбца КодКонтакта, и щелкните на кнопке Готово.

Сохраните изменения и переключитесь в режим ввода данных. Теперь в третьем столбце таблицы вместо чисел видны фамилии, соответствующие этим числам, хотя само значение поля КодКонтакта не изменилось. При добавлении в таблицу Список новых записей вы можете выби-

рать людей из списка подстановки, сформированного на базе полей Фамилия и Имя таблицы Контакты (рис. 5).

Дата	Описание	Контакт	Ключ
25 января 2006 г.	Переговоры	Петров	1
1 февраля 2006 г.	Заключение договора	Петров	2
25 января 2006 г.	Переговоры	Иванов	3
26 января 2006 г.	Согласование заказа	Андреев	4
26 января 2006 г.	Переговоры	Андреев	5

Запись: 5 из 5

Иванов	Николай
Иванов	Антон
Петров	Иван
Петров	Андрей
Петров	Василий
Петров	Игорь
Петров	Николай

Рис. 5

### 2.2.3. Использование формы

Формы облегчают ввод, редактирование и восприятие информации, могут содержать вспомогательные подписи и элементы оформления. Обычно на форме размещаются надписи, являющиеся именами полей БД, и текстовые поля, содержащие данные из БД.

Создание формы можно проводить двумя способами с использованием:

- Конструктора (сложный путь), который позволяет начать создание формы с нуля;
- Мастера форм (более простой путь), который с помощью серии диалоговых панелей помогает пользователю в создании формы.

Недостатком форм, создаваемых Мастером, является то, что они однообразны и не содержат пояснительных надписей. Чтобы приукрасить форму и расположить поля более удобным образом, следует воспользоваться Конструктором, который позволяет передвигать и масштабировать элементы формы, связывать их с источником данных и настраивать любые другие параметры формы.

### 2.3. Связь таблиц.

В п. 2.2.2 с помощью Мастера подстановок была сформирована связь между двумя таблицами. Такая связь позволяет установить правила взаимодействия между таблицами. Организуется связь по команде Сервис/Схема данных перетаскиванием указателя мыши от связываемого поля одной таблицы к нужному полю другой. Для настройки параметров связи дважды щелкните на линии связи. Откроется диалоговое окно Изменение связей. Установка флажка Обеспечение целостности данных предотвращает появление «висящих» записей (записей в подчиненной таблице Список, не имеющих соответствующих записей в главной таблице Контакты). Установка флажка Каскадное обновление связанных полей позволяет при изменении значения в ключевом поле главной таблицы автоматически изменять соответствующее значение в связанных записях. Установка флажка Каскадное удаление связанных записей позволяет при удалении записи в главной таблице автоматически удалять и все связанные с ней записи в подчиненной таблице.

Закройте окно Схема данных, откройте для просмотра таблицу Контакты. В левой части записей расположены небольшие знаки «плюс». Их присутствие говорит о наличии связи ключевого поля таблицы с другой таблицей. По щелчку на знаке «плюс» открывается вложенная таблица, содержащая те записи таблицы Список, значение поля КодКонтакт которых равно величине одноименного поля записи в таблице Контакты.

### 2.4. Обработка данных в БД

#### 2.4.1. Быстрый поиск данных

Access позволяет с помощью команды Правка/Найти.... производить поиск записей, в которых значения определенного поля полностью или частично совпадают с некоторой величиной. В поле Образец необходимо ввести искомый текст, а в поле Совпадение выбрать пункт С любой частью поля.

#### 2.4.2. Поиск данных с помощью фильтров

Гораздо больше возможностей для поиска данных в БД предоставляют *фильтры*, устанавливаемые командой Записи/Фильтр. Фильтры позволяют отбирать записи, которые удовлетворяют заданным условиям. Ус-

ловия отбора записей создаются с использованием операторов сравнения (=, >, < и т. д.). Простые фильтры содержат условие отбора записей только для одного поля. Сложные фильтры (Расширенный фильтр...) содержат несколько условий для различных полей. В результате применения сложного фильтра будут отобраны только те записи, которые удовлетворяют всем условиям одновременно. Можно сказать, что условия в сложных фильтрах связаны между собой операцией логического умножения.

#### 2.4.3. Сортировка данных

*Сортировка* записей производится по какому-либо полю. Значения, содержащиеся в этом поле, располагаются в определенном порядке, который определяется типом поля. Сортировка записей может производиться либо по возрастанию, либо по убыванию значений поля. В процессе сортировки целостность записей сохраняется, то есть они переносятся из одного места таблицы в другое целиком. Сортировка записей базы данных производится командой Записи/Сортировка. Могут реализовываться вложенные сортировки, то есть сортировки, которые последовательно производятся по нескольким полям. После сортировки по первому столбцу производится сортировка по второму столбцу и так далее.

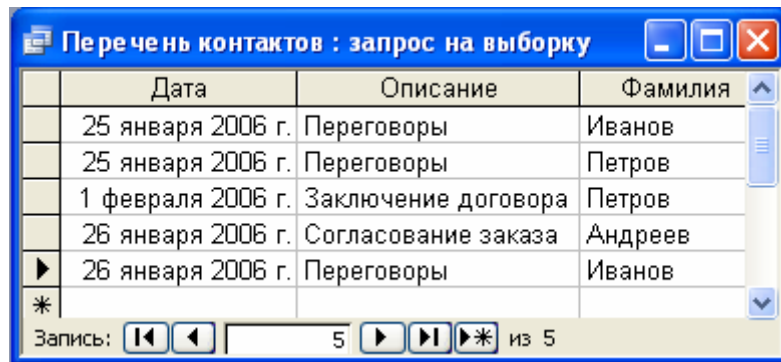
#### 2.4.5. Поиск данных с помощью запросов

*Запросы* осуществляют поиск данных в БД так же, как и фильтры. Различие между ними состоит в том, что запросы – самостоятельные объекты БД, а фильтры привязаны к конкретной таблице. Запрос является производным объектом от таблицы. Запрос – это набор условий, согласно которым производится выборка информации из таблиц. В результате выполнения запроса, адресованного одной или нескольким таблицам, по определенному закону формируется подмножество информационного множества исходных таблиц. Если обрабатываемый объем исходной информации велик, выделение необходимых данных в такое подмножество позволяет существенно сократить время их обработки. В системах типа клиент-сервер, где основные базы данных хранятся на файловом сервере, система запросов позволяет уменьшить объем информации, передаваемой через сеть. Ввиду важности данной темы рассмотрим выборку данных из БД с помощью запросов более подробно.

## 2.5. Запросы

### 2.5.1 Мастер запросов

Мастер запросов позволяет автоматизировать процесс построения запроса. С помощью Мастера выполните выборку информации из таблиц Контакты и Список. В диалоговом окне включите в запрос поля Дата и Описание таблицы Список, а также поле Фамилия таблицы Контакты. Введите имя запроса Перечень контактов. Результат выполнения запроса показан на рис. 6.



	Дата	Описание	Фамилия
	25 января 2006 г.	Переговоры	Иванов
	25 января 2006 г.	Переговоры	Петров
	1 февраля 2006 г.	Заключение договора	Петров
	26 января 2006 г.	Согласование заказа	Андреев
▶	26 января 2006 г.	Переговоры	Иванов
*			

Рис. 6

### 2.5.2. Конструктор запросов

Мастер запросов умеет конструировать только простые условия отбора. Чтобы наложить дополнительные ограничения, следует пользоваться Конструктором запросов, обеспечивающим полное управление параметрами запроса и построение сложных условий отбора данных.

В верхней части окна Конструктора отображаются списки полей таблиц, к которым обращается запрос, и связи между таблицами. Нижняя область содержит бланк выбора полей таблиц, условий отбора и режимов сортировки. Дополните запрос полем Имя таблицы Контакты, перетащив название этого поля в верхнюю ячейку четвертой строки бланка запроса. Назначьте из раскрывающегося списка режим сортировки По возрастанию для полей Фамилия и Имя.

Добавление в запрос Условия отбора позволяет выбирать из таблицы не все записи, а лишь те, которые удовлетворяют определенным критериям. Условия отбора могут содержать константы, встроенные функции, операции сравнения, оператор конъюнкции и дизъюнкции, значение null и not null (для поиска записей с пустыми и непустыми значениями в опреде-

ленном поле). Для создания условий отбора можно использовать Построитель выражений.

*Запрос с параметрами* позволяет пользователю перед выполнением запроса задать условие отбора данных в диалоговом окне. Для этого в конструкторе запросов в качестве условия отбора следует ввести в квадратных скобках текст сообщения для диалогового окна, которое выводится на экран при запуске запроса. Например, [Введите дату контакта:].

