

Министерство сельского хозяйства Российской Федерации

Федеральное государственное образовательное учреждение
высшего профессионального образования
«Государственный университет по землеустройству»

Кафедра информатики

ПРАКТИКУМ ПО ИНФОРМАТИКЕ

Под общей редакцией профессора
М.И. Коробочкина

Москва 2008

УДК 681.3
ББК 32.97
П 69

Под общей редакцией профессора, д.т.н.
М.И. Коробочкина,

Ответственный редактор:
С.С.Репин

Рецензенты:
В.В. Коссинский, профессор д.э.н.;
В.Н. Никонов, к.э.н

П 69 Практикум по информатике / М.И. Коробочкин, Г.З. Гарбер, Е.Е. Дмитриева, Калинова Е.В., Репин С.С., Красницкий В.С, Тимохин Ю.Н.; Гос. ун-т по землеустройству. – М.: ГУЗ, 2008. – **461 с.**

ISBN

Практикум предназначен для поддержки курса «Информатика». Содержит 6 глав, в которых рассматриваются программные приложения пакета Microsoft Office 2007: Word, Power Point, Excel, Access, основные возможности операционной среды Windows. Содержатся основные сведения, необходимые для работы в сети Internet. Излагается теоретический и практический материал для освоения основных приемов работы в интегрированной среде программирования Delphi. Практикум по информатике содержит упражнения и практические задания, снабжен индивидуальными заданиями, способствующими усвоению изучаемого материала.

Книга предназначена для студентов, слушателей факультетов повышения квалификации и широкого круга пользователей.

УДК 681.3
ББК 32.97

ISBN

© Государственный университет по землеустройству, 2008
© Коробочкин М.И., научный редактор, 2008
© Репин С.С., ответственный редактор, 2008

ВВЕДЕНИЕ

Практикум предназначен для поддержки теоретического курса, лабораторных работ и самостоятельных занятий студентов по дисциплине «Информатика». Он состоит из 6 глав, в которых рассматриваются использование программных приложений Microsoft Office 2007 (текстовый редактор Word, приложение Microsoft Power Point, табличный процессор Excel, система управления базами данных Access), основные возможности и технология работы в среде операционной системы Windows, Internet, а также основы технологии разработки проектов на языке Delphi в интегрированной среде Delphi.

Практикум содержит упражнения и практические задания, способствующие усвоению курса.

Глава 1 посвящена работе в среде текстового редактора Word: создание и сохранение текстового документа, загрузка и редактирование, работа с выделенными участками текста, установка необходимого размера документа, размещение текста колонками, вставка таблиц, формул, рисунков, гиперссылок, вёрстка многостраничных документов. Показан пример создания презентаций с помощью приложения Microsoft Power Point на основе материала, подготовленного в Word.

Глава 2 содержит сведения об основных возможностях операционной системы Windows XP, упражнения и задания для индивидуального выполнения.

Глава 3 содержит описание основных принципов работы в среде табличного процесса Excel. Существенное внимание уделено работе с абсолютными и относительными адресами, построению и копированию формул, работе с логическими структурами. Рассмотрены вопросы работы с массивами, матричные операции, решение систем линейных уравнений, решение уравнений путём подбора параметров. Описана технология решения оптимизационных задач линейного, квадратичного и нелинейного программирования с помощью процедуры «Поиск решения». Необходимое внимание уделено графическому представлению данных. Рассмотрено использование Excel для управления базами данных, включая анализ данных с использованием сводных таблиц. Приведены примеры создания макросов и одновременного использования Excel и других программ пакета Office.

Глава 4 содержит описание структуры баз данных (БД) реляционного типа и описание технологии работы в среде системы управления БД Access. Приведен пример создания БД. Рассмотрены вопросы рационального выбора параметров полей БД, создания связи между таблицами в рамках одной базы данных, создания формы для ввода данных в таблицу и редактирования таблицы. Дано описание технологии создания отчета, а также сортировки, фильтрации и организации выборки данных.

Глава 5 содержит описание идеологии построения сети Internet и некоторые элементы технологии работы в среде программ, обслуживающих эту сеть. Описана работа с поисковыми системами, электронная почта, создание Web-страниц и размещение их на сайтах.

Глава 6 посвящена изложению основных принципов объектного программирования на языке Delphi в интегрированной среде Delphi. Дано описание ряда технологических приемов создания программных проектов. Приведено большое количество примеров и задач для самостоятельной разработки проектов линейных, разветвляющихся и циклических программ, а также требующих использования процедур и функций программиста, с вводом информации с внешних носителей и различными видами вывода результатов.

Первое издание «Практикума по информатике» вышло в 2002 году, второе издание – в 2003 году. Настоящее третье издание существенно переработано и дополнено.

Изложение материала полностью опирается на работу с пакетом Office 2007, который существенно отличается от предыдущих версий.

«Практикум по Информатике» можно рекомендовать в качестве учебного пособия по курсу «Информатика» студентам вузов, слушателям факультетов повышения квалификации и широкому кругу пользователей.

Глава 1 – ст. преп. Репин С.С., доц., к.т.н. Красницкий В.С.; Глава 2 – доц., к.т.н. Дмитриева Е.Е.; Глава 3 – проф., д.т.н. Коробочкин М.И., доц., к.т.н. Калинова Е.В.; Глава 4 – ст. преп. Тимохин Ю. Н.; Глава 5 – проф., д.т.н. Г. З. Гарбер; Глава 6 – доц., к.т.н. Дмитриева Е.Е.

Глава 1

РАБОТА В СРЕДЕ ТЕКСТОВОГО РЕДАКТОРА WORD

§1. Общие сведения, терминология Microsoft Word

Word – это текстовый редактор, программа, которая предлагает пользователю ряд услуг, необходимых при создании на компьютере текста, рисунков, таблиц, диаграмм, формул и т.п.

Пользователем называют лицо, использующее программу в своих целях.

Использование той или иной компьютерной программы называют работой в среде этой программы.

Создание текстовых документов – наиболее распространённый способ использования компьютеров. С этим сталкиваются многие: студенты при выполнении курсовых и дипломных работ, секретари при печати деловых бумаг и контрактов, писатели при создании текстов книг и др. Трудно представить сферу деятельности, в которой компьютер не применяется для подготовки к печати документов.

Огромный спрос на удобные приложения для создания документов побудил производителей программного обеспечения разрабатывать и развивать свои продукты.

Бесспорным лидером в данной области является программа Microsoft Word. Её разработчики предусмотрели, казалось бы, всё, с чем вы можете столкнуться на различных этапах создания документа. Приложение Microsoft Word предоставляет все необходимые возможности для создания, редактирования и форматирования текста, создания нумерованных и маркированных списков, таблиц, предметных указателей и оглавлений. Обширный набор инструментов для работы с графикой позволит производить обработку иллюстраций, не прибегая к помощи графических редакторов. Множество готовых стилей оформления документов, иллюстраций и таблиц позволят быстро придать документу профессиональный и стильный вид.

Новый интерфейс программы Microsoft Word версии 2007 может показаться непривычным, но через некоторое время вы поймёте и оцените, насколько удобно расположены элементы ин-

терфейса. Главное новшество – разработчики программы Microsoft Word 2007 полностью отказались от строки меню, которая была в предыдущих версиях программы. Панели инструментов программы сгруппированы на вкладках. Если в более ранних версиях программы вы видели перед собой одну и ту же панель инструментов, независимо от того, какую операцию вы выполняете, то теперь в верхней части окна программы расположены инструменты, необходимые для выполнения конкретной задачи. Выбрав любую другую вкладку, вы отобразите на экране новый набор инструментов. Такая организация инструментальной палитры позволила разместить больше кнопок и других элементов управления в верхней части окна Microsoft Word без ущерба для рабочего пространства.

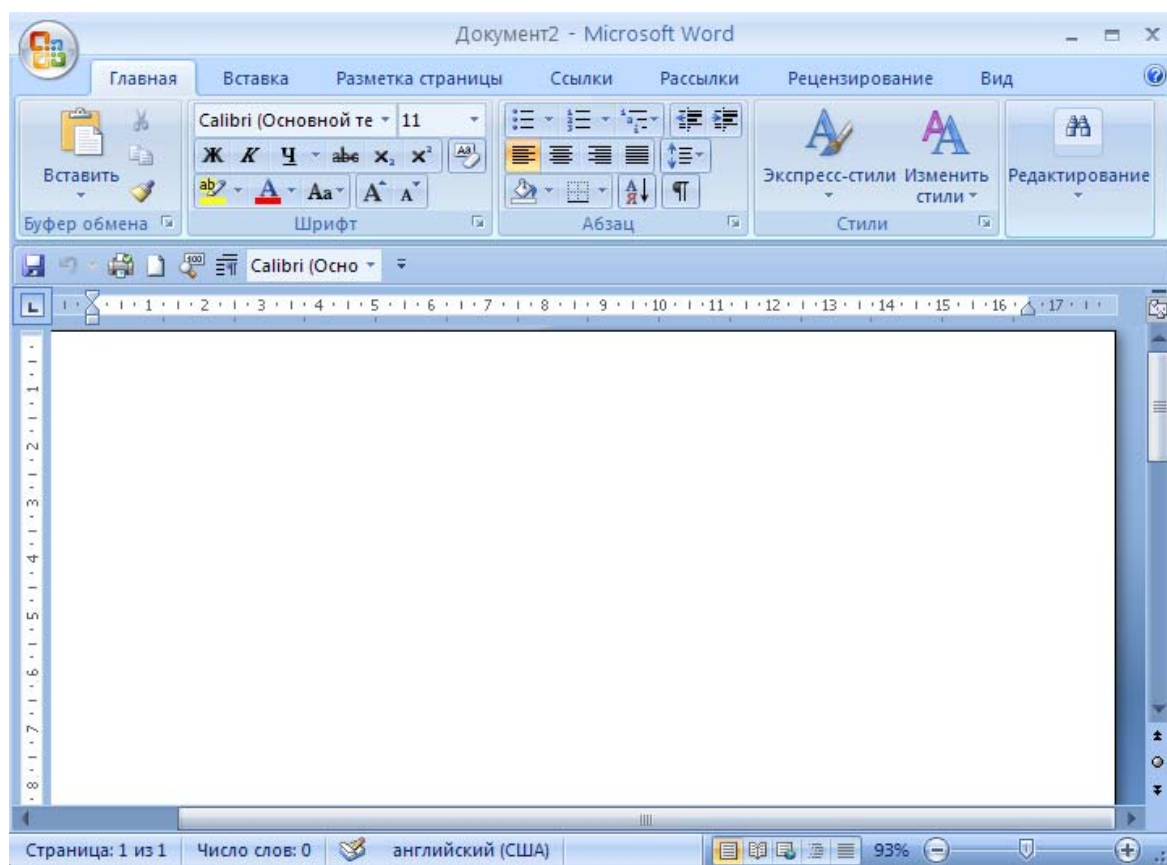


Рис. 1.1. Главное окно программы Microsoft Word

§2. Создание и сохранение нового документа

При запуске Microsoft Word автоматически создаётся новый документ на основе шаблона **Основной шаблон**. Вы можете соз-

дать любое количество новых документов, не закрывая предыдущих, и работать с этими документами одновременно.

Допустим, нужно создать новый документ.

1. Нажмите кнопку **Office**. На экране отобразится окно с перечнем команд.



(кнопка **Office**)

2. В появившемся окне выберете команду **Создать**. На экране появится окно **Создание документа** (рис. 1.2).

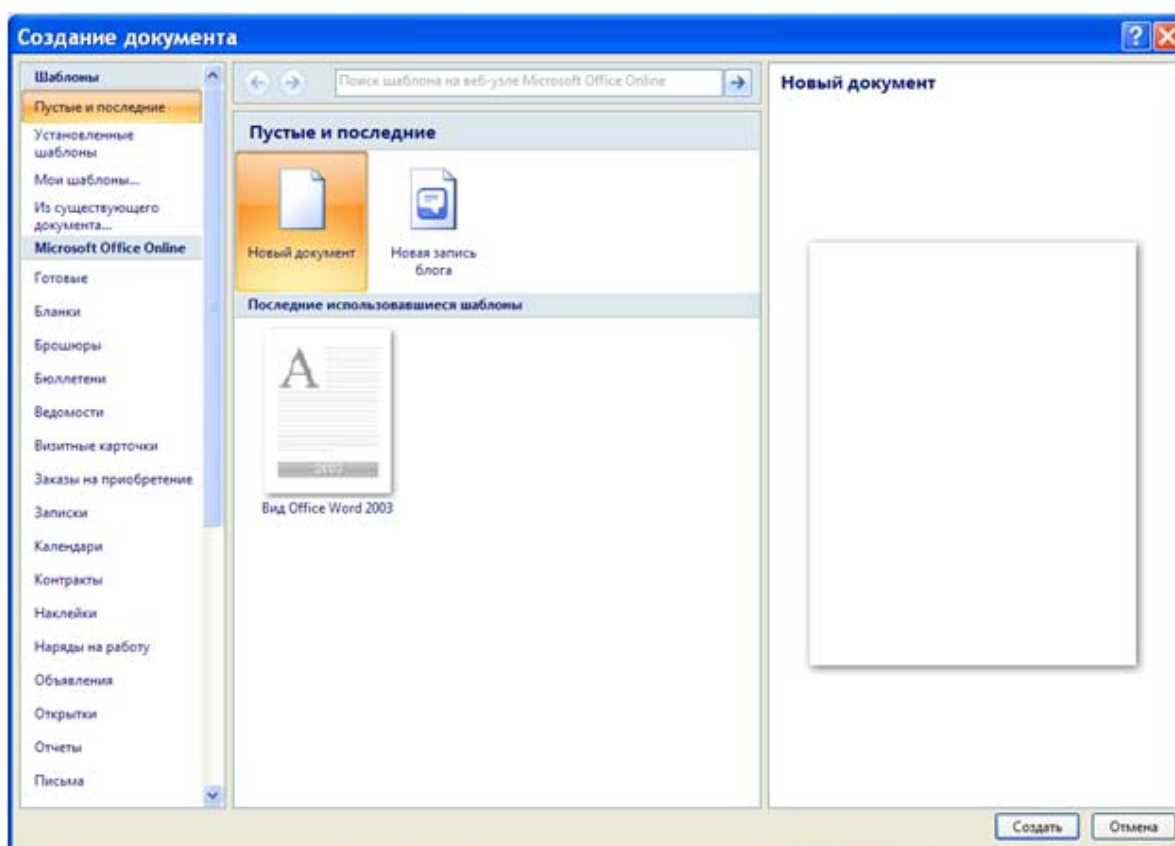


Рис. 1.2. Окно **Создание документа**

При создании документа вы можете воспользоваться готовым шаблоном, содержащим необходимую разметку и элементы оформления страницы.