### 2.5.3. Итоговый запрос

Запросы позволяют не только выбирать записи из таблиц, но и вычислять различные статистические параметры. Например, можно подсчитать суммарное количество контактов и вывести даты первого и последнего контакта с каждым из людей, включенных в таблицу Контакты. Построим такой запрос в режиме конструктора (рис. 7).

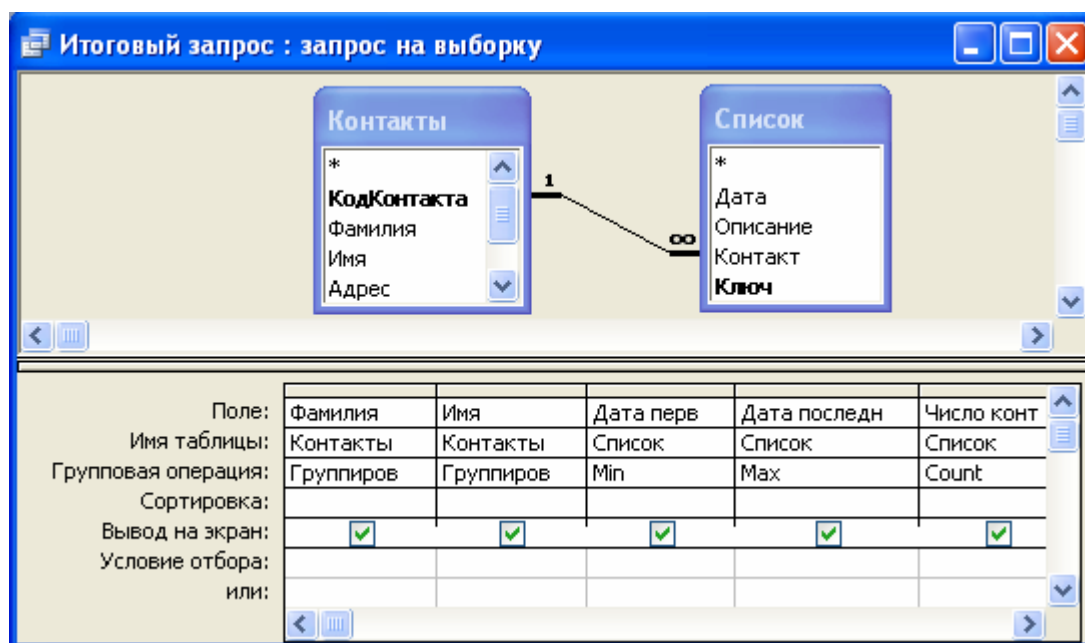


Рис. 7

Добавьте таблицы Контакты и Список в верхнюю область конструктора запроса. Щелкните на кнопке Групповые операции панели инструментов. В бланке запроса появится дополнительная строка Групповая операция, позволяющая выполнять статистические операции со значениями конкретных полей. Перетащите поля Фамилия и Имя таблицы Контакты

соответственно в первый и второй столбцы бланка запроса. В третий, четвертый и пятый столбцы бланка запроса перетащите поле **Дата** таблицы **Список**. В раскрывающемся списке **Групповая операция** третьего столбца бланка запроса выберите пункт **Min**, четвертого – **Max**, пятого – **Count**.

Групповые операции (доступные групповые операции перечислены в табл. 5) построенного запроса обработают все записи таблицы **Список**, соответствующие конкретному человеку из таблицы **Контакты**, и вместо самих данных таблицы **Список** выведут в соответствующее поле результата запроса только значение величины, вычисляемой по определенной формуле.

*Таблица 5*

Операция	Функция
Sum	Сумма значений поля по всем записям
Avg	Среднее значение поля
Min	Минимальное значение
Max	Максимальное значение
Count	Количество записей, соответствующее поле которых не содержит величины Null
StDev	Стандартное отклонение величин поля от среднего
Var	Вариация значений поля.
Группировка (Group By)	Поле, определяющее группу записей, по которой вычисляются статистические параметры. К одной группе относятся все записи, для которых значения поля с режимом Группировка
Выражение	Вычисляемое поле, значение которого рассчитывается по сложной формуле
Условие	Режим задания условия отбора для поля, по которому не выполняется группировка. Access автоматически делает такое поле скрытым

Скорректируем имена столбцов. В ячейке **Поле** третьего столбца замените имя **Дата** на текст **Дата** первого контакта: **Дата**. Правая часть этого выражения, расположенная правее двоеточия, по-прежнему задает имя поля, а левая определяет название столбца результата запроса. Таким образом, любому столбцу запроса можно назначить произвольное имя. В ячейке **Поле** четвертого столбца введите **Дата** последнего контакта: **Дата**. В



первой строке пятого столбца бланка запроса введите Число контактов: Дата. Окончательный результат запроса показан на рис. 8.

	Фамилия	Имя	Дата первого контакта	Дата последнего контакта	Число контактов
▶	Андреев	Николай	26.01.2006	03.02.2006	2
	Иванов	Антон	26.01.2006	02.02.2006	2
	Иванов	Иван	25.01.2006	05.02.2006	3
	Петров	Андрей	04.02.2006	04.02.2006	1
	Петров	Василий	01.02.2006	01.02.2006	1
	Петров	Николай	25.01.2006	01.02.2006	2

Запись: 1 из 6

Рис. 8

#### 2.5.4. Запросы на изменение

Кроме выполнения выборки данных и некоторых вычислений, запросы могут применяться также для добавления, удаления и обновления группы записей таблицы. Такие запросы являются мощным инструментом преобразования данных, они называются *запросами на изменение*. Предположим, что по каким-то причинам вам понадобилось скорректировать даты контактов, заменив во всех записях таблицы Список январь на февраль. Подобную операцию трудно проделать вручную, если в таблице содержится несколько тысяч записей. *Запрос на обновление* позволяет быстро решить поставленную задачу:

1. В окне базы данных щелкните на кнопке Таблицы.
2. Выделите таблицу Список, данные которой нужно обновить.
3. В палитре кнопки Новый объект выберите пункт Запрос. Откроется окно диалога Новый запрос, показанное на рис. 9. Подобное окно открывается и при щелчке на кнопке Создать окна базы данных. Оно позволяет выбрать наиболее удобный способ создания объекта.
4. Дважды щелкните на строке Конструктор. Таблица Список, выделенная в окне базы данных на шаге 2, автоматически появится в окне конструктора запросов.
5. Перетащите в бланк запроса поле Дата, значение которого нужно обновлять.
6. В палитре кнопки Тип запроса выберите пункт Обновление. Описание всех возможных вариантов запросов, предлагаемых программой Access, приведено в табл. 6. Структура бланка запроса видоизменяется в соответствии с типом запроса. В бланке выбранного варианта запроса на обновление появляется поле Обновление, в которое нужно ввести новое зна-

чение поля. Чтобы изменить месяц даты с января на февраль, достаточно прибавить к дате 31 день.

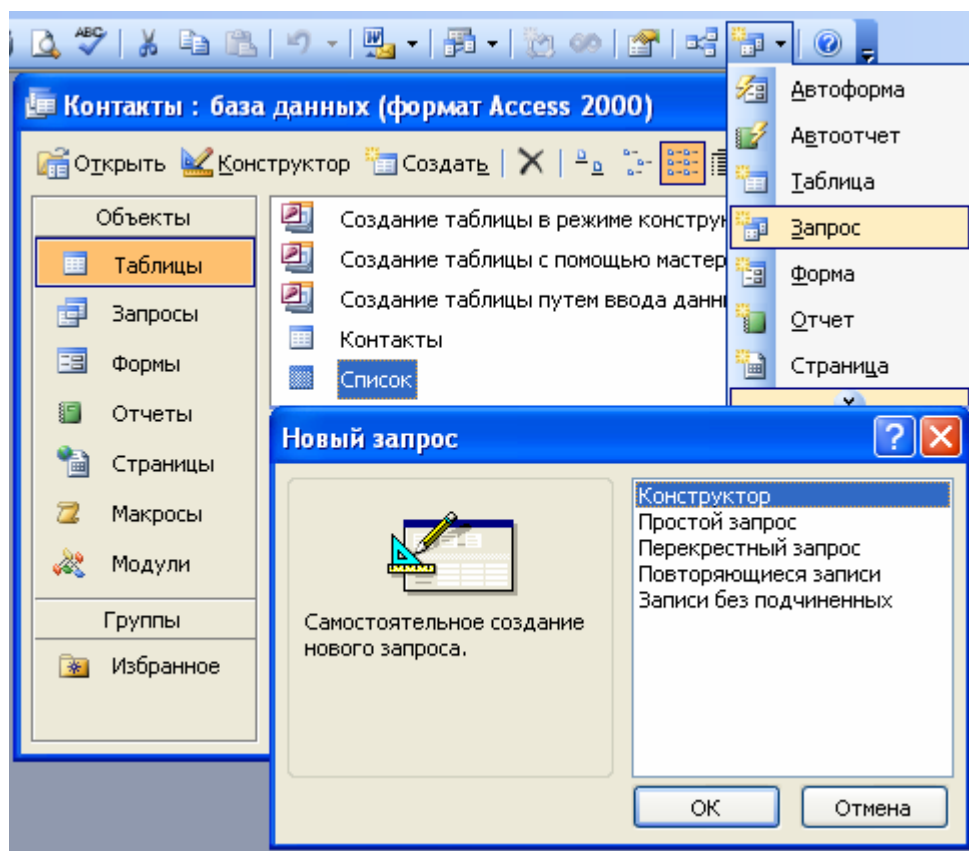


Рис. 9

Таблица 6

Тип	Описание
Выборка (Select)	Выборка данных в таблицу результата запроса на основе указанных условий отбора
Перекрестный (Crosstab)	Результат запроса выводит статистические значения (сумму, количество или среднее) для одного из полей таблицы в зависимости от двух параметров других полей таблицы, задающих заголовки строк и столбцов результата перекрестного запроса
Создание таблицы (Make-Table)	Создание новой таблицы в текущей или в другой базе данных на базе информации из имеющихся таблиц
Обновление (Update)	Обновление данных таблицы
Добавление (Append)	Добавление набора записей в таблицу
Удаление (Delete)	Удаление записей таблицы в соответствии с указанным критерием

7. Введите в ячейку Обновление формулу [Дата]+31.
8. Чтобы обновлялись только даты, относящиеся к январю 2006 г., введите в ячейку Условие отбора формулу Between DateValue ("1.01.06") And DateValue ("31.01.06") (рис. 10).

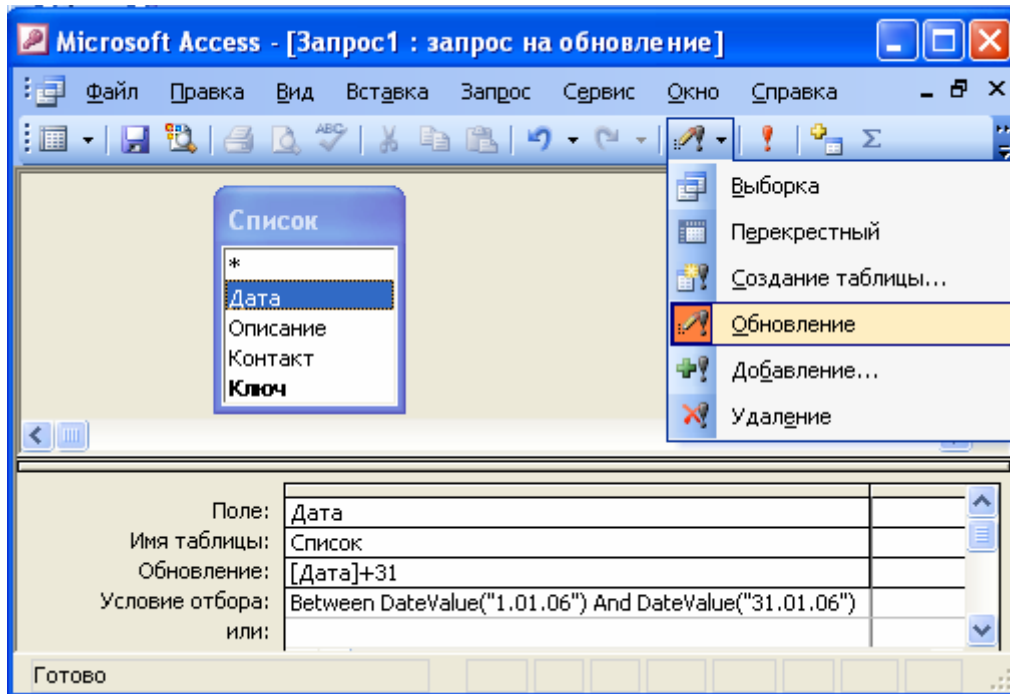


Рис. 10

9. Закройте запрос, сохранив его под именем Обновление;
10. Щелкните на кнопке Таблицы окна базы данных и двойным щелчком на значке Список откройте эту таблицу. В ней есть четыре записи, относящиеся к январю 2006 г.;
11. В окне базы данных щелкните на кнопке Запросы;
12. Дважды щелкните на значке Обновление;
13. Ответьте Да на вопрос о необходимости запуска запроса на обновление. Access проинформирует вас об обнаружении четырех записей, удовлетворяющих условию отбора, и попросит подтвердить необходимость их изменения;
14. Щелкните на кнопке Да и изучите изменения, произошедшие с данными таблицы Список.

## 2.6. Фильтры

Фильтры, как и запросы, предназначены для отбора определенных записей базы данных. Но фильтр сохраняется вместе с таблицей и не мо-

жет использоваться с другими объектами, если он не был записан в виде запроса.

### 2.6.1. Установка фильтра

Установка фильтра – простейший способ отбора части записей в таблице, запросе или форме. Выделим несколько записей таблицы Контакты:

1. В окне базы данных щелкните на кнопке Таблицы, а затем дважды на значке Контакты;
2. В столбце Фамилия щелкните на ячейке с фамилией Петров;
3. Щелкните на кнопке Фильтр по выделенному. В результате видимыми останутся только те записи, в поле Фамилия которых присутствует значение Петров (рис. 11);

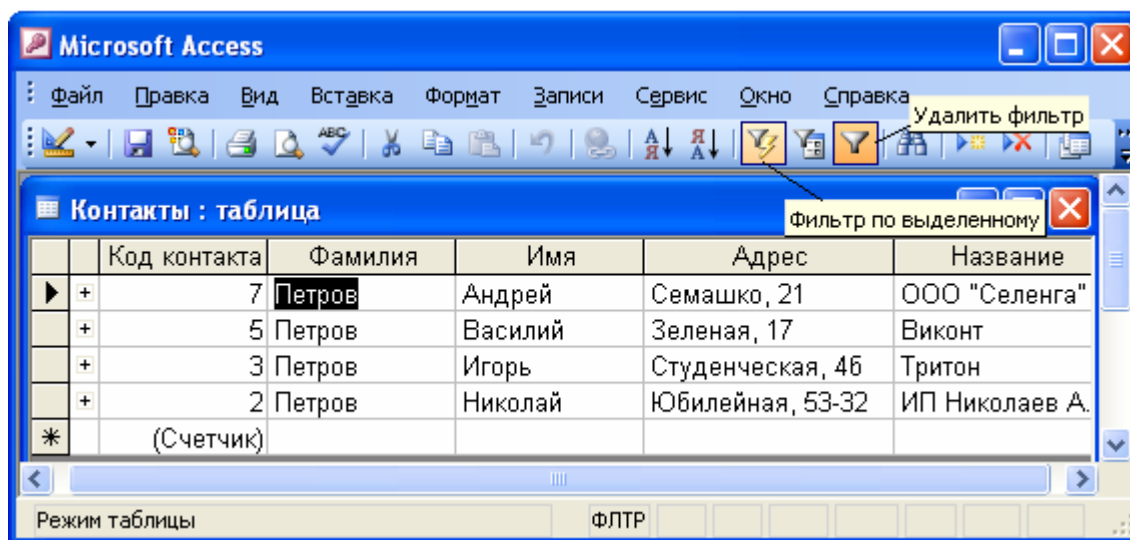


Рис. 11

4. Чтобы отменить влияние фильтра, щелкните на кнопке Удалить фильтр;

5. Фильтр может быть и более сложным. Например, чтобы оставить все записи с фамилией Петров и непустым номером телефона, выберите команду Записи/Фильтр/Расширенный фильтр. Откроется окно, похожее на конструктор запроса. Критерий равенства поля Фамилия значению Петров уже будет присутствовать в бланке фильтра;

6. Перетащите поле РабочийТелефон во второй столбец бланка;

7. В ячейку Условие отбора введите формулу <>Null. Значение любого пустого поля равно величине Null. Условие <>Null проверяет поле на неравенство величине Null, то есть отбирает все непустые поля.

8. Выберите команду **Фильтр/Применить фильтр**. Настроенный фильтр будет применен к таблице **Контакты**. В результате в ней останутся только две записи;

9. С помощью команды **Записи/Удалить фильтр** отмените фильтрацию.