В списке **Шаблоны** окна **Создание документа** расположены названия категорий шаблонов. В средней части окна отображается вид шаблонов, доступных в выбранной категории. В правой

части окна выводится выбранный шаблон в более крупном виде, а также некоторые его свойства.

По умолчанию окно **Создание документа** отображает шаблоны категории пустые и последние. В нем находится основной шаблон, обозначенный в окне как **Новый документ**, а также последние шаблоны, с которыми вы работали. Если ранее вы не работали с этой программой, то должны отображаться только два шаблона: **Новый документ** и **Новая запись блога**.

1. Последовательно выберите несколько других категорий шаблонов, например **Календари, Открытки, Счета**. Каждый раз при выборе новой категории вы будете видеть миниатюрные изображения страниц, создаваемых при использовании этих шаблонов.

2. Создайте документ с нуля: вернитесь в категорию **Пустые и последние**, выберите **Новый документ** и нажмите кнопку **Создать**. Окно **Создание документа** будет закрыто, а новый пустой документ будет создан.

Для быстрого создания документа с использованием основного шаблона вы можете воспользоваться сочетанием клавиш «Ctrl+N». Новый документ будет создан без открытия окна **Создание документа**.

Задание №1. Составьте резюме.

Пример готового резюме.

О себе: Я, Соколов Михаил Александрович. Родился 15 декабря 1990 г. в г. Москве.

О родителях Отец: Соколов Александр Владимирович. Профессия – инженер-геодезист. Мать: Соколова Евгения Олеговна – домохозяйка.

Об образовании: Окончил общеобразовательную школу №17 г. Москвы в 2007 г.

Какие интересы: Увлекаюсь спортом, люблю слушать современную музыку.

Почему решил поступать в ГУЗ: Хочу получить специальность геодезиста, чтобы работать по этой специальности, как мой отец.

Настройка параметров документа

Вы создали новый документ с использованием основного шаблона. В ряде случаев вам придётся изменить некоторые параметры документа и его страниц, например фоновый цвет, поля страниц или размер бумаги.

Размеры страницы

По умолчанию документ, созданный на основе шаблона **Новый документ**, адаптирован для последующей печати на бумаге формата А4 (21x29,7см). Но, возможно, вам понадобится создать документ для последующей печати на бумаге формата А3 или, например, А5, А6, а может и вовсе нестандартного формата. В этом случае вам необходимо изменить размеры страницы в программе Microsoft Word.

1. Перейдите на вкладку **Разметка страницы**. Нажмите кнопку **Размер**. На ленте отобразятся инструменты для изменения свойств страниц документа.

2. На экране отобразится список различных стандартных форматов страницы (рис. 1.3).

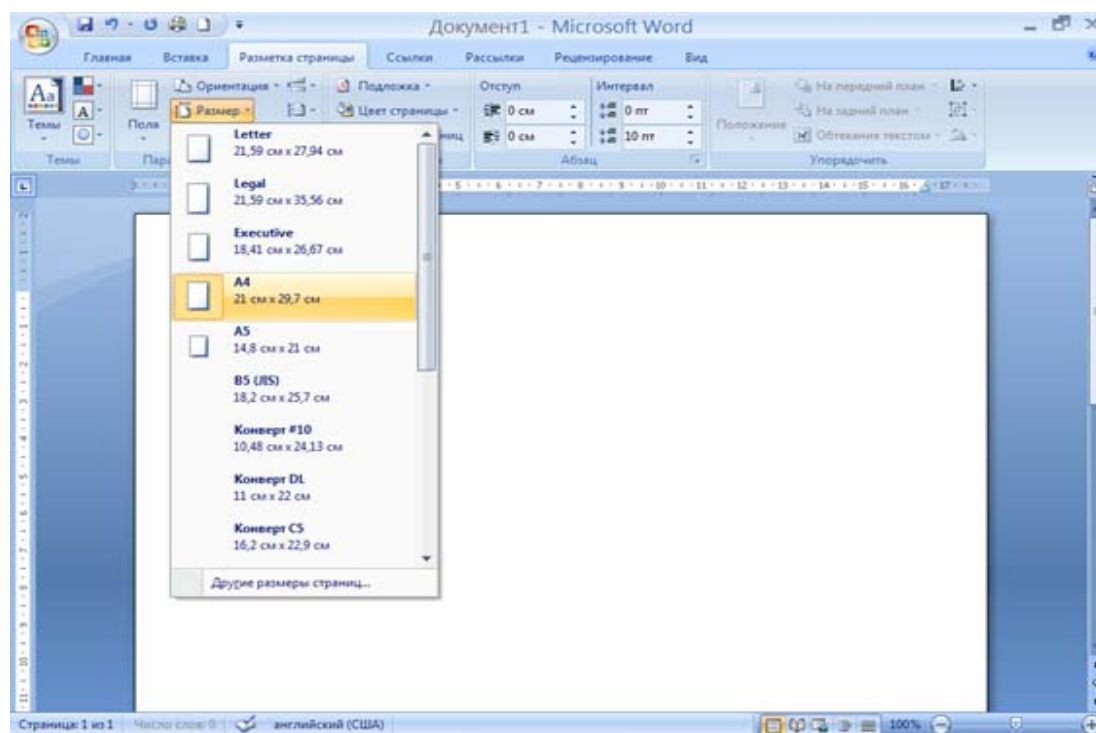


Рис. 1.3. Список стандартных форматов страницы

Щелкнув кнопкой мыши на любом из пунктов списка, вы немедленно преобразуете страницу документа в выбранный формат.

В данном случае не выбирайте стандартный формат страницы, а создайте собственный, чтобы увидеть, как это делается.

1. Щелкните кнопкой мыши на пункте **Другие размеры страниц**, расположенном в нижней части меню **Размер**. На экране появится окно **Параметры страницы**, открытое на вкладке **Размер Бумаги** (рис. 1.4).

В раскрывающемся списке области **Размер бумаги** вы также можете выбрать один из стандартных форматов. Вам нужно установить нестандартный размер страницы, поэтому ничего не выбирайте в этом списке.

2. В поле **Ширина** установите значение ширины страницы в сантиметрах, например 15.

3. В поле **Высота** установите значение высоты страницы, например 20.

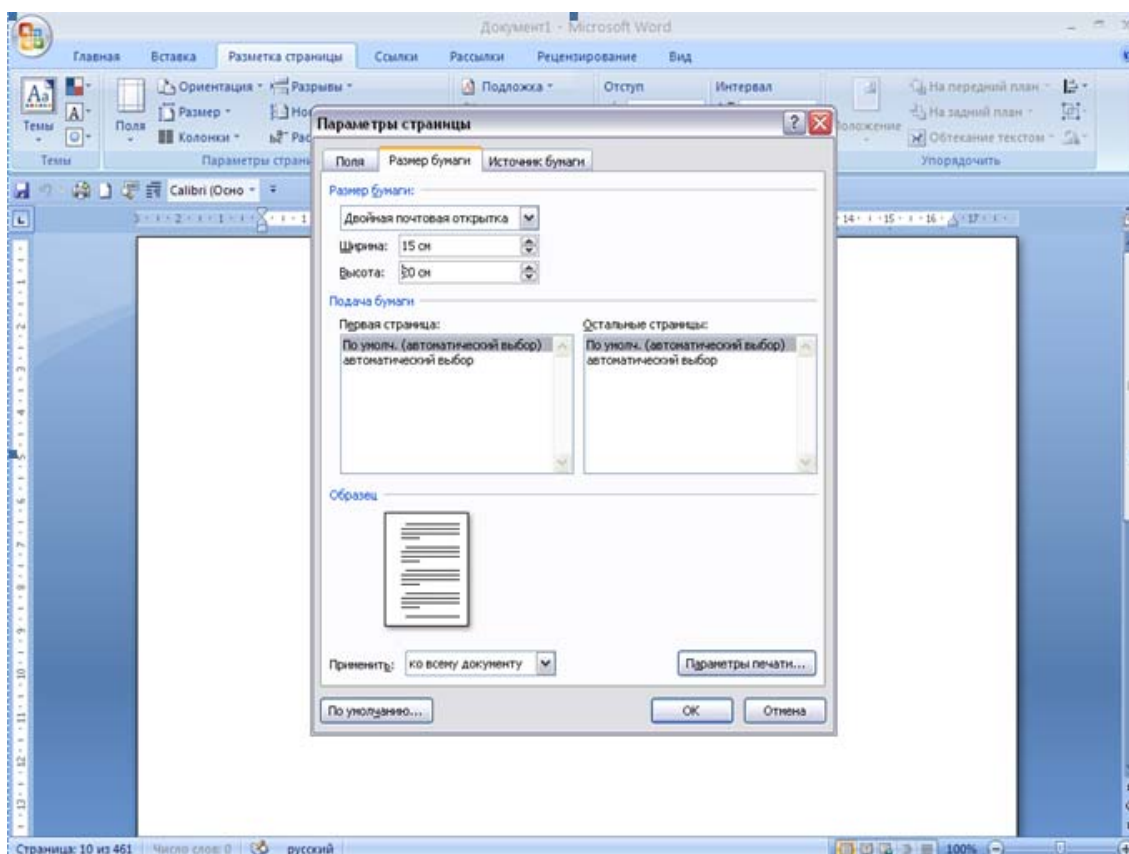


Рис. 1.4. Вкладка **Размер Бумаги** окна **Параметры страницы**

4. Нажмите кнопку **ОК** окна **Параметры страницы**. Окно будет закрыто, а размеры страницы изменятся в соответствии с вашими установками.

5. Перейдите на вкладку **Вид** и в группе **показать** или **скрыть** установите флажок **Линейка**. В верхней и левой частях страницы появятся линейки, отображающие размеры документа. Убедитесь, что размеры документа соответствуют заданным.

6. Теперь создайте документ со страницами формата А4. Вернитесь на вкладку **Разметка страницы**.

7. Нажмите на ленте кнопку **Размер** и в появившейся галерее выберите вариант А4 (21x27,9 см). Размеры страницы изменятся в соответствии с форматом А4.

Вместо выполнения шагов 6 и 7 вы можете просто нажать кнопку **Отмена** на панели быстрого доступа, чтобы отменить изменение размеров страницы.

Поля

Поля страницы представляют собой невидимые линии, являющиеся границами текста. Поля необходимы для правильного размещения текста в документе, а также для адаптации документа к свойствам принтера, который вы будете использовать для печати. Например, если текст в документе будет расположен от края до края страницы, то при последующей печати возможна потеря части информации, так как не все принтеры способны печатать на всю площадь листа. По этой причине по периметру документа необходимо создать отступы, называемые полями. Вы можете задать любой размер отступа в соответствии с вашими требованиями и особенностями принтера. Часто в левой части документа оставляют чуть большее поле для последующего переплёта или подшивки распечатанного документа.

Изменим размер поля нашего документа.

1. Убедитесь, что лента открыта на вкладке **Разметка страницы**, и нажмите кнопку **Поля**. На экране появится галерея предустановленных размеров полей (рис. 1.5)

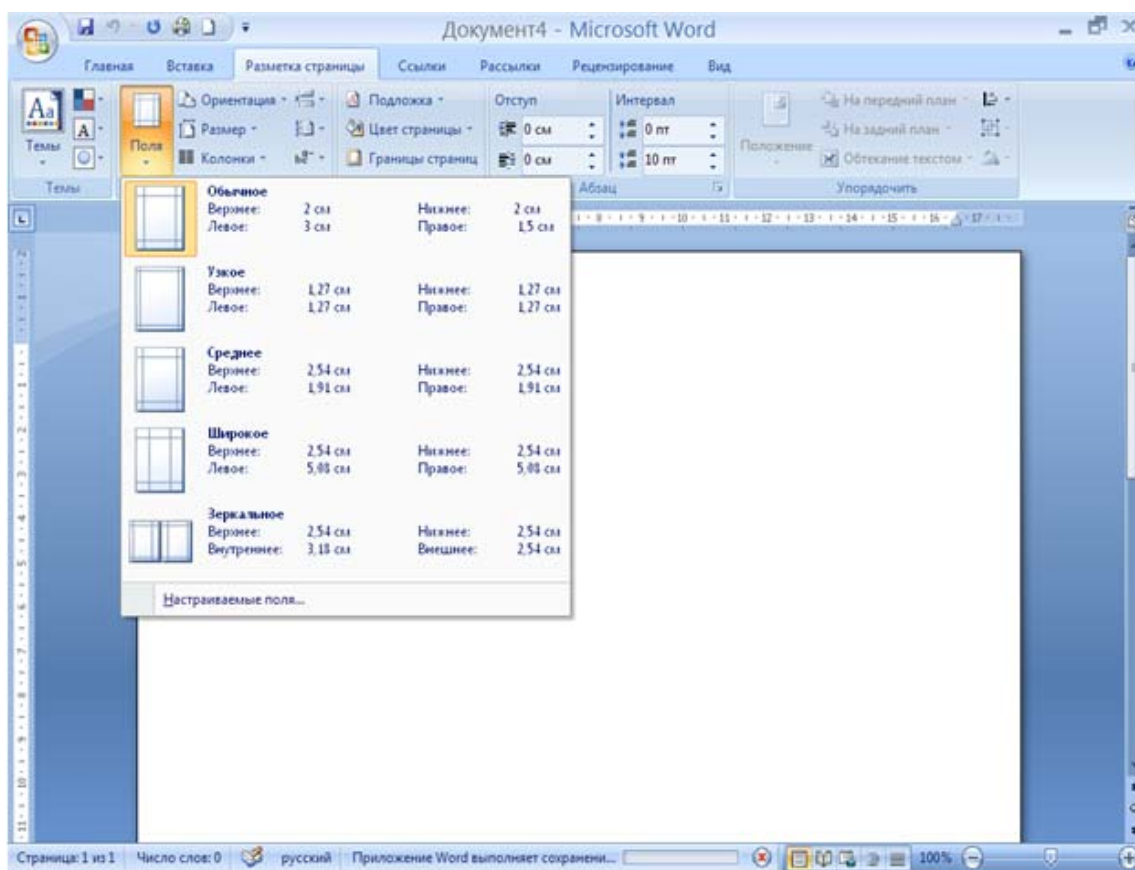


Рис. 1.5. Вкладка **Поля**

В данном случае не будем использовать предустановки, а определим размеры полей самостоятельно.