### *2.6.2. Преобразование в запрос*

К сожалению, назначение нового фильтра автоматически стирает все предыдущие, которые уже невозможно применить повторно. Если фильтр достаточно сложный, его многократная настройка может отнимать много времени, и вам, вероятно, захочется как-то спасти результат своего труда. Access предлагает простой способ сохранения фильтров. Так как окно фильтра подобно конструктору запроса, было бы логично записать фильтр в виде запроса с теми же условиями отбора. Чтобы превратить фильтр в запрос, выполните следующие шаги:

1. Командой **Записи/Фильтр/Расширенный фильтр** активизируйте окно сложного фильтра, созданного в предыдущем упражнении;

2. Выполните команду **Файл/Сохранить как запрос**;

3. В открывшемся окне диалога введите имя **Фильтр** и щелкните на кнопке **ОК**;

4. Закройте окно фильтра;

5. В окне базы данных появился новый запрос с названием **Фильтр**. Дважды щелкните на значке этого запроса. Откроется окно результата выполнения запроса, записи которого будут повторять записи фильтрованной таблицы **Контакты**. Этим запросом можно пользоваться когда угодно, выводя на экран записи, отобранные по соответствующим критериям;

6. Любой запрос можно превратить в фильтр. Щелчком активизируйте окно таблицы **Контакты** (если таблица закрыта, откройте ее);

7. Чтобы назначить новый фильтр (и стереть предыдущий), щелкните на ячейке с фамилией **Петров** правой кнопкой мыши и выберите в контекстном меню команду **Исключить выделенное**. В таблице останутся только те записи, в поле **Фамилия** которых нет слова **Петров**;

8. Щелчком на кнопке **Удалить фильтр** отмените фильтрацию данных;

9. Теперь давайте вернем прежний сложный фильтр. Выберите команду **Записи/Фильтр/Расширенный фильтр**;

10. Выберите команду **Файл/Загрузить из запроса**;

11. В открывшемся окне диалога выделите запрос **Фильтр** и щелкните на кнопке **ОК**. Старые условия фильтрации появятся в бланке окна фильтра. Теперь для применения сохраненного фильтра достаточно щелкнуть на кнопке **Применение фильтра** панели инструментов.

### 2.7. Печать данных с помощью отчетов

Можно осуществлять печать непосредственно таблиц, форм и запросов с помощью команды **Файл/Печать**. Однако для красивой печати документов целесообразно использовать отчеты. Отчеты являются производными объектами БД и создаются на основе таблиц, форм и запросов. В Access входит несколько средств для создания отчетов (рис. 12):

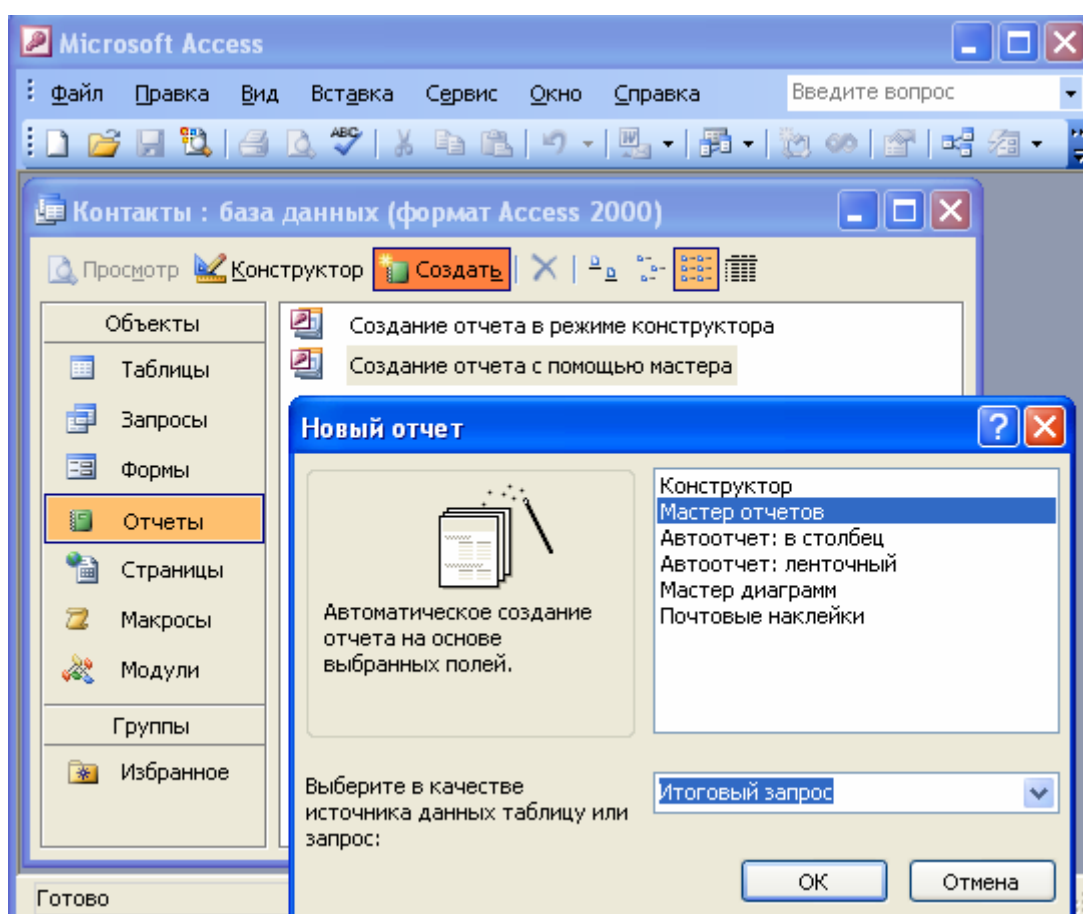


Рис. 12

- Автоотчет;
- Мастер отчетов (для создания отчетов, основанных на одной или нескольких таблицах);
- Мастер почтовых наклеек;

- Мастер диаграмм;
- Конструктор отчетов.

*Автоотчет* – это наиболее простое средство для создания отчета на основе таблицы или запроса, работает быстрее, чем мастер. В Автоотчет включаются все поля выбранной таблицы или запроса. *Автоотчет в столбце* отображает записи из таблицы или запроса по столбцам. *Ленточный автоотчет* отображает записи в виде таблицы. Его удобно использовать для таблиц и запросов с небольшим количеством полей. Автоотчет можно использовать в качестве базы для быстрого создания многотабличного отчета. Для этого сначала создайте отчет для главной таблицы. Затем откройте отчет в окне конструктора и перетащите в него из базы данных подчиненную таблицу.

*Мастер отчетов* позволяет:

- выбрать поля из одной или нескольких таблиц и запросов;
- сгруппировать записи по определенным полям;
- отсортировать записи в отчете по четырем (или менее) полям;
- вывести при необходимости итоговые величины, а также скрыть основанные на них подчиненные записи;
- выбрать макет, ориентацию и стиль отчета.

В режиме *конструктора отчетов* можно разработать отчет «с нуля» или доработать отчет, созданный с помощью Мастера отчетов. В отчет можно добавить итоговые и вычисляемые поля.

*Мастер почтовых наклеек* создает отчеты с оформлением, которое используется для печати наклеек стандартных форматов, таких как *Avery*, *EXPE*, *Herma*, *Zweckform*. С помощью этого мастера можно задать собственные параметры наклеек, если не подходит ни один из стандартных типов.

Довольно часто встает задача создания отчета, в котором необходимо отобразить не все, а только выбранные на основании некоторого условия записи – *отчета по выборке*. Сделать это можно несколькими способами:

- если эти условия постоянны, создайте отчет на основе запроса, в котором определены необходимые условия отбора;
- если условия отбора изменяются, создайте запрос с параметрами, а на основании этого запроса – отчет;

- откройте таблицу, примените к ней фильтр и сохраните ее. Затем создайте для данной таблицы отчет. Перед созданием отчета требуется активизировать фильтр.

### **3. Задание к работе**

Разработать реляционную базу данных, содержащую несколько связанных таблиц, входные формы, запросы и отчеты по самостоятельно выбранной теме или в соответствии с предложенными вариантами.

*Вариант 1.* База данных «Городской телефонный справочник».

*Вариант 2.* База данных «Алфавитный каталог библиотеки».

*Вариант 3.* База данных «Систематический каталог библиотеки».

*Вариант 4.* База данных «Деканат».

*Вариант 5.* База данных «Кафедра».

*Вариант 6.* База данных «Учебный план».

*Вариант 7.* База данных «Выпускники».

*Вариант 8.* База данных «Учебная группа».

*Вариант 9.* База данных «Компьютерный салон».

*Вариант 10.* База данных «Музыкальная коллекция».

*Вариант 11.* База данных «Библиотечный абонемент».

### **4. Содержание отчета**

1. Цель работы.
2. Вариант индивидуального задания.
3. Описание предметной области.
4. Структура базы данных.
5. Разработанные формы, запросы и отчеты.
6. Выводы по работе.

### **5. Контрольные вопросы**

1. Какие базовые информационные процессы реализуются при использовании баз данных?
2. В каких базовых и прикладных информационных технологиях могут применяться базы данных?
3. Назовите виды баз данных.
4. Каковы этапы проектирования баз данных?
5. Каковы основные функции системы управления базами данных?
6. Перечислите основные объекты СУБД MS Access.



7. Перечислите виды отношений между таблицами.
8. Что такое первичный и внешний ключи?
9. В чем сущность индексации базы данных?
10. Поясните процесс нормализации базы данных.
11. Какие типы полей данных допустимы в СУБД MS Access?

## **Лабораторная работа № 4**

### **ПОИСК ОПТИМАЛЬНЫХ РЕШЕНИЙ С ПОМОЩЬЮ MS EXCEL**

#### **1. Цель работы**

Изучить методы поиска оптимальных решений на примере задачи линейного программирования, на практике освоить возможности приложения MS Excel для решения подобных задач.

#### **2. Основные сведения**

Оптимальные решения являются в заданных условиях наилучшими из всех возможных решений в смысле принятого критерия оптимальности. В экономических расчетах стараются достигнуть максимальной прибыли, принимая величину прибыли за критерий оптимальности. Принятие оптимальных решений связано с громоздкими расчетами и требует не только средств вычислительной техники, но и соответствующих программ. Приложение MS Excel является удобным общеупотребительным инструментом решения этих задач.

Решение большинства задач оптимизации обладает рядом общих характерных черт. Во-первых, необходимо выбрать критерий. Во-вторых, надо описать зависимость между исходными данными и искомыми величинами, другими словами – составить математическую модель исследуемого объекта (процесса или явления). На этом этапе необходимо определить переменные и установить, какие ограничения должны быть наложены на переменные, чтобы выполнялись условия, характерные для изучаемого объекта. И, наконец, для решения задачи нужны достоверные исходные данные.

Наиболее распространенной в экономике задачей принятия оптимальных решений является задача линейного программирования. Рассмотр-

рим решение этой задачи на примере задачи оптимального распределения ресурсов.

### 2.1. Постановка задачи линейного программирования

Если финансы, оборудование, сырье и даже людей полагать ресурсами, то большинство экономических задач можно рассматривать как задачи распределения ресурсов.

Предположим, что необходимо определить, в каком количестве надо выпускать продукцию четырех типов П1, П2, П3, П4, для изготовления которой требуются ресурсы трех видов: сырье, финансы и трудовые ресурсы. Количество ресурсов каждого типа, необходимое для выпуска единицы продукции конкретного типа, называется нормой расхода. Нормы расхода, а также прибыль, получаемая от реализации единицы каждого типа продукции, приведены в табл. 7. Там же приведено наличие ресурсов каждого вида.

Таблица 7

Ресурсы	Продукция				Знак	Наличие
	П1	П2	П3	П4		
Трудовые	1	1	2	1	$\leq$	240
Сырье	6	5	4	3	$\leq$	1000
Финансы	4	6	8	12	$\leq$	1600
Прибыль	80	70	120	130	maximum	?

Как видно из таблицы 7, для выпуска единицы продукции П1 необходимы один человек/день, шесть единиц сырья и четыре условные единицы денег. При реализации единицы продукции этого типа будет получена прибыль в размере 80 условных единиц. Аналогичные сведения приведены для продукции П2, П3, П4. При этом предприятие располагает на месяц тысячью единиц сырья, 1600 условными единицами денежных средств и десятью рабочими (240 человек/день). Необходимо определить количество выпускаемой продукции каждого типа, обеспечивающее получение максимальной прибыли при заданных ограничениях на ресурсы.

Составим математическую модель, для которой введем следующие обозначения:

$F$  – прибыль;

$x_j$  – количество выпускаемой продукции  $j$ -го типа;

$b_i$  – количество располагаемого ресурса  $i$ -го вида;

$a_{ij}$  – норма расхода ресурса  $i$ -го вида для выпуска единицы продукции  $j$ -го типа;  
 $c_j$  – прибыль, получаемая от реализации единицы продукции  $j$ -го типа;  
 $i = 1, m; j = 1, n$ .

Тогда стандартная модель задачи линейного программирования примет вид:

$$F = \sum_{j=1}^n c_j x_j \Rightarrow \max \quad \text{– целевая функция;}$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \quad \text{– ограничения;}$$

$$x_{j \text{ нижн}} \leq x_j \leq x_{j \text{ верхн}} \quad \text{– границы.}$$

Подставив в нее числовые значения коэффициентов из табл. 7, получим:

$$F = 80x_1 + 70x_2 + 120x_3 + 130x_4 \Rightarrow \max;$$

$$x_1 + x_2 + 2x_3 + x_4 \leq 240 \quad \text{– затраты трудовых ресурсов;}$$

$$6x_1 + 5x_2 + 4x_3 + 3x_4 \leq 1000 \quad \text{– затраты сырья;}$$

$$4x_1 + 6x_2 + 8x_3 + 12x_4 \leq 1600 \quad \text{– затраты финансов;}$$

$$x_j \geq 0; \quad j = 1, 4.$$

Здесь дополнительно введено простое ограничение на количество выпускаемой продукции – количество продукции любого вида должно быть больше нуля, что вполне согласуется со здравым смыслом. Следует заметить, что ограничения могут иметь и более сложный вид. Например, в задаче составления оптимальной номенклатуры товаров для крупного предприятия розничной торговли количество товара любого наименования должно быть больше одной упаковки, но меньше или кратно количеству продукции, помещающейся в стандартный контейнер.

Приведенная выше математическая модель является линейной, поэтому для решения поставленной задачи пригодны методы линейного программирования. Линейная задача распределения ресурсов решается с помощью симплекс-метода.

## 2.2. Решение задачи линейного программирования с помощью MS Excel

Сначала создайте форму для ввода исходных данных и зависимостей, характеризующих математическую модель данной задачи. Составьте две таблицы (рис. 13). В первую (диапазон ячеек A2:G6) введите переменные  $x_i$  и  $F$ , а также связывающее их соотношение  $F = 80x_1 + 70x_2 + 120x_3 + 130x_4$ , записанное в обозначениях Excel с помощью стандартной функции в виде: =СУММПРОИЗВ(\$B\$3:\$E\$3;B6:E6). Во вторую таблицу (диапазон ячеек A8:H11) введите соотношения, характеризующие затраты ресурсов на производство – в обозначениях Excel диапазон ячеек (F9:F11).

Обратите внимание, что значения переменных  $x_1, x_2, x_3, x_4$  еще неизвестны и по умолчанию Excel принимает содержимое ячеек B3:E3 равным нулю, поэтому и результаты расчетов в ячейках F6, F9:F11 также равны нулю.

	A	B	C	D	E	F	G	H
1	Переменные							
2	Имя	П1	П2	П3	П4			
3	Значение							
4	Нижн. гр.	0	0	0	0			
5	Верхн. гр.					ЦФ	Напр.	
6	Коэф. в ЦФ	80	70	120	130	=СУММПРОИЗВ(\$B\$3:\$E\$3;B6:E6)	max	
7	Ограничения							
8	Вид	П1	П2	П3	П4	Левая часть	Знак	Правая часть
9	Трудовые ресурсы	1	1	2	1	=СУММПРОИЗВ(\$B\$3:\$E\$3;B9:E9)	<=	240
10	Сырье	6	5	4	3	=СУММПРОИЗВ(\$B\$3:\$E\$3;B10:E10)	<=	1000
11	Финансы	4	6	8	12	=СУММПРОИЗВ(\$B\$3:\$E\$3;B11:E11)	<=	1600

Рис. 13

Теперь активизируйте команду Поиск решения... пункта Сервис основного меню Excel. На экране появится диалоговое окно Поиск решения. Начните ввод информации с поля Ограничения. Для этого нажмите кнопку Добавить. На экране появится очередное диалоговое окно Добавить ограничение. Поочередно введите в поля этого окна ограничения на переменные  $x_j$  и ресурсы, нажимая после ввода очередного ограничения кнопку

Добавить (рис. 14):  $B3 \geq B4$ ;  $C3 \geq C4$ ;  $D3 \geq D4$ ;  $E3 \geq E4$  и  $F9 \leq H9$ ;  $F10 \leq H10$ ;  $F11 \leq H11$ .

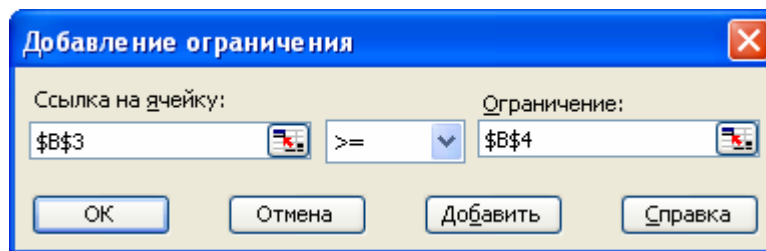


Рис. 14

После ввода последнего ограничения нажмите кнопку ОК и на экране снова появится диалоговое окно Поиск решения. В поля этого окна введите адрес целевой ячейки ( $F6$ ), которая в результате решения должна принять максимальное значение, и адреса ячеек, где должны помещаться значения переменных  $x_j$  (ячейки  $B3:E3$ ) (рис. 15).