2. Щелкните кнопкой мыши на пункте **Настраиваемые поля**, расположенном после галереи предустановок размеров полей. На экране появится уже знакомое вам окно **Параметры страницы**, но открытое на вкладке **Поля** (рис. 1.6).

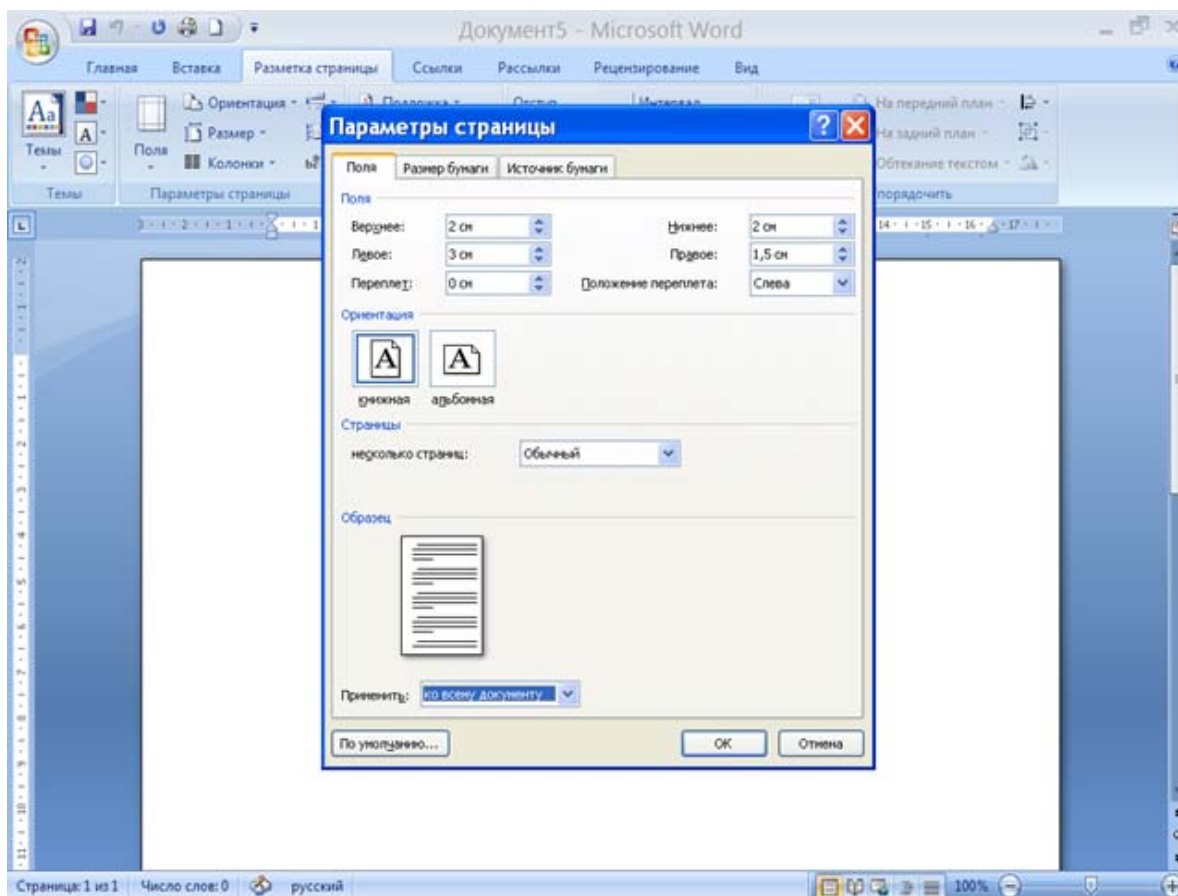


Рис. 1.6. Вкладка **Поля** окна **Параметры страницы**

3. В полях **Верхнее**, **Нижнее**, **Левое**, **Правое** установите размеры соответственно верхнего, нижнего, левого и правого полей документа. Вы можете выбрать любые устраивающие вас значения.

4. Нажмите кнопку **ОК**. Окно **Параметры страницы** будет закрыто, а размеры полей изменятся.

Обратите внимание на линейки, расположенные в верхней и левой части страницы. Вы видите тёмные и светлые области линеек. Тёмные области – это поля. Светлая область показывает доступное пространство для ввода текста. При вводе текста, как только символы достигнут правой границы документа (правого поля), набираемое на клавиатуре слово автоматически перенесётся на следующую строку, а при достижении нижней границы (нижнего поля) – на следующую страницу. При этом новая страница документа будет создана автоматически.

Вы можете быстро изменить размеры полей документа следующим образом. Установите указатель мыши между тёмной и

светлой областью линейки, чтобы указатель принял вид противоположно направленных стрелок и, нажав и удерживая левую кнопку мыши, перенесите границу на нужное расстояние.

Задание №2. Создайте текст длиной от двух до трёх страниц. Страницы должны быть пронумерованы (вкладка **Вставка** → **Номер страницы**). Установите верхний и нижний колонтитулы (вкладка **Вставка** → **Верхний колонтитул/Нижний колонтитул**). В верхнем колонтитуле напишите: Текст был написан (число. месяц. год.), а в нижнем колонтитуле напишите: Текст создал студент название факультета, № группы. Номер варианта может быть предоставлен вам преподавателем.

Номер варианта	Характер текста
1	Автобиография. Можете фантазировать. Не изменяйте только свою фамилию, имя и отчество
2	Резюме
3	Какая периодическая телепередача Вам нравится и за что. Характеристика участников этой передачи
4	Рассказ о вашем хобби
5	Содержание понравившегося Вам фильма или сериала
6	Какая музыкальная телепередача Вам нравится и за что. Характеристика участников этой передачи
7	Рассказ о вашей семье (родители, братья, сестры, жена (муж), дети). Можете фантазировать. Не изменяйте только свою фамилию, имя и отчество
8	Рассказ о том, как вы провели лето
9	Рассказ о любимом предмете в школе
10	Рассказ о любимом мультфильме
11	Рассказ о любимом кинофильме
12	Рассказ о любимой книге
13	Рассказ о любимом писателе
14	Рассказ о любимом блюде
15	Описание любимой картины
16	Рассказ о любимом времени года
17	Рассказ о поездке на дачу
18	Рассказ о поездке в другой город
19	Рассказ о любимом музее
20	Рассказ о друге/подруге
21	Рассказ о своей улице
22	Рассказ о моём посвящении в студенты 1-го сентября
23	Рассказ об отдыхе за границей
24	Рассказ о любимом домашнем животном
25	Рассказ о любимой музыкальной группе
26	Рассказ о любимом празднике

27	Рассказ о выпускном вечере
28	Рассказ о любимом виде спорта
29	Рассказ о любимой компьютерной игре
30	Рассказ о посещении театра

Ориентация страницы

Вы можете создавать документ как с вертикальным (книжная ориентация), так и с горизонтальным (альбомная ориентация) расположением страниц. Чтобы правильно планировать текст и размещать графические объекты, необходимо изменить отображение положения страницы на экране.

1. Нажмите кнопку **Ориентация**. На экране появится галерея, состоящая всего из двух пунктов: **Книжная** и **Альбомная**.

2. В появившейся галерее выберите нужную ориентацию страниц (книжную или альбомную).

Задание №3. Создайте документ, представляющий собою заявление. Стандартное заявление состоит из адреса, содержания и подписей.

Текст документа вводите по частям. Перед вводом каждой части выполняйте необходимые установки.

Последовательность создания документа

1. Установите шрифт **Arial**, размер 16, полужирный. Технология установки шрифта: на вкладке **Главная** выберите Вид шрифта и размер шрифта (рис. 1.7).



Рис. 1.7. Окна для выбора шрифта
(наименование шрифта и размер, показанные в окнах, могут быть другими)

В правой части окон находятся кнопки . При щелчке по кнопке, расположенной в окне *Шрифт*, появится список шрифтов, снабженный полосой прокрутки, облегчающей перемещение по списку. Наименования шрифтов расположены по алфавиту. Найдите шрифт **Arial** и щелкните по его имени левой клавишей

мышь. Имя копируется в окно *Шрифт*. Аналогично установите размер шрифта **16**.

Программа Microsoft Word содержит инструменты для автоматического создания отступа абзаца. Рассмотрим создание такого отступа.

Текст:

Декану заочного факультета
(*фамилия и.о. декана*)
от студента группы (№ группы)
(*фамилия и.о. студента*)

Приведенный выше текст обычно предваряет заявление. Будем называть этот текст «адресом».

1. Наберите первую строчку и выделите её. Установите указатель мыши на верхнем маркере, расположенном в левой части линейки. На экране появится всплывающая подсказка **Отступ первой строки**. Установите верхний маркер, сдвинув его на сколько возможно вправо (рис. 1.8).

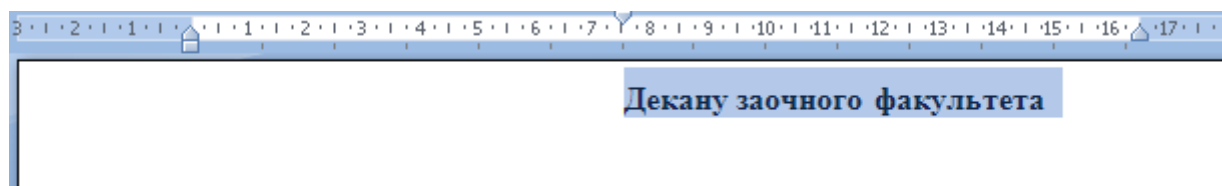


Рис. 1.8. Отступ абзаца создан с помощью маркера линейки

2. Набор каждой строки адреса завершайте нажатием клавиши **Enter**.

3. Далее установите автоматический перенос текста. Для этого зайдите на вкладку **Разметка страницы** и включите режим установки переносов (рис. 1.9).

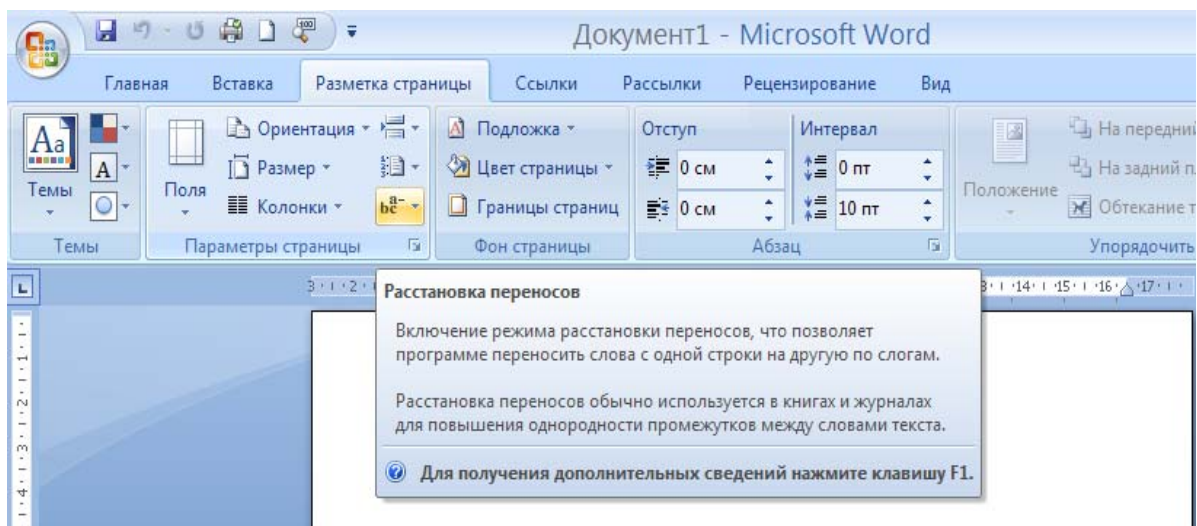


Рис. 1.9. Установка режимов переноса текста

Фамилию И.О. декана, группу и свою фамилию укажите фактические. При этом используйте тот же шрифт, что и в первой строке обращения.

5. После ввода адреса пропустите одну строку нажав клавишу **Enter**, установите выравнивание «по центру», проверьте сохранились ли выполненные вами раньше установки шрифта и введите слово «Заявление».

6. После ввода слова «Заявление» отключите жирный шрифт, остальные установки шрифта оставьте прежними. Пропустите строку, установите отступ первой строки клавишей «Tab», способ выравнивания «по ширине» и введите текст заявления (текст см. ниже). При вводе не используйте клавишу **Enter**, так как текст заявления представляет собою один абзац. Переход со строки на строку должен происходить автоматически. Абзац заканчивается на слове «прошёл». После него (точнее, после точки) и следует нажать на клавишу **Enter**.

Текст заявления:

Прошу разрешить мне досрочную сдачу экзамена по курсу «Информатика». С программой курса ознакомился. Все задания и контрольные работы выполнил. Собеседование с преподавателем (*фамилия и.о. преподавателя*) прошел.

Завершение ввода документа

Пропустите строку, проверьте установки шрифта, они должны быть такими же, какими вы их установили в начале работы над

документом. Установите способ выравнивания «по левому краю» и введите последнюю часть документа (см. ниже). Отступы первой строки для приведенных выше элементов текста установите примерно такими же, как показано выше.

Тот, кто умеет это делать, может поместить в заявление даты в режиме вставки. Поставить подпись в заявлении, можно расписавшись в программе Paint, а затем, предварительно вырезав подпись, вставить её в заявление.

Ниже показан документ в том виде, в каком он должен у Вас получиться:

Декану заочного факультета
(фамилия и.о. декана)
от студента группы (имя группы)
(фамилия и.о. студента)

Заявление

Прошу разрешить мне досрочную сдачу экзамена по курсу “Информатика”. С программой курса ознакомился. Все задания и контрольные работы выполнил. Собеседование с преподавателем (фамилия и.о. преподавателя) прошел.

(День. Месяц. Год.)

Ваша подпись

(Ф.И.О. фамилия студента)

Против досрочной сдачи экзамена не возражаю.

(День. Месяц. Год.)

Ваша подпись

(Ф.И.О. фамилия преподавателя)

Задание №4. Наберите заявления по вариантам.

Номер варианта	Характер заявления
1	О предоставлении очередного отпуска
2	О предоставлении отпуска без содержания
3	Об увольнении без указания причины
4	Об увольнении, в связи с достижением пенсионного возраста
5	Об увольнении в связи с отъездом на постоянное жительство в др. город

6	Об увольнении в связи с отъездом на постоянное жительство за рубеж
7	О предоставлении отпуска по уходу за ребенком
8	Объяснительную записку по поводу прогула
9	О допуске к вступительным экзаменам в вуз
10	О получении загранпаспорта
11	О предоставлении ребенку питания в школе
12	О предоставлении мед. книжки
13	О предоставлении водительского удостоверения
14	О проведении экспертизы бытовой техники
15	О предоставлении материальной помощи
16	О предоставлении места в детском саду
17	О подаче судебных документов
18	О регистрации сделки
19	Явка с повинной
20	О вступлении в наследство
21	О выдаче паспорта
22	О незаконном строительстве на земельном участке
23	О предоставлении академического отпуска
24	О принятии в члены садового товарищества
25	О предоставлении ребёнку путёвки в лагерь
26	О предоставлении пенсии
27	О предоставлении проездного билета на метро
28	Об усыновлении ребёнка
29	О предоставлении регистрации
30	О прохождении медкомиссии

В процессе работы вам придётся сохранять создаваемый документ для продолжения работы с ним через какой-либо период времени.

Чтобы сохранить документ, выполните следующие действия (вы можете сохранить даже пустой документ).

1. Нажмите кнопку **Office** и в появившемся списке выберите команду **Сохранить как**. На экране появится окно **Сохранение документа** (рис. 1.10.).

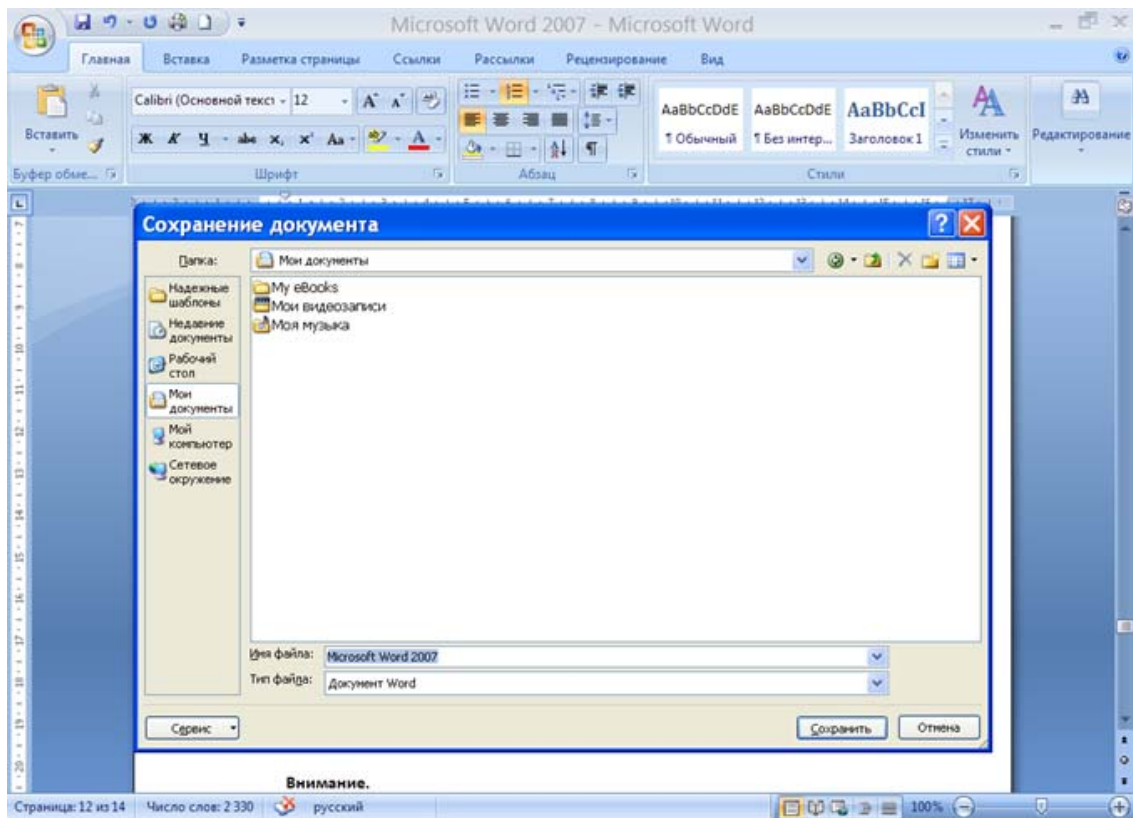


Рис. 1.10. Окно Сохранение документа

2. В появившемся окне выберите папку, в которой вы хотите сохранить файл документа.
3. В поле **Имя файла**, введите имя файла сохраняемого документа.
4. Если вы хотите сохранить документ в формате программы Microsoft Word, то убедитесь, что в открывающемся списке **Тип файла** выбрано значение **Документ Word**.

Если в дальнейшем вы планируете загружать сохранённый документ в более ранние версии программы Microsoft Word, то установите флажок **Поддерживать совместимость с Word 97 – 2003**. В противном случае вы сможете загрузить сохранённый документ только в программу Microsoft Word версии 2007.

5. Нажмите кнопку **Сохранить** окна **Сохранение документа**.
Окно будет закрыто, а файл документа сохранён.

Единожды сохранив документ и дав файлу документа имя, в дальнейшем вы можете быстро сохранять его, нажимая кнопку **Сохранить** на панели быстрого доступа или сочетанием клавиш

«Shift+F12». Документ будет сохранён в ранее указанной папке под тем же именем. При этом предыдущая копия файла будет удалена.

Если вы хотите хранить все версии документа с внесёнными изменениями, то сохраняйте эти версии под разными именами.

Вы можете сохранить документ в другом формате. Программа Microsoft Word позволяет экспортировать содержимое документа в файлы различных форматов: Microsoft Works, Word Perfect, Rich Text Format (RTF), Обычный текст и др.

При экспорте содержимого в другой формат следует учитывать, что некоторые свойства документа могут быть изменены, а также может быть потеряна часть информации. Например, если вы экспортируете документ, содержащий форматирование и графику, в формат обычного текста, то в сохранённом TXT-файле будут отсутствовать и графика, и форматирование.

Программа Microsoft Word обладает функцией автоматического сохранения создаваемого документа. Представьте следующую ситуацию. Вы создаёте большой документ в течение нескольких часов работы. Если вы периодически не нажимаете кнопку **Сохранить**, то можете потерять результат нескольких часов работы в случае сбоя в работе компьютера. Функция автоматического сохранения позволит избежать эту неприятность.

1. Нажмите кнопку **Office**. На экране появится список доступных команд.
2. Нажмите кнопку **Параметры Word** в нижней части списка команд. На экране появится одноименное окно.
3. Выберите категорию **Сохранение** в списке, расположенном в левой части окна **Параметры Word**. Содержание окна изменится (рис. 1.11).

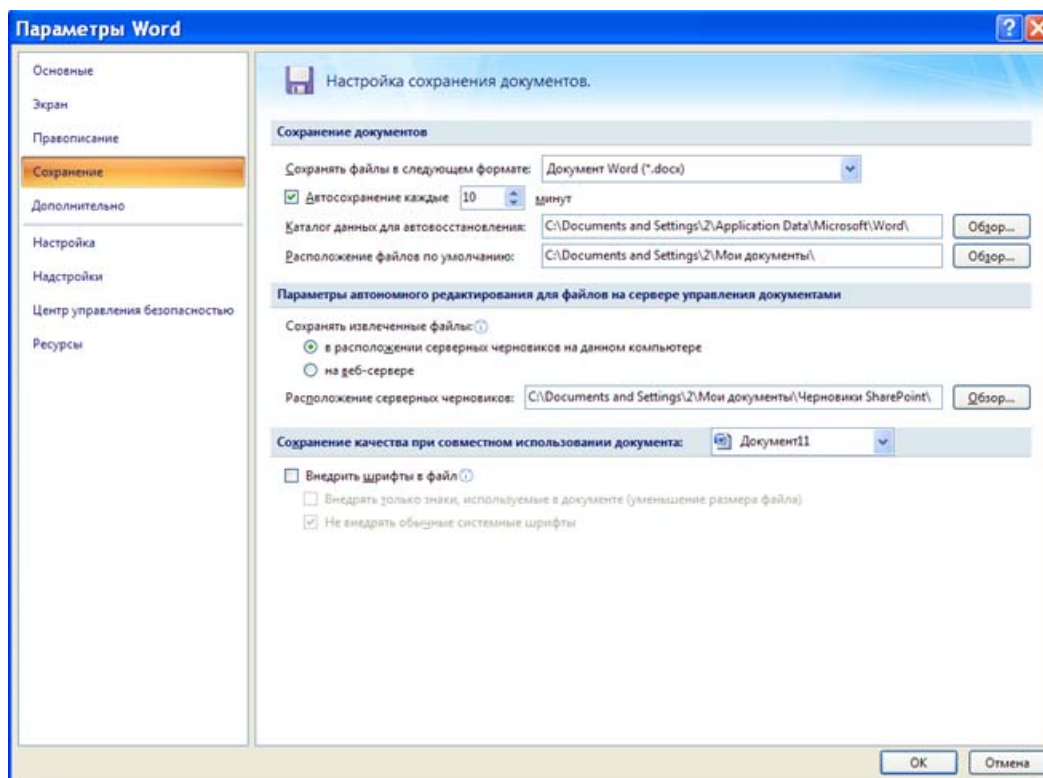


Рис. 1.11. Категория **Сохранение** окна **Параметры Word**

4. Установите флажок **Автосохранение каждые**, а в поле минут, расположенном правее, установите периодичность сохранения.

Таким образом, установив периодичность сохранения, например, пять минут, вы при непредвиденном сбое потеряете в худшем случае результат работы за последние пять минут (каждые пять минут программа Microsoft Word автоматически будет сохранять ваш документ). Установив меньший интервал сохранения, например одну минуту, вы ещё более сократите объём потерянной информации при сбое системы.

5. Нажмите кнопку **ОК**. Окно будет закрыто, а изменения применены.

Следует учитывать, что изменения, сделанные в окне **Параметры Word**, применяются не к текущему документу, а к программе и, следовательно, ко всем создаваемым впоследствии документам. Иными словами, при создании следующего документа вам не придётся заново настраивать параметры автосохранения.

Первоначальное сохранение документа на съёмном диске или на дискете, а не на жестком диске, выполняется только тогда, когда вы намерены покинуть компьютер, ни разу до этого не сохранив документ на жестком диске. Признаком того, что операция сохранения документа еще не выполнялась, является его имя: «Документ1», «Документ2», «Документ3» и т.д.

Повторные сохранения того же документа на дискете так же, как и повторные сохранения на жестком диске, диалога не требуют.

§3. Загрузка и редактирование документа

В процессе работы вам, возможно, придётся загружать в программу Microsoft Word ранее созданные документы, а также текстовые документы, созданные в других программах. Чтобы загрузить в программу документ, необходимо выполнить следующие действие.

1. Нажмите кнопку **Office**. На экране появится список доступных команд.
2. В появившемся списке выберете команду **Открыть**. На экране появится окно **Открытие документа**. (рис. 1.12).
3. В окне **Открытие документа** перейдите в папку, содержащую файл, который вы хотите загрузить, затем выберите файл и нажмите кнопку **Открыть**. Окно будет закрыто, а файл загружен в программу.