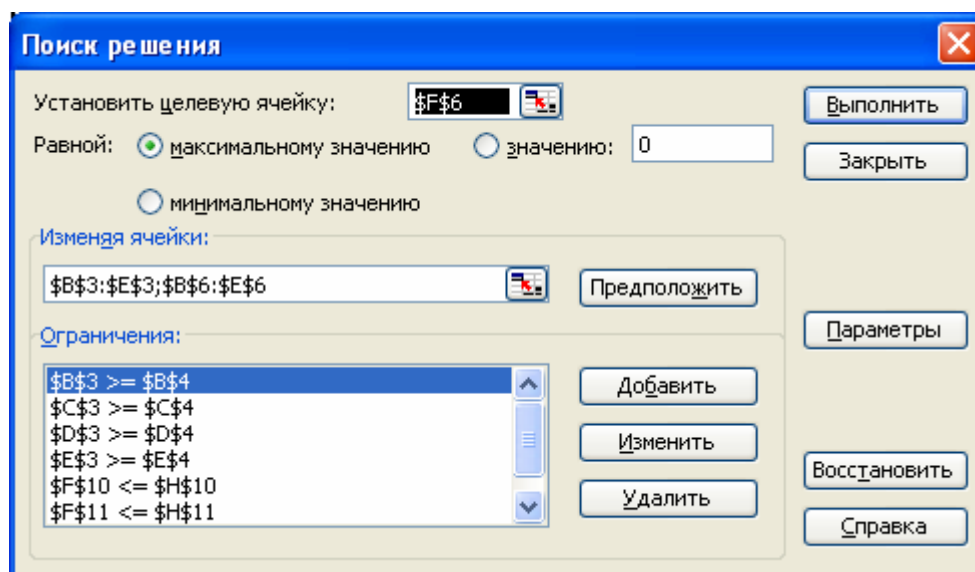


Рис. 15

Теперь остается установить параметры процесса оптимизации, нажав кнопку Параметры. Для этого в появившемся диалоговом окне Параметры поиска решения необходимо установить флажок Линей-

ная модель, который обеспечивает применение для решения задачи линейного программирования симплекс-метода, а также включить кнопку Линейная поля Оценка (рис. 16).

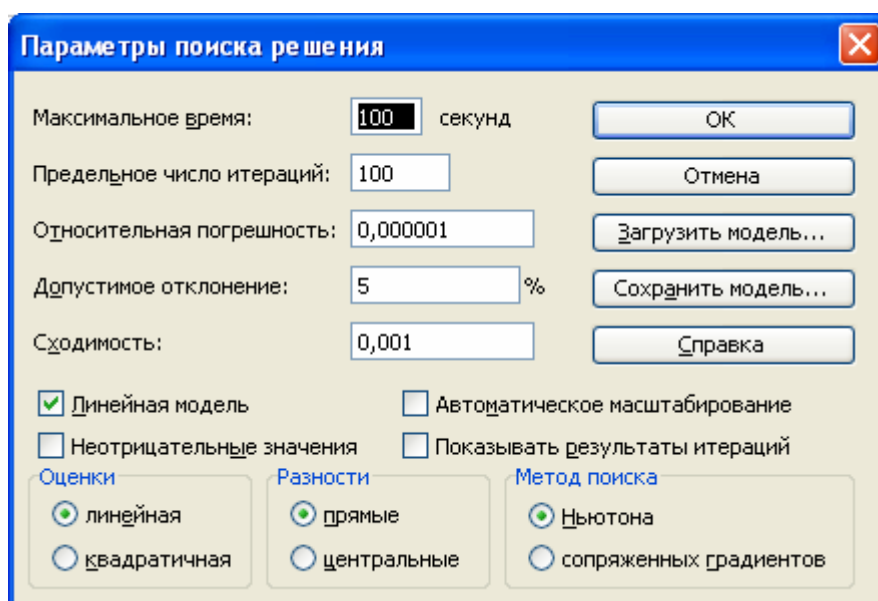


Рис. 16

Остальные параметры поиска решения целесообразно сохранить теми, которые установлены по умолчанию (см. рис. 16). После установки требуемых параметров остается нажать кнопку ОК. Будет выполнен поиск оптимального решения. Если задача была поставлена правильно и математическая модель соответствовала условиям задачи, то на экран будет выведено диалоговое окно Результаты поиска решения с сообщением «Решение найдено. Все ограничения и условия оптимальности выполнены» (рис. 17).

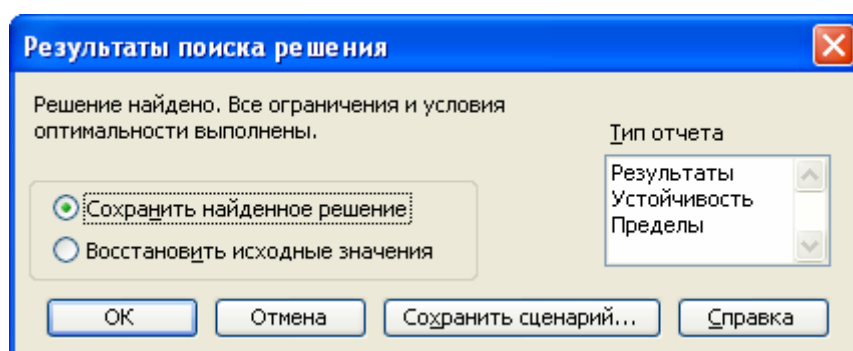


Рис. 17

Результаты решения появятся в таблице (рис. 18). Если условия задачи несовместны, то в этом окне будет помещено другое сообщение «Поиск не может найти подходящего решения». И, наконец, если целевая

функция не ограничена, то сообщение примет вид «Значения целевой ячейки не сходятся».

Установленные параметры поиска решения (см. рис. 16) подходят для большинства задач. При необходимости, обусловленной спецификой задачи, эти параметры могут быть изменены. Например, максимальное время можно увеличить до 32767 с (более 9 часов!).

	A	B	C	D	E	F	G	H
1	Переменные							
2	Имя	П1	П2	П3	П4			
3	Значение							
4	Нижн. гр.	110	0	25	80			
5	Верхн. гр.					ЦФ	Напр.	
6	Коэф. в ЦФ	80	70	120	130	22200	max	
7	Ограничения							
8	Вид	П1	П2	П3	П4	Левая часть	Знак	Правая часть
9	Трудовые ресурсы	1	1	2	1	240	<=	240
10	Сырье	6	5	4	3	1000	<=	1000
11	Финансы	4	6	8	12	1600	<=	1600

Рис. 18

### 2.3. Анализ оптимального решения

Анализ итоговой таблицы (см. рис. 18) показывает, что оптимальное решение существует ( $П1 = 110$ ;  $П2 = 0$ ;  $П3 = 25$ ;  $П4 = 80$ ). Максимальная прибыль составляет 22200 условных единиц. Все ресурсы оказываются полностью востребованными.

Однако с практической точки зрения приведенных в таблице результатов недостаточно. В реальной жизни условия меняются значительно чаще, чем хотелось бы. Например, вполне может измениться количество располагаемых ресурсов, может также измениться прибыль, получаемая при продаже единицы продукции конкретного типа. По этой причине представляют практический интерес ответы на такие вопросы:

- Изменится ли оптимальное решение при определенном изменении исходных данных?
- Как изменится прибыль при изменении исходных данных?

MS Excel автоматически отвечает на эти и некоторые другие вопросы. С помощью диалогового окна Результаты поиска решения (см. рис. 17) можно

получить подробные отчеты трех видов: отчеты по результатам, по устойчивости и по пределам изменения исходных данных.

### 2.3.1. Отчет по результатам

Отчет состоит из трех таблиц. В них содержатся данные о целевой функции, искомым переменных и ограничений до и после оптимизации. Также приведены зависимости, которые были введены в поле Ограничения диалогового окна Поиск решения, и данные по оставшимся не востребуемым ресурсам (столбец Разница таблицы Ограничения).

### 2.3.2. Отчет по устойчивости

Состоит из двух таблиц. В первой содержатся результат решения задачи, коэффициенты целевой функции, а также верхний (столбец Допустимое увеличение) и нижний (столбец Допустимое уменьшение) пределы коэффициентов целевой функции, при которых сохраняется набор переменных, входящих в оптимальное решение. Во второй таблице приводятся аналогичные значения для ограничений: величины исходных и использованных ресурсов, теневая цена (показывает, как изменится целевая функция при изменении ресурсов на единицу), а также верхний (столбец Допустимое увеличение) и нижний (столбец Допустимое уменьшение) пределы ресурсов, при которых сохраняется набор переменных, входящих в оптимальное решение.