Вышеуказанным способом вы можете загрузить только документы, созданные в программе Microsoft Word с расширением DOC и DOCX. Окно **Открытие документа** содержит фильтр, скрывающий файлы, имеющие другие расширения. Однако вы можете изменить настройку фильтра, чтобы отобразить файлы других форматов.

В качестве примера загрузим в программу Microsoft Word текстовый файл с расширением TXT.

1. Нажмите кнопку **office** и в появившемся списке выберите команду **Открыть**. На экране появится окно **Открытие документа**.
2. В окне **Открытие документа** перейдите в папку, содержащую текстовый файл, который хотите загрузить.
3. В раскрывающемся списке **Тип файлов** выберите нужный тип файла. В данном примере это пункт **Текстовые файлы**. Фильтр окна будет изменён.

Теперь файлы с расширением TXT будут отображаться в списке файлов окна **Открытие документов**.

4. Выберите нужный текстовый файл и нажмите кнопку **Открыть**. Окно будет закрыто, а файл – загружен в программу.

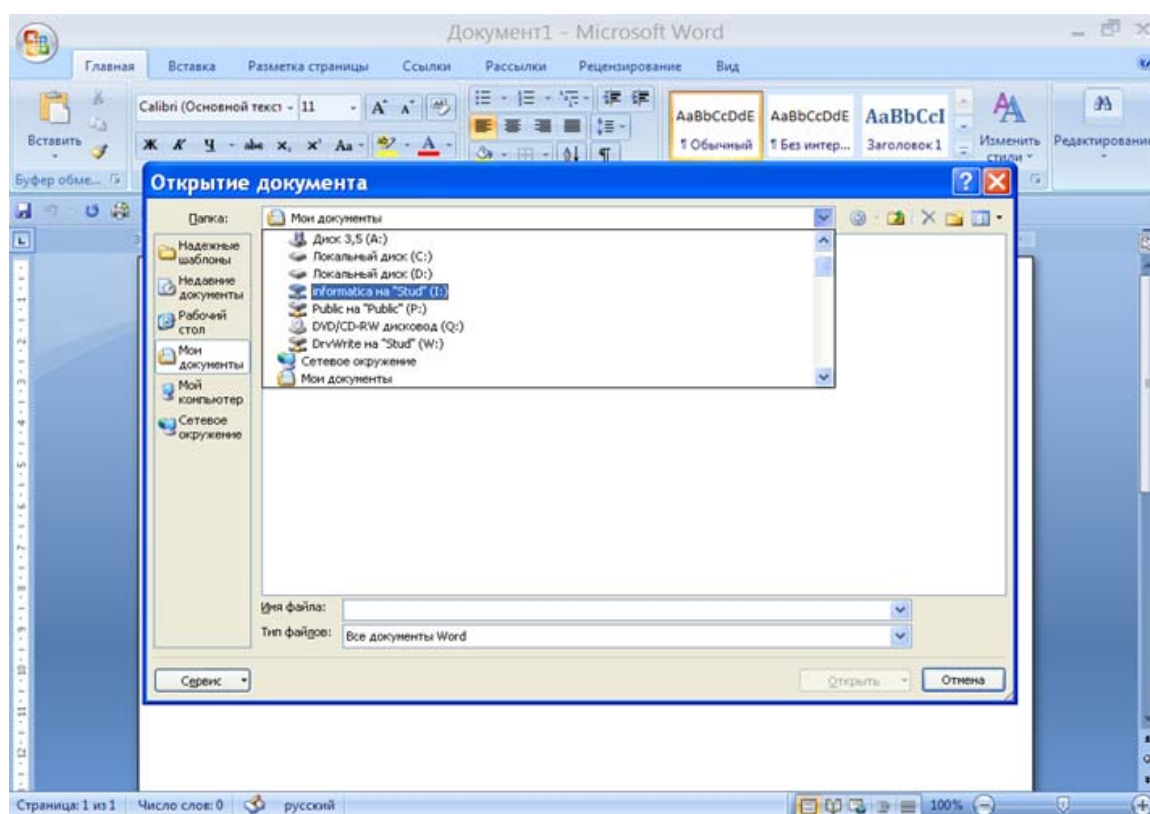


Рис.1.12. Окно **Открытие документа**

Программа Microsoft Word позволяет загружать файлы документов самых разнообразных форматов. Вы можете убедиться в

этом, посмотрев содержимое открывающегося списка Тип файлов окна Открытие документа.

Редактирование текста

Понятие редактирование происходит от латинского слова «redactus» - приведенный в порядок. Редактирование – это процесс изменения содержания документа, а именно: добавление, удаление и перемещение фрагментов текста (слов, предложений и абзацев).

Программа Microsoft Word предназначена в первую очередь для создания текстовых документов, поэтому умение быстро вводить и корректировать текст поможет вам создавать документы в минимальные сроки.

Абзац – это фрагмент текста, как правило, выделенный отступом в первой строке. Абзац обычно несет в себе определенную законченную логическую мысль, которую автор желает выделить. Текст, разбитый на абзацы, легче читается. При создании больших документов крайне желательно научиться правильно, делить текст на абзацы, чтобы придать документу удобочитаемый вид. При вводе текста, если вы нажимаете клавишу **Enter**, то переносите курсор на следующую строку и создаете начало нового абзаца. При вводе текста абзац будет создаваться до тех пор, пока вы снова не нажмете клавишу **Enter**, чем закончите создание текущего абзаца и обозначите начало следующего. В пределах абзаца не переносите текст вручную – он должен переноситься автоматически.

Для выделения абзацев наиболее часто применяется метод отступа первой строки. Как правило, начало первой строки абзаца расположено не у левого поля документа, а чуть смещено вправо (рис. 1.13).

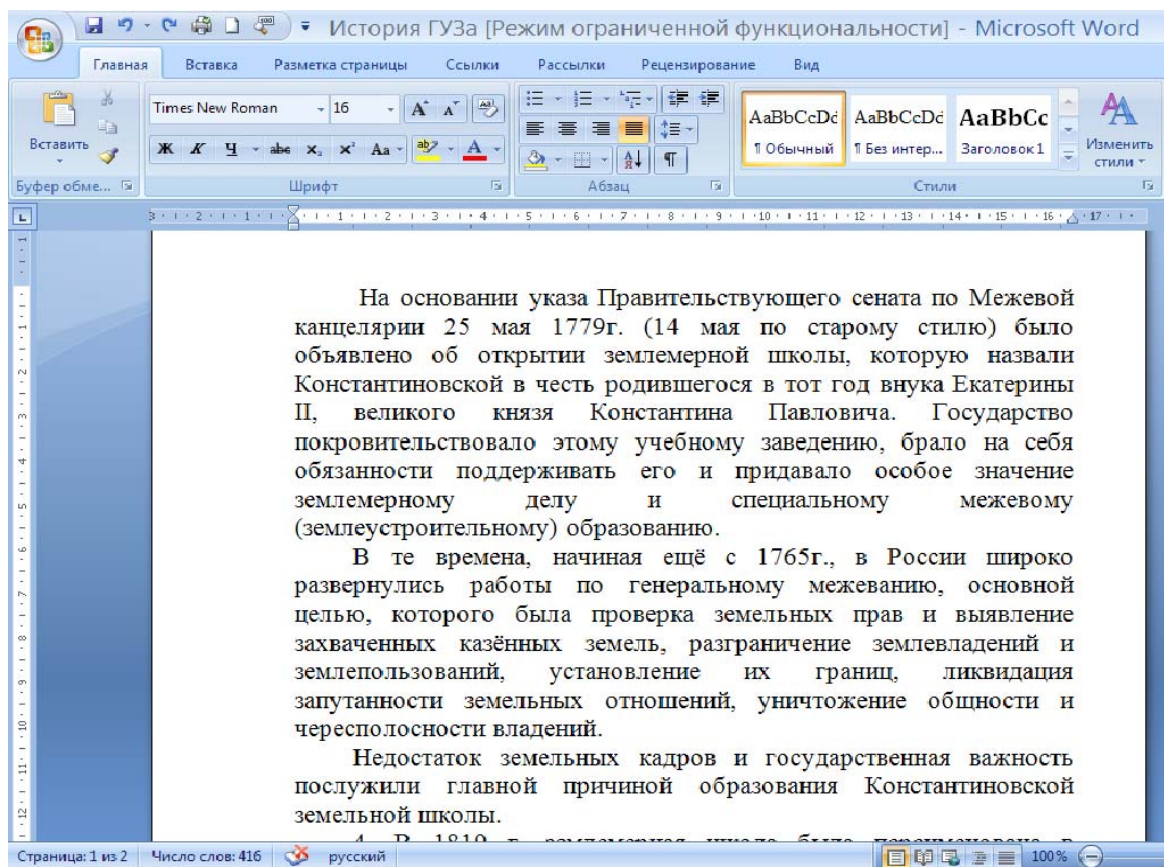


Рис. 1.13. Отступ в начале первой строки зрительно отделяет абзац от предыдущего.

Программа Microsoft Word содержит инструменты для автоматического создания отступа первой строки абзаца. Рассмотрим метод создания такого отступа.

1. Введите какой-либо текст документа и разбейте его на несколько абзацев.
2. Убедитесь, что в левой и верхней частях страницы документа отображаются линейки. Если линейки не отображаются, перейдите на вкладку **Вид** и в группе **Показать или скрыть** установите флажок **Линейка**. В левой и правой частях верхней линейки (на границах полей) расположены треугольные маркеры. Эти маркеры служат для быстрой установки отступа первой строки и остального текста абзаца.
3. Установите курсор в любую позицию абзаца. Для этого щёлкните кнопкой мыши на любом слове в пределах абзаца.
4. Установите указатель мыши на верхнем маркере, расположенном в левой части линейки. На экране появится всплывающая подсказка **Отступ первой строки**.

5. Нажав и удерживая левую кнопку мыши, переместите маркер отступа первой строки на 1-2 см вправо. Начало первой строки абзаца будет смещено (рис. 1.14).

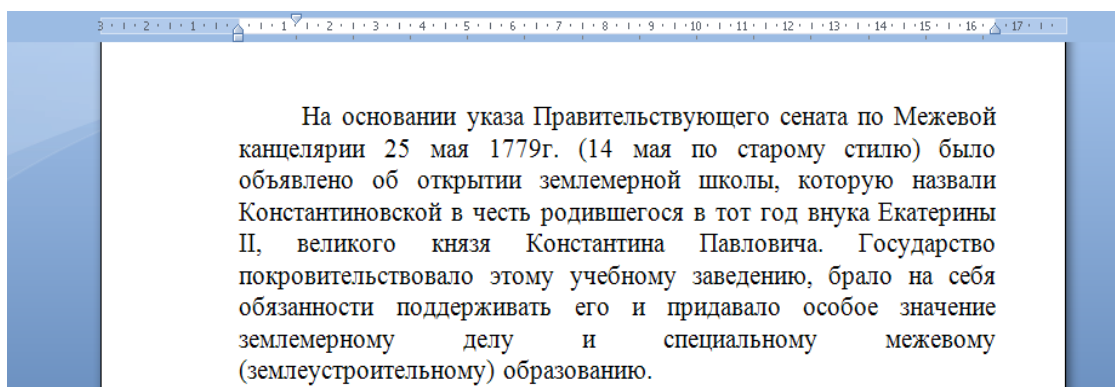


Рис. 1.14. Отступ первой строки абзаца на 1 см создан с помощью маркера линейки.

Нижними маркерами, расположенными в левой и правой частях линейки, вы можете изменить размер отступа остальных строк абзаца соответственно от левого и правого полей. При создании отступа первой строки с помощью маркера это изменение коснулось лишь того абзаца, на котором установлен курсор. Если создаваемый документ достаточно объемный, такой способ изменения отступа может показаться не самым удобным. Но, нажав **Ctrl+A**, вы сможете выделить весь текст, а затем переместить маркеры так, как требуется. Изменения будут применены ко всему тексту.

При чтении различных текстов вы можете заметить, что абзацы отделены друг от друга интервалом, превышающим интервал между строками внутри абзаца. Это также вносит определенное удобство при чтении текста. Для установки Поля **Интервал до** и **Интервал после** в области **Интервал** должна быть выбрана вкладка **Разметка страницы**. В поле **Интервал до** вы можете изменить интервал между выделенным абзацем (в котором находится курсор) и вышестоящим абзацем. В нижнем поле **Интервал после** вы можете изменить интервал между выделенным и нижестоящим абзацем. Изменяя значение в полях **Интервал до** и **Интервал после**, вы устанавливаете не новое значение интервала, а величину, на которую следует увеличить обычный межстрочный интервал. Чтобы установить интервал между абзацами

во всем тексте, вам также придется выделить весь текст, нажав сочетание клавиш **Ctrl+A**, и установить нужный интервал между всеми абзацами текста. Если вы устанавливаете значение интервала между абзацами в пустом документе, то во время ввода текста интервал с заданным вами значением будет автоматически создаваться при нажатии клавиши **Enter**.

В процессе создания документа вы будете допускать ошибки, опечатки, а возможно, и изменять содержимое документа, заменяя слова, фразы или отдельные предложения другими. Рассмотрим несколько способов удаления фрагментов текста.

1. Установите курсор слева от символа, который хотите удалить.
2. Нажмите клавишу **Delete**. Символ будет удален, а курсор окажется слева от следующего в слове символа.
3. При повторном нажатии клавиши **Delete** будет удален следующий символ, слева от которого установлен курсор.
4. Нажмите клавишу **Backspace**. Символ, стоящий слева от курсора, будет удален.

Данный пример демонстрирует, что действие клавиш **Delete** и **Backspace**, похоже, только при нажатии клавиши **Delete** удаляется символ, расположенный справа от курсора, а при нажатии клавиши **Backspace** – слева от курсора.

Программа Microsoft Word содержит удобные интеллектуальные алгоритмы выделения текста. Ранее вы выделяли слово в тексте путем перетаскивания указателя мыши при нажатой левой кнопке. Есть более удобные способы.

- Дважды щелкните кнопкой мыши на любом слове в тексте. Все слово будет выделено.
- Щелкните кнопкой мыши на любом слове в тексте, удерживая нажатой клавишу «**Ctrl**». Будет выделено все предложение.
- Трижды щелкните кнопкой мыши на любом слове. Будет выделен весь абзац.
- Установите указатель мыши на левое поле (за левой границей текста) и нажмите кнопку мыши. Будет выделена вся строка, расположенная правее указателя мыши.
- Установите курсор в начале выделяемого фрагмента. Прокрутите документ до конца выделяемого фрагмента и, нажав

и удерживая клавишу «Shift», щелкните кнопкой мыши на окончании выделяемого фрагмента. Весь фрагмент будет выделен. Такой способ удобен для выделения больших участков текста.

- Нажав и удерживая левую кнопку мыши и клавишу «Alt», перетащите указатель мыши по диагонали. Будет выделен прямоугольный фрагмент текста (рис. 1.15).

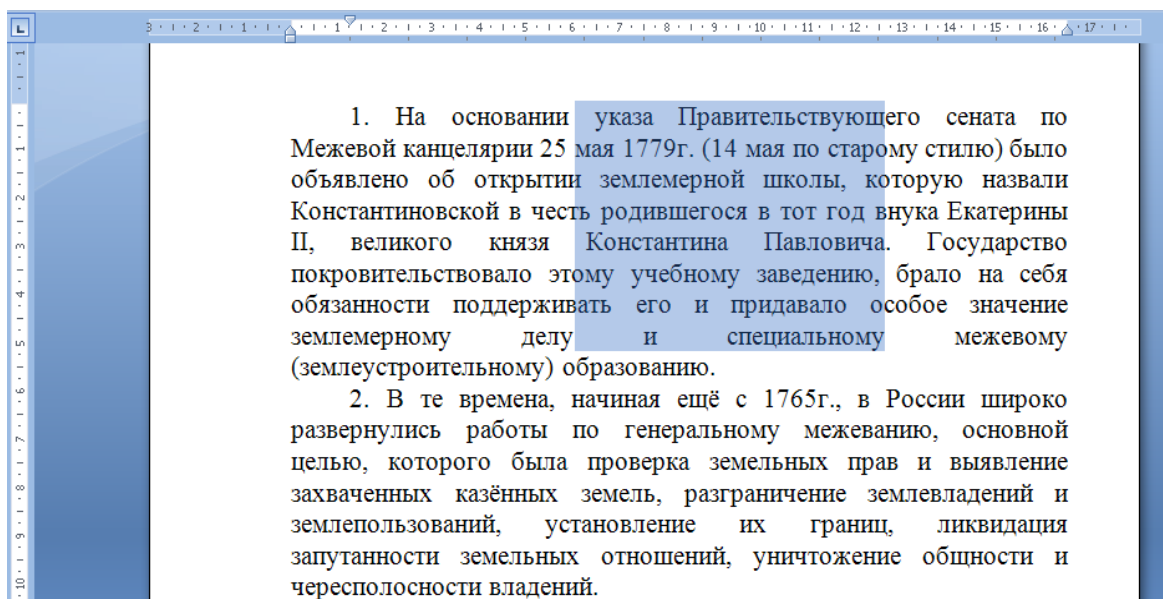


Рис. 1.15. Выделение прямоугольного фрагмента

Выделение фрагмента текста необходимо не только для удаления и замены содержимого документа. В процессе работы с документом вам придется постоянно выделять различные фрагменты, чтобы изменить их свойства: размер, цвет шрифта, выравнивание, стиль и т.д.

Буфер обмена Microsoft Word 2007 предусматривает хранение до 24 фрагментов текста, графики и других данных. Иными словами, буфер обмена программы Microsoft Word содержит 24 ячейки, в каждую из которых вы можете поместить какой-либо набор данных и брать его оттуда по мере необходимости.

Работа с буфером обмена предусматривает выполнение одной из трех операций:

1. вырезать – выделенный фрагмент текста копируется в буфер обмена и удаляется из текста;
2. копировать – выделенный фрагмент копируется в буфер обмена, но при этом остается в тексте;

3. вставить – в текст вставляется ранее вырезанный или скопированный фрагмент.

Вышеперечисленные операции применимы не только к тексту, но и к другим типам данных: графике, видеофайлам, звуку и т.д.

Рассмотрим несколько примеров работы с буфером обмена.

1. Выделите любое слово или словосочетание в документе.
2. Нажмите правую кнопку мыши и в появившемся контекстном меню выберите команду **Копировать**. Вы также можете нажать одноименную кнопку на вкладке **Главная** или словосочетание клавиш «Ctrl+C» (Ctrl+Insert).
3. Установите курсор в позицию документа, в которую хотите вставить скопированный фрагмент.
4. Нажмите правую кнопку мыши и в появившемся контекстном меню выберите команду **Вставить**. Вы можете нажать кнопку **Вставить**, расположенную на вкладке **Главная**, или комбинацию клавиш «Shift+Insert». Ранее скопированное слово или словосочетание будет вставлено в позицию курсора.
5. Установите курсор в новую позицию документа и снова нажмите кнопку **Вставить** или словосочетание клавиш «Shift+Insert». Ранее скопированное слово или словосочетание снова будет вставлено в позицию курсора.
6. Таким образом, вы можете вставлять ранее скопированный фрагмент в любую позицию документа неограниченное количество раз.

В данный момент в буфере обмена программы Microsoft Word содержится один скопированный вами фрагмент. Давайте скопируем в буфер еще несколько фрагментов, чтобы научиться вставлять в текст именно нужный фрагмент.

1. Последовательно скопируйте несколько разных фрагментов текста в буфер обмена.
2. Перейдите на вкладку **Главная**.
3. Нажмите кнопку, расположенную в правой нижней части группы **Буфер обмена**. В левой части окна Microsoft Word появится список содержимого буфера обмена.
4. Установите курсор в позицию, в которую хотите вставить текст из буфера обмена. (рис. 1.16).

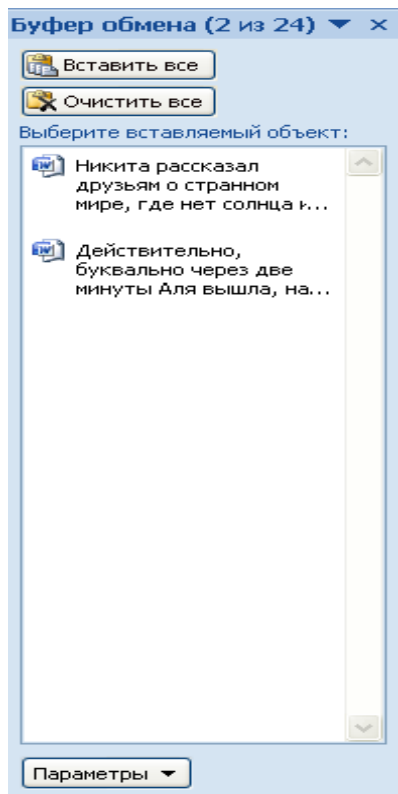


Рис. 1.16. Окно буфера обмена

5. Установите указатель мыши на тот пункт в списке содержимого буфера обмена, текст которого хотите вставить. Справа от пункта появится кнопка с изображением стрелки.
6. Нажмите кнопку. На экране появится контекстное меню (рис. 1.17).

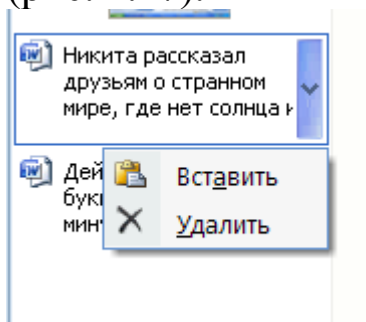


Рис. 1.17. Окно контекстного меню

7. В появившемся контекстном меню выберите команду **Вставить**. Содержимое выбранной ячейки буфера обмена будет вставлено в текст.
8. Установите курсор в другую позицию и вставьте в это место содержимое другой ячейки буфера обмена.

Таким образом, вы можете вставлять содержимое разных ячеек буфера обмена в разные места документа.

Вам нет необходимости удалять содержимое ячейки буфера обмена. Вы можете сделать это, когда фрагмент, содержащийся в данной ячейке, вам больше не будет нужен. Если вы скопируете в буфер обмена более 24 фрагментов, последние 24 останутся в буфере, а фрагменты, которые были скопированы первыми, удалятся. Иными словами, запись данных в буфер обмена происходит циклически. Если весь буфер обмена заполнен, то при записи очередного фрагмента из буфера стирается самый старый фрагмент.

Задание №5. Набрать и скопировать

1. Все 10 абзацев нижеприведенного текста в буферы обмена.
2. Вставить из буфера обмена абзацы в следующем порядке:

Абзац 1
Абзац 3
Абзац 5
Абзац 7
Абзац 9
Абзац 10
Абзац 9
Абзац 8
Абзац 7
Абзац 6
Абзац 5
Абзац 4
Абзац 3
Абзац 2
Абзац 1

Номер подгруппы. Фамилия, имя, отчество студента.

1. На основании указа Правительствующего сената по Межевой канцелярии 25 мая 1779 г. (14 мая по старому стилю) было объявлено об открытии землемерной школы, которую назвали Константиновской в честь родившегося в тот год внука Екатерины II, великого князя Константина Павловича. Государство покровительствовало этому учебному заведению, брало на себя обязанности поддерживать его и придавало особое значение зем-

лемерному делу и специальному межевому (землеустроительному) образованию.

2. В те времена, начиная ещё с 1765 г., в России широко развернулись работы по генеральному межеванию, основной целью, которого была проверка земельных прав и выявление захваченных казённых земель, разграничение землевладений и землепользований, установление их границ, ликвидация запутанности земельных отношений, уничтожение общности и чересполосности владений.

3. Недостаток земельных кадров и государственная важность послужили главной причиной образования Константиновской земельной школы.

4. В 1819 г. землемерная школа была переименована в Константиновское земельное училище, а в 1835 г. преобразована в Константиновский межевой институт. С этого времени она считается высшим учебным заведением.

5. Первым директором Константиновского межевого института и автором первого его устава стал известный русский писатель Сергей Тимофеевич Аксаков. Преподавателем русского языка в институте в то время работал великий русский литературный критик и философ Виссарион Григорьевич Белинский, преподавателем истории, чуть позже, - известный историк Иван Егорович Забелин.

6. С момента образования институт стал основным и единственным высшим учебным заведением, методическим и учебным центром по землеустройству в России. Сначала это было закрытое заведение с численностью обучающихся 200 человек и продолжительностью курса 4 года. С 1867 г. имел статус военного заведения.

7. В 1917 г. вуз был переименован в Московский межевой институт, в следующем году был разработан его новый устав. Землеустроительное образование получает политическую направленность, в январе 1919 г. в институте создаётся РКП(б), активно участвующая в его работе.

8. Постановлением СНК СССР от 2 февраля 1930 г. Межевой институт был передан в ведение Наркомзема СССР и реорганизован. В соответствии с решением Комиссии СНК СССР по реформе высшего и среднего образования на основе земельно-

переселенческого факультета был создан Московский институт землеустройства, геодезического факультета – Московский геодезический институт (впоследствии Московский государственный университет геодезии и картографии). В 1945 г. Московский институт землеустройства переименовывается в Московский институт инженеров землеустройства (МИИЗ).

9. В соответствии с постановлением Совета Министров РСФСР от 18 января 1991 г. на базе Московского института инженеров землеустройства был создан Государственный университет по землеустройству с подготовкой специалистов по земельному праву, землеустройству, почвоведению, геоботанике, геодезии, архитектуре и планировке сельских населённых мест.

10. В настоящее время университет является единственным в России специализированным и самым крупным высшим учебным заведением по подготовке инженеров в области землеустройства, земельного и городского кадастров. Здесь готовят также геодезистов, архитекторов, юристов, экономистов-менеджеров в области управления земельными ресурсами и земельного рынка, оценщиков земли и недвижимости.

Задание №6. Набрать текст

1. Содержащий 10 первых предложений приведенных ниже пар.
2. Скопировать этот текст и преобразовать каждое из скопированных первых предложений пар во вторые предложения этих пар.
3. Скопировать каждую пару предложений (не преобразованное и преобразованное) в отдельную ячейку буфера обмена.
4. Вставить из буфера обмена сначала все пары предложения с нечетными номерами, а затем пары с четными номерами.

Варианты пар предложений:

1. В школе Петр Иваныч слыл ябедником, за что был, презираем и не единожды бит. → Презираем и не единожды бит был Петр Иваныч в школе, так как слыл там ябедником.

2. Умницу Сидорова – зачислить, а дурака Петрова – исключить. → Умницу Петрова – зачислить, а дурака Сидорова – исключить

3. В огороде – бузина, а в Киеве – дядька. → В Киеве – бузина, а в огороде – дядька.

4. Куй железо, пока горячо. → Пока железо горячо – куй.

5. Мой дядя самых честных правил. → Правил самых честных мой дядя.

6. Кто не курит и не пьет, тот здоровеньким умрет. → Кто не пьет и не курит, тот умрет здоровеньким.

7. Полюбил на свое горе Петя английскую королеву. → Английскую королеву полюбил Петя на свое горе.