### 2.3.3. Отчет по пределам

Этот отчет показывает, в каких пределах может изменяться выпуск продукции, вошедшей в оптимальное решение, при сохранении структуры оптимального решения. Приводятся оптимальные значения, нижний и верхний пределы изменения искомым переменных (объемов выпуска продукции). Указаны значения целевой функции при выпуске данного типа продукции на нижнем пределе. Далее приводятся верхние пределы изменения переменных и значения целевой функции при выпуске продукции, вошедшей в оптимальное решение на верхних пределах.

## 3. Задание к работе

Для закрепления пройденного материала подберите оптимальные объемы выпуска продукции для условий, приведенных в табл. 8.



Таблица 8

Ресурсы	Продукция				Знак	Наличие
	П1	П2	П3	П4		
Трудовые	1	1	1	1	<=	16
Сырье	6	5	4	3	<=	110
Финансы	4	6	10	13	<=	100
Прибыль	60	70	120	130	maximum	?

Сформируйте задачу по определенному преподавателем варианту как задачу линейного программирования и решите ее средствами MS Excel.

### Вариант 1

Фирма производит два продукта – А и В, рынок сбыта которых неограничен. Каждый продукт должен быть обработан каждой из машин I, II, III. Время обработки в часах для каждого из изделий приведено ниже:

	I	II	III
A	0,5	0,4	0,2
B	0,25	0,3	0,4

Время работы машин I, II, III соответственно 40, 36 и 36 ч в неделю. Прибыль от изделий А и В соответственно 5 и 3 дол. Фирме надо определить недельные нормы выпуска изделий А и В, максимизирующие прибыль.

### Вариант 2

Фирме требуется уголь с содержанием фосфора не более 0,03 % и с долей зольных примесей не более 3,25 %. Три сорта угля А, В, С доступны по следующим ценам (за 1 т), указанным в таблице.

Сорт угля	Содержание примеси, %		Цена, дол.
	фосфор	зола	
A	0,06	2,0	30
B	0,04	4,0	30
C	0,02	3,0	45

Как их смешивать, чтобы получить минимальную цену и удовлетворить ограничениям на содержание примесей?

### Вариант 3

Средства очистки пола оценивают по следующим трем показателям: а) очищающие свойства, б) дезинфицирующие свойства, в) раздражающее

воздействие на кожу. Каждый из этих показателей измеряется по линейной шкале от 0 до 100 единиц. Продукт на рынке должен иметь, по крайней мере, по 60 единиц очищающих и дезинфицирующих свойств по соответствующей шкале. При этом раздражающее воздействие на кожу должно быть минимальным. Конечный продукт должен быть смесью трех основных очистителей, характеристики которых приводятся в таблице.

Очиститель	Очищающие свойства	Дезинфицирующие свойства	Раздражающее воздействие на кожу
A	90	30	70
B	65	85	50
C	45	70	10

Решите задачу нахождения оптимальной смеси.

#### Вариант 4

Фирма занимается составлением диеты, содержащей, по крайней мере, 20 единиц белков, 30 единиц углеводов, 10 единиц жиров и 40 единиц витаминов. Содержание этих элементов в некоторых продуктах и цена последних (все в условных единицах) приведены в таблице.

Продукт	Содержание элементов, ед.				Цена за 1 кг (1 л)
	Белки	Углеводы	Жиры	Витамины	
Хлеб	2	12	1	2	12
Соя	12	0	8	2	36
Сушеная рыба	10	0	3	4	32
Фрукты	1	4	0	6	18
Молоко	2	3	4	2	10

Как дешевле всего достичь желаемого при указанных в таблице ценах за 1 кг (или 1 л) пяти имеющихся продуктов?

#### Вариант 5

В некоторой местности в пунктах А и В имеется потребность в дополнительном транспорте. В пункте А требуется 5 дополнительных автобусов, а в пункте В – 7. Известно, что 3, 4

Гараж	Расстояние до пункта	
	А	В
G1	3	4
G2	1	3
G3	4	2

и 5 автобусов могут быть получены соответственно из гаражей G1, G2, G3. Как следует распределить эти автобусы между пунктами, чтобы минимизировать их суммарный пробег?

Расстояние от гаражей до пунктов А и В приведены в таблице.

### Вариант 6

Компания производит полки для ванных комнат двух размеров – А и В. Агенты по продаже считают, что в неделю на рынке может быть реализовано до 550 полок. Для каждой полки типа А необходимо  $2 \text{ м}^2$  материала, типа В –  $3 \text{ м}^2$  материала. Компания может получить до  $1200 \text{ м}^2$  материала в неделю. На изготовление одной полки типа А необходимо 12 мин машинного времени, типа В – 30 мин; ЭВМ можно использовать 160 ч в неделю. Если прибыль от продажи полок типа А составляет 3 дол., а от полок типа В – 4 дол., то сколько полок каждого типа следует выпускать в неделю?

### Вариант 7

Автозавод выпускает две модели: "Каприз" и (более дешевую) "Фиаско". На заводе работает 1000 неквалифицированных и 800 квалифицированных рабочих, каждому из которых оплачивается 40 ч в неделю. Для изготовления модели "Каприз" требуется 30 ч неквалифицированного и 50 ч квалифицированного труда; для "Фиаско" – 40 ч неквалифицированного и 20 ч квалифицированного труда. Каждая модель "Фиаско" требует затрат в размере 500 дол. на сырье и комплектующие изделия, тогда как каждая модель "Каприз" – 1500 дол.; суммарные затраты не должны превосходить 900 000 дол. в неделю. Рабочие, осуществляющие доставку, работают по пять дней в неделю и могут забирать с завода не более 210 машин в день.

Каждая модель "Каприз" приносит фирме 1000 дол. прибыли, а каждая модель "Фиаско" – 500 дол. прибыли. Какой объем выпуска каждой модели вы бы порекомендовали? Что, по вашему мнению, необходимо для повышения прибыли фирмы?

### Вариант 8

Заводы фирмы расположены в городах Лидсе и Кардиффе; они доставляют товары на склады городов Манчестер, Бирмингем и Лондон. Расстояния между этими городами приведены в таблице (расстояния округлены до десятков миль).

Пункт назначения	Манчестер	Бирмингем	Лондон
Лидс	40	110	190
Кардифф	170	100	150

Завод в г. Лидсе выпускает в год 800 т товаров, а в г. Кардиффе – 500 т. Манчестерский склад вмещает 400 т, бирмингемский – 600 т, а лондонский – 300 т. Как следует транспортировать товары для минимизации цен на перевозки?

### Вариант 9

Фирма производит три вида продукции (А, В, С), для выпуска каждого из которых требуется определенное время обработки на всех четырех устройствах I, II, III, IV. Данные приведены в таблице.

Вид продукции	Время обработки, ч.				Прибыль, дол.
	I	II	III	IV	
А	1	3	1	2	3
В	6	1	3	3	6
С	3	3	2	4	4

Пусть время работы на устройствах – соответственно 84, 42,

21 и 42 ч. Определите, какую продукцию и в каких количествах следует производить. (Можете предположить, что рынок сбыта для каждого продукта неограничен; временем, требуемым для переключения устройства в зависимости от вида продукции, можно пренебречь; рассмотрите только задачу максимизации прибыли).

### Вариант 10

Производитель безалкогольных напитков располагает двумя различными машинами – А и В. Машина А спроектирована для пол-литровых бутылок, а машина В – для литровых, но каждая из них может использоваться для обоих типов бутылок с некоторой потерей эффективности в со-

Машина	Количество бутылок, производимых в одну минуту	
	Пол-литровые	Литровые
А	50	20
В	40	30

ответствии с приведенными в таблице сведениями о работе машин.

Каждая из машин работает ежедневно по 6 ч при пятидневной рабочей неделе. Прибыль от пол-литровой бутылки составляет 4 цента, а от литровой – 10 центов. Недельная продукция не может превосходить 50 000 л; рынок принимает не более 44 000 пол-литровых бутылок и 30 000 литровых.

Производитель хочет максимизировать свою прибыль при имеющихся средствах.

### Вариант 11

Небольшая фирма производит два типа подшипников – А и В, каждый из которых должен быть обработан на трех станках: токарном, шлифовальном и сверлильном. Полное возможное время работы станков в не-

делю соответственно 160, 120 и 150 ч. Время, требуемое для каждой из стадий производственного процесса, приведено в таблице.

Тип подшипника	Время обработки на станке, ч			Прибыль от продажи одного подшипника, центы
	Токарный	Шлифовальный	Сверлильный	
А	0,01	0,02	0,04	80
В	0,02	0,01	0,01	125

Фирма хотела бы производить подшипники в количествах, максимизирующих ее прибыль.