8. Карл у Клары украл кораллы, а Клара у Карла украла кларнет. → Украла Клара кларнет у Карла, а Карл кораллы у Клары украл.

9. Карл у Клары украл кораллы, а Клара у Карла украла кларнет →

Украл у Клары Карл кораллы, а Клара у Карла кларнет украла.

10. Когда едешь на Кавказ, солнце светит в правый глаз. →

В правый глаз солнце светит, когда на Кавказ едешь.

§4. Создание таблиц

Таблицы являются удобным средством хранения и анализа информации. Word предлагает разнообразный и эффективный инструментарий для создания таблиц. Мы познакомимся с технологией создания таблиц на достаточно простом примере.

Попробуем создать лист календаря на один месяц, представив дни недели и числа в табличном виде.

Создадим календарь на июль 2007 года.

1. Перейдите на вкладку **Вставка**.

2. Нажмите кнопку **Таблица**, расположенную в группе таблицы. Под кнопкой появится меню (рис. 1.18).

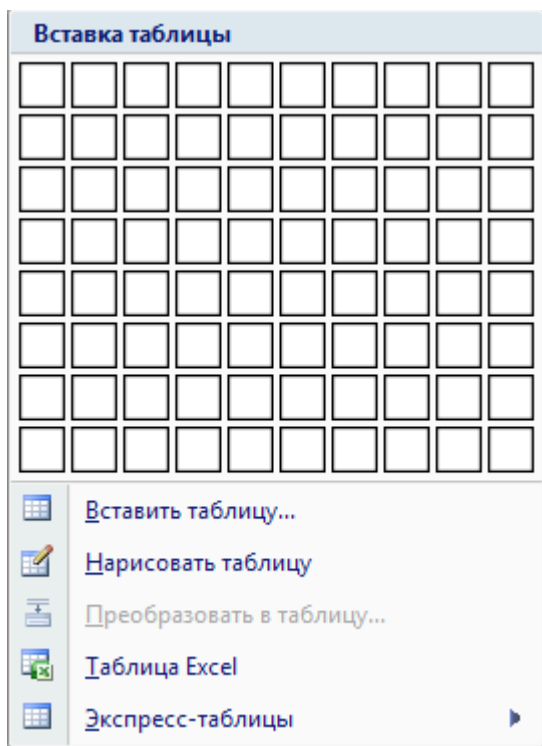


Рис. 1.18 . Меню кнопки **Таблица**

Обратите внимание на множество квадратов, расположенных над командами меню. Вы можете быстро выбрать количество строк и столбцов таблицы, выбрав количество квадратов по вертикали и горизонтали.

Таблица будет содержать семь столбцов (по количеству дней недели).

1. Установите указатель мыши на левый верхний квадрат. В документе появится одна ячейка таблицы, растянутая от левого до правого поля документа.
2. Переместите указатель мыши на седьмой квадрат слева в верхней строке. Ранее появившаяся ячейка будет разделена на семь частей. Таким образом вы обозначили количество столбцов ячейки.
3. Переместите указатель мыши вниз до пятого сверху квадрата (старайтесь не выводить указатель мыши из седьмого столбца). В документе появится таблица, состоящая из семи столбцов и пяти строк. При этом в верхней части панели команд должна появиться надпись **Таблица 7x5**.

Если вы ещё не нажали кнопку мыши, то можете перемещать указатель по квадратам, пока не наберёте нужное количество строк и столбцов таблицы.

- Щёлкните кнопкой мыши на седьмом слева квадрате, расположенном в пятом ряду сверху. Панель команд будет закрыта. Теперь ваш документ содержит таблицу, состоящую из пяти строк и семи столбцов, (рис. 1.19).

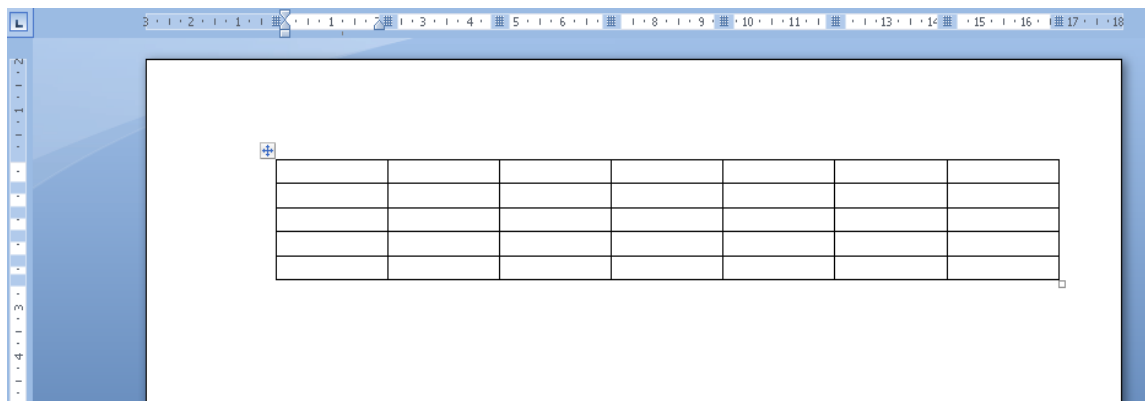


Рис. 1.19. Таблица добавлена в документ

Чуть позже при необходимости вы добавите недостающие ячейки, а сейчас введите в созданную таблицу данные.

- Щёлкните кнопкой мыши на левой верхней ячейке таблицы. В ячейку будет установлен курсор.
- Введите слово «Понедельник».
- Щёлкните кнопкой мыши на второй слева ячейке в верхнем ряду. Курсор будет установлен в неё.
- Введите слово «Вторник».
- Вышеописанным способом введите в остальные верхние ячейки таблицы названия оставшихся дней недели по порядку.

Вы можете быстро перемещать курсор по ячейкам таблицы, используя клавиши для управления курсором.

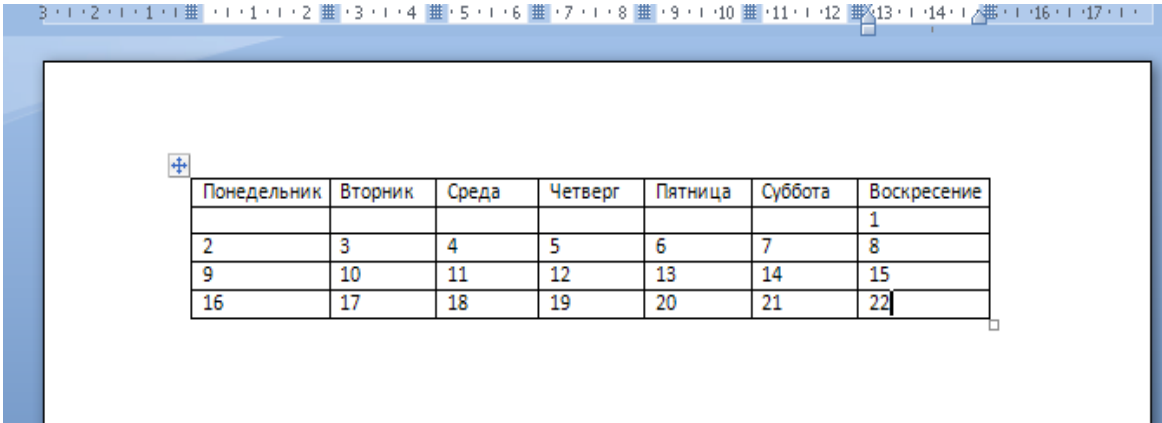
Итак, верхняя строка таблицы будет содержать названия дней недели. В остальных ячейках таблицы будут указаны числа месяца в соответствии с днями недели.

1 июля 2007 года приходится на воскресенье. Поэтому цифру 1 нам нужно разместить в правой ячейке второй строки таблицы.

- Установите курсор в правую ячейку второй строки таблицы.
- Видите цифру 1.
- Установите курсор в левую ячейку третьей строки. Вы можете быстро сделать это, дважды нажав клавишу «>>».

При первом нажатии курсор установится за пределами таблицы (справа от ячейки с цифрой 1), а при втором – переместится в первую ячейку следующей (в данном случае – третьей) строки.

4. Введите цифру 2.
5. Вышеописанными способами введите остальные числа месяца в соответствии с днями недели (рис. 1.20).



	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота	Воскресение
							1
2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17
18	19	20	21	22			

Рис. 1.20. Дни месяца введены в ячейки таблицы.

Вы должны были остановиться на цифре 22, так как для ввода остальных дней месяца вам не хватило ячеек таблицы. Таблица с недостаточным количеством строк создана специально, чтобы показать, как её дополнить.

1. Убедитесь, что курсор установлен в правой нижней ячейке. Курсор может быть установлен в любой ячейке нижней строки. В данном случае на результат это не повлияет
2. Щёлкните правой кнопкой мыши. На экране появится контекстное меню.
3. Выберите команду контекстного меню **Вставить**. Обратите внимание, что в контекстном меню находятся две такие команды. Первая (верхняя) относится к работе с буфером обмена. Нам нужна вторая (нижняя) команда **Вставить**. На экране отобразится вложенное подменю. (рис. 1.21).

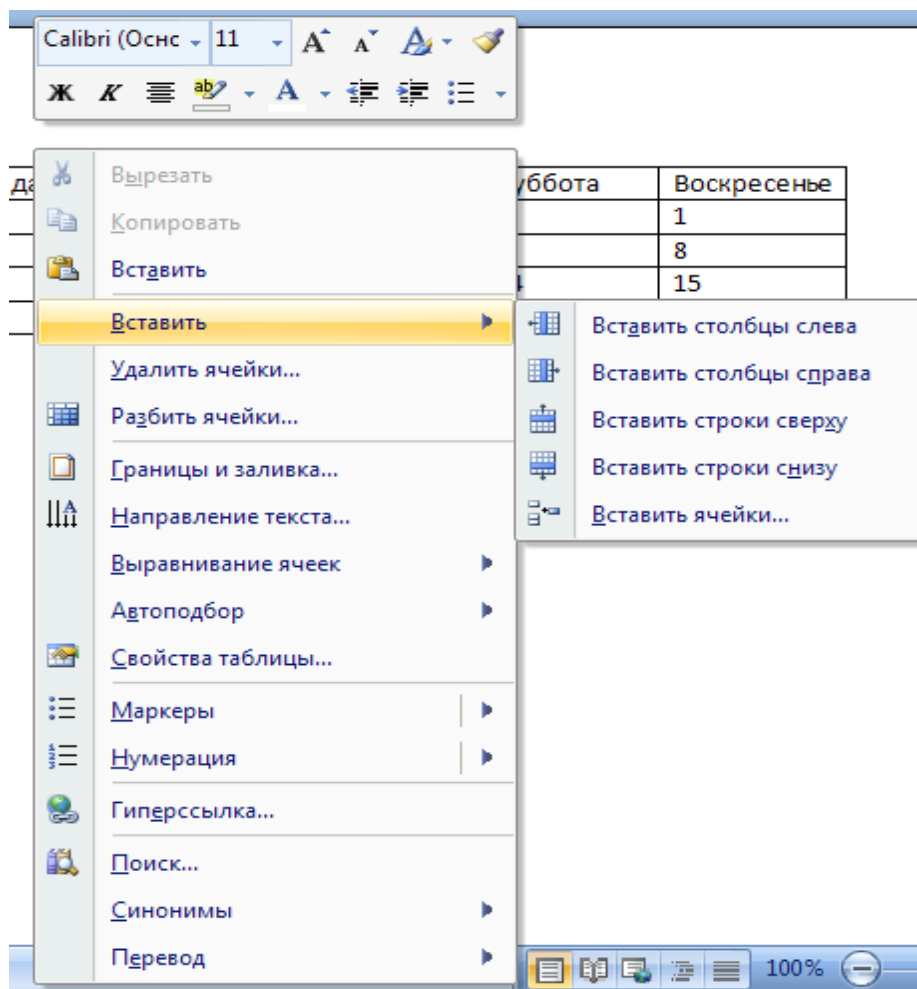


Рис. 1.21. Подменю команды **Вставить**

4. Выберите команду **Вставить строки снизу**. В нижней части таблицы появится новая (шестая) строка.
5. Установите курсор в левую ячейку добавленной (шестой) строки и введите следующий день месяца (23).
6. Введите числа месяца в оставшиеся ячейки. Вы остановитесь на числе 29, так как вам опять не хватит ячеек таблицы.
7. Если вы находитесь в последней ячейке нижней строки, то быстро добавить новую строку можно следующим образом.
8. Нажмите клавишу **Tab**. Новая строка будет автоматически создана, а курсор перейдет в первую ячейку новой строки.
9. Введите остальные два числа месяца (30 и 31) в первые две ячейки нижней строки.

Вы создали таблицу, которая представляет собой календарь на июль 2007 года, (рис. 1.22).

Понедельник	Вторни	Среда	Четверг	Пятница	Суббота	Воскресенье
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Рис. 1.22. Календарь на июль 2007 года

Задание №7. Создайте нижеприведенные таблицы с номерами по указанию преподавателя.

Толщина и вид линий обрамления таблиц и их элементы должны совпадать с образцом. Размер шрифта выбирайте таким образом, чтобы надписи и данные помещались в предназначенных для них ячейках. Имейте в виду, что если установить курсор в любую клетку таблицы, а затем установить автоматический выбор высоты строки (Таблица - Высота и ширина ячейки - Вкладка строка - Высота ячейки "Авто"), то по мере ввода данных в клетку будет автоматически увеличиваться высота всей строки.