### Вариант 12

Нефтяная компания закупает необработанную нефть из нескольких источников (W, X, Y, Z) и занимается ее очисткой, вырабатывая различные виды (А, В, С) смазочных масел, готовых к продаже. Ограничения при продаже на количество каждого вида смазочных масел приведены в таблице.

Масло	Состав, %	Возможное количество для продажи, галлоны
А	Не меньше 10 (W) Не больше 25 (Z)	90 000
В	Не меньше 15 (W)	100 000
С	Не меньше 20 (X) Не больше 50 (Y)	120 000

Цена (в условных единицах) одного галлона сырья и смазочных масел приведена в таблице.

Предполагая, что необработанная нефть доступна в неограниченном количестве, сформулируйте задачу максимизации прибыли как задачу линейного программирования и найдите оптимальное решение.

Сырье	Цена	Масло	Цена
X	72	А	90
Y	60	В	87
Z	67	С	84
W	75		

### Вариант 13

Фирма рекламирует свою продукцию с использованием четырех средств: телевидения, радио, газет и афиш. Из различных рекламных экспериментов, которые проводились в прошлом, известно, что эти средства приводят к увеличению прибыли соответственно на 10, 3, 7 и 4 дол. в расчете на 1 дол., затраченный на рекламу.

Распределение рекламного бюджета по различным средствам подчинено следующим ограничениям:

- а) полный бюджет не должен превосходить 500 000 дол.;
- б) следует расходовать не более 40 % бюджета на телевидение и не более 20 % – на афиши;
- в) вследствие привлекательности для подростков радио, на него следует расходовать, по крайней мере, половину того, что планируется на телевидение.

Сформулируйте задачу распределения средств по различным источникам как задачу линейного программирования и решите ее.

#### Вариант 14

Фирма производит продукты А и В, продавая их соответственно по 8 и по 15 центов за упаковку; рынок сбыта для каждого из них практически неограничен. Продукт А обрабатывается на машине 1, продукт В – на машине 2. Затем оба упаковываются на фабрике (рис. 19).

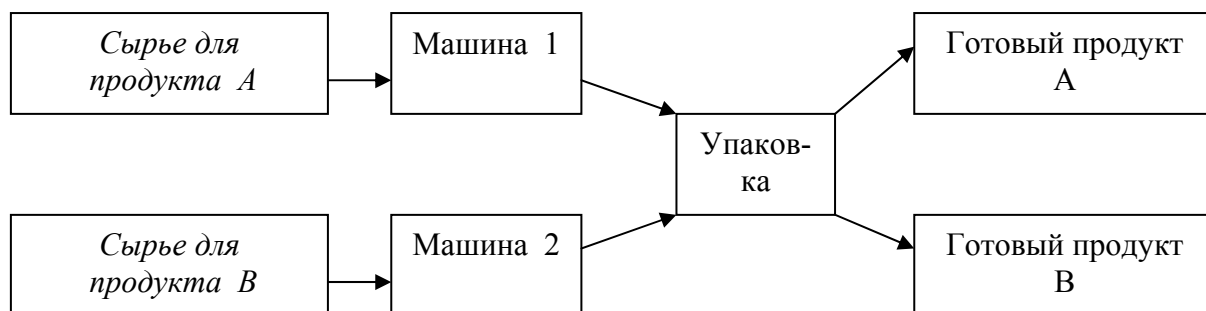


Рис. 19

1 кг сырья стоит 6 центов; машина 1 обрабатывает 5000 кг в час с потерями 10 %. Машина 2 обрабатывает 4000 кг в час с потерями 20 %. Машина 1 доступна 6 часов в день, ее использование стоит 288 дол. в час. Машина 2 доступна 5 часов в день, и ее использование стоит 336 дол. в час. Упаковка продукта А весит 1/4 кг, упаковка продукта В – 1/3 кг. Фабрика может работать 10 часов в день, производя в час продукцию стоимостью 360 дол. За час можно упаковать 12000 продуктов А и 8000 продуктов В.

Компания хочет определить такие значения  $x_1$  и  $x_2$  потребления сырья для продуктов А и В (в тысячах килограммов), при которых дневная прибыль максимальна.

#### 4. Содержание отчета

1. Цель работы.
2. Вариант индивидуального задания.
3. Математическая модель задачи.
4. Форма исходных данных для решения задачи в MS Excel.
5. Результаты и анализ решения задачи.

#### 5. Контрольные вопросы

1. В каком базовом информационном процессе может быть применена задача линейного программирования?
2. В какой базовой информационной технологии может быть применена задача линейного программирования?
3. Как строится математическая модель задачи?
4. Какие средства MS Excel используются для нахождения оптимального решения?
5. Какие виды анализа полученного оптимального решения можно выполнить в MS Excel?

#### СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Информатика : учебник / под ред. Н. В. Макаровой. – М. : Финансы и статистика, 1997. – 768 с. – ISBN 5-279-01841-4.
2. *Советов, Б. Я.* Информационные технологии: учеб. для вузов / Б. Я. Советов, В. В. Цехановский. – М. : Высш. шк., 2003. – 263 с. : ил. – ISBN 5-06-004275-8.
3. *Грабауров, В. А.* Информационные технологии для менеджеров / В. А. Грабауров. – М. : Финансы и статистика, 2001. – 368 с. : ил. – (Прикладные информационные технологии). – ISBN 5-279-02299-3.
4. *Острейковский, В. А.* Информатика : учеб. для вузов / В. А. Острейковский. – М. : Высш. шк., 2001. – 511 с. : ил. – ISBN 5-06-003533-6.
5. *Юзвишин, И. И.* Основы информациологии : учебник / И. И. Юзвишин. – М. : Информациология ; Высш. шк., 2000. – 517 с. : ил. – ISBN 5-87489-029-7 (Информациология); ISBN 5-06-003973-0 (Высш. шк.).
6. Практикум по информатике : учеб. пособие для студентов высш. учеб. заведений / А. В. Могилев, Н. И. Пак, Е. К. Хеннер ; под. ред. Е. К. Хеннера. – М. : Издательский центр «Академия», 2001. – 608 с. – ISBN 5-7695-0529-X.

7. Новиков, Ф. А. Microsoft Office 2000 в целом / Ф. А. Новиков, А. Д. Яценко. – СПб. : БХВ-Петербург, 2001. – 728 с. : ил. – ISBN 5-8206-0035-5.

8. Робинсон, С. Microsoft Access 2000 : учеб. курс : [пер. с англ.]. – СПб : Питер, 2001. – 512 с. : ил. – ISBN 5-272-00004-8.

9. Жигалов, И. Е. СУБД и распределенная обработка информации : лаб. практикум / И. Е. Жигалов. – Владимир : Владим. гос. ун-т, 1999. – 80 с. – ISBN 5-89368-149-5.

10. Браун, М. Использование HTML 4 : [пер. с англ.]. / М. Браун, Р. Марк, Д. Хоникат [и др]. – 4-е изд., спец. – М. [и др.]: Вильямс, 1999. – 784 с. : ил. – ISBN 5-8459-0015-8.

11. Курицкий, Б. Я. Поиск оптимальных решений средствами Excel 7.0 / Б. Я. Курицкий. – СПб. : ВHV – Санкт-Петербург, 1997. – 384 с. : ил. – ISBN 5-7791-0037-3.

## ОГЛАВЛЕНИЕ

Введение.....	3
Лабораторная работа № 1. КОМПЬЮТЕРНЫЕ ПРЕЗЕНТАЦИИ.....	5
Лабораторная работа № 2. ОСНОВЫ ЯЗЫКА РАЗМЕТКИ ГИПЕРТЕКСТОВЫХ ДОКУМЕНТОВ .....	11
Лабораторная работа № 3. СОЗДАНИЕ И РАБОТА С РЕЛЯЦИОННОЙ БАЗОЙ ДАННЫХ .....	28
Лабораторная работа № 4. ПОИСК ОПТИМАЛЬНЫХ РЕШЕНИЙ С ПОМОЩЬЮ MS EXCEL....	49
Список рекомендуемой литературы.....	63



## ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Методические указания к лабораторным работам

Составитель

КИРИЛЛОВА Светлана Юрьевна

Ответственный за выпуск – зав. кафедрой профессор А.В. Костров

Редактор Е.В. Афанасьева

Технический редактор Н.В. Тупицына

Корректор В.В. Гурова

Компьютерная верстка С.Ю. Кирилловой

Подписано в печать 14.09.06.

Формат 60x84/16. Бумага для множит. техники. Гарнитура Таймс.

Печать на ризографе. Усл. печ. л. 3,72. Уч.-изд. л. 3,95. Тираж 100 экз.

Заказ

Редакционно-издательский комплекс

Владимирского государственного университета.

600000, Владимир, ул. Горького, 87.