Таблица 1

Счет за пользование интернетом				
Дни недели	Тарифы		Расход за месяц	
	Время суток	Стоимость 1 минуты	Время в минутах	Стоимость в рублях
Рабочие дни (с 8 ⁰⁰ понедельника до 20 ⁰⁰ пятницы)	8 ⁰⁰ -20 ⁰⁰			
	20 ⁰⁰ -8 ⁰⁰			
Выходные дни (с 20 пятницы до 8 ⁰⁰ понедельника)	10 ⁰⁰ -22 ⁰⁰			
	22 ⁰⁰ -2 ⁰⁰			
	2 ⁰⁰ -6 ⁰⁰			
	6 ⁰⁰ -10 ⁰⁰			

Таблица 2. (Эту таблицу создавайте при горизонтальной ориентации листа).
Сведения о полных семьях (отец, мать, один или два ребенка).

		Родители				Дети				
№	Отец		Мать		1-й ребенок			2-й ребенок		
	Фамилия Имя Отчество	Дата рождения	Фамилия Имя Отчество	Дата рождения	Пол	Имя	Дата рождения	Пол	Имя	Дата рождения
1	Соколов Михаил Алек- сандрович	8 марта 1965 г.	Соколова Елена Владимировна	23 февраля 1968 г.	м	Вик- тор	14 ок- тября 1989 г.	ж	Ольга	16 марта 1992 г.
2										
3										

Таблица 3

Распределение площадей между основными культурами							
Хозяйство	Зерновые культуры				Овощные культуры		
	Рожь	Пшеница	Гречиха	Кукуруза	Картофель	Свекла	Капуста
Проскурин П.Ф.							
АО "Трудолюб"							

Таблица 4

Соотношение между площадями под зерновыми культурами и валовыми сборами.								
Хозяйство	К у л ь т у р ы							
	Рожь		Пшеница		Гречиха		Кукуруза	
	Пло- щадь	Вал. сбор	Площадь	Вал. сбор	Площадь	Вал. сбор	Площадь	Вал. сбор
Проскурин П.Ф.								
АО "Трудолюб"								

Таблица 5

Соотношение между площадями под овощными культурами и валовыми сборами						
Хозяйство	К у л ь т у р ы					
	Картофель		Свекла		Капуста	
	Пло- щадь	Вал. Сбор	Пло- щадь	Вал. сбор	Пло- щадь	Вал. сбор
Проскурин П.Ф.						
АО "Трудолюб"						

Таблица 6

Счет за пользование интернетом				
Дни недели	Тарифы		Расход за месяц	
	Время суток	Стоимость 1 минуты	Время в минутах	Стоимость в рублях
В любой день недели	20 ⁰⁰ -8 ⁰⁰	500 р. в месяц		500
В любой день недели, кроме субботы и воскресенья.	8 ⁰⁰ -12 ⁰⁰	40 коп.		
	12 ⁰⁰ -20 ⁰⁰	60 коп.		
Суббота и воскресенье	8 ⁰⁰ -12 ⁰⁰	20 коп.		
	12 ⁰⁰ -18 ⁰⁰	40 коп.		
	18 ⁰⁰ -20 ⁰⁰	60 коп.		
Общая плата за месяц				

Таблица 7

Распределение по возрасту и полу лиц до 20 лет									
Населенный пункт	Всего до 20	до 3-х лет		от 3 до 8 лет		от 8 до 15 лет		от 15 до 20 лет	
		муж.	жен.	муж.	жен.	муж.	жен.	муж.	жен.

Таблица 8

Распределение по возрасту и полу							
Населенный пункт	Общее кол-во населения	до 30 лет		от 30 до 60 лет		Старше 60 лет	
		муж.	жен.	муж.	жен.	муж.	жен.

Таблица 9

Счет за услуги телефонной сети									
Наименование телефонного узла:									
Номер телефона:									
Фамилия И.О. абонента:									
Адрес: улица									
Дом		Корпус			Квартира				
Оплачиваемый месяц (месяцы), год:									
Абонен. плата		Парал. аппарат		Факс		Прочие услуги		Всего	

Таблица 10

Счет за газ							
Фамилия И.О.:							
Адрес: улица		Дом		Корпус		Квартира	
Оплачиваемый месяц(ы), год							
Количество жильцов							
Тариф (за жильца в месяц)							
Количество месяцев							
Всего по счету							
Пени							
Сумма							
Подпись ответственного квартиросъемщика							

§5. Редактор формул

Текст документа (учебного пособия, научной статьи и т.д.) может содержать математические формулы.

В качестве примера вставим в текст, формулы для решения квадратного уравнения.

1. Установите курсор в позицию, в которую хотите вставить формулу.
2. Убедитесь, что на ленте открыта вкладка **Вставка**.
3. Нажмите кнопку **Формула** в группе **Символы**. В документе появится поле **Место для формулы**, а на ленте – новая контекстная вкладка **Конструктор**.

4. Нажмите на кнопку **Формула** в группе **Сервис** вкладки **Конструктор**. На экране появится меню с галереей, содержащей несколько известных формул (рис. 1.23)
5. В галерее щёлкните кнопкой мыши на формуле **Квадратное уравнение**. В поле документа появится изображение выбранной формулы.

Вы можете подставить в формулу свои символы вместо предлагаемых. Для этого следует выделить символ (букву, цифру, знак) в формуле и впечатать вместо него нужный. Иными словами, редактирование формулы производится теми же методами, что и редактирование текста.

Формула, содержащая дробь, может быть представлена в двух форматах: линейном и профессиональном.

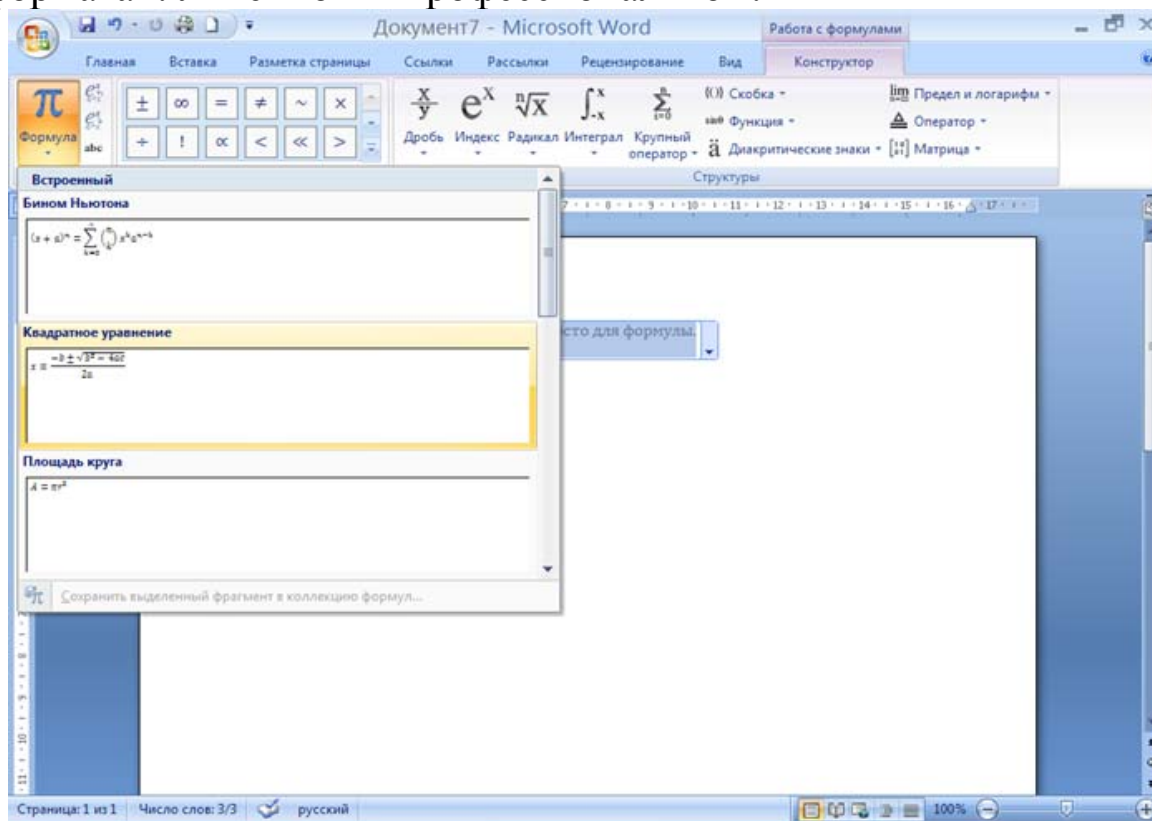


Рис. 1.23. Галерея формул

1. Щёлкните кнопкой мыши на треугольничке в правой части поля формулы. На экране появится контекстное меню.

2. В появившемся контекстном меню выберите команду **Линейный**. Представление формулы изменится. Теперь все символы формулы располагаются в одну строку.
3. Снова вызовите контекстное меню формулы и выберите команду **Профессиональный**. Представление формулы будет изменено на профессиональное (с вертикальным расположением дроби) (рис. 1.24.)

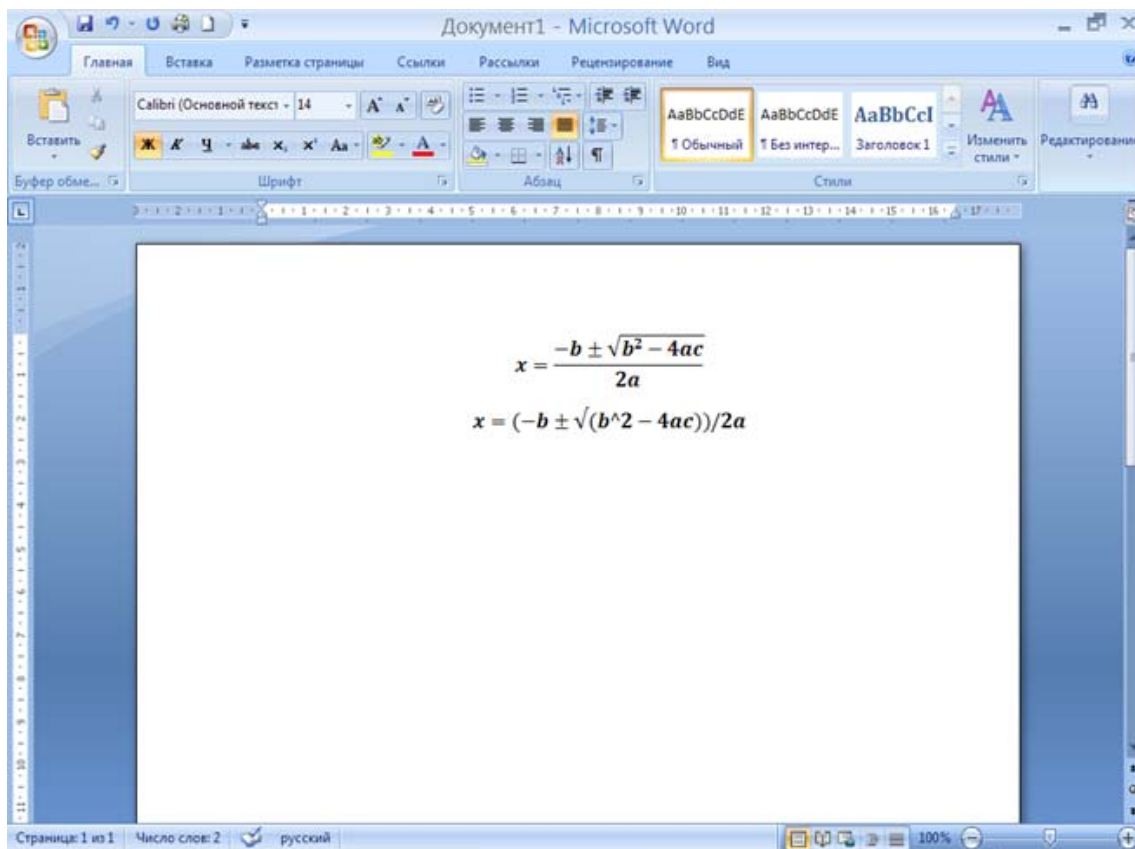


Рис. 1.24. Профессиональное и линейное представление одной и той же формулы

Если в галерее формул вы не нашли нужной, то можете создать собственную формулу.

Создадим простую формулу, где x будет равняться отношению синуса y к косинусу y .

1. Установите курсор в позицию, в которую хотите вставить формулу.

2. Нажмите на кнопку **Формула** в группе **Символы** вкладки **Вставка**. В документе появится поле **Место для формулы**.
3. В поле формулы введите символ x .
4. На контекстной вкладке **Конструктор** в группе **Символы** щёлкните кнопкой мыши на знаке « \Rightarrow » (но можно просто набрать этот символ на клавиатуре). После символа x в формуле появится знак равенства. Теперь нам необходимо создать дробь.
5. Выполните **Конструктор**, **Структуры**, **Дробь** и в появившейся галерее выберите шаблон **Простая дробь** с вертикальным расположением числителя и знаменателя. После знака равенства в формуле появится горизонтальная линия дроби, сверху и снизу которой расположены пунктирные прямоугольники. Нам необходимо заменить эти прямоугольники выражениями $\sin u$ и $\cos u$.
6. Выделите верхний прямоугольник, щёлкнув на нём кнопкой мыши.
7. В группе **Структуры** нажмите кнопку **Функция** и в появившейся галерее выберите шаблон \sin . Верхний квадрат заменится выражением \sin , справа от которого, в свою очередь, тоже появится квадрат (рис. 1.25.)
8. Выделите квадрат справа от выражения \sin и нажмите клавишу Y . Теперь в верхней части дроби имеется нужное выражение – $\sin u$.
9. Вышеописанным способом замените квадрат в знаменателе дроби выражением $\cos u$. Выражение \cos вы можете найти в галерее **Функции** группы **Структуры**.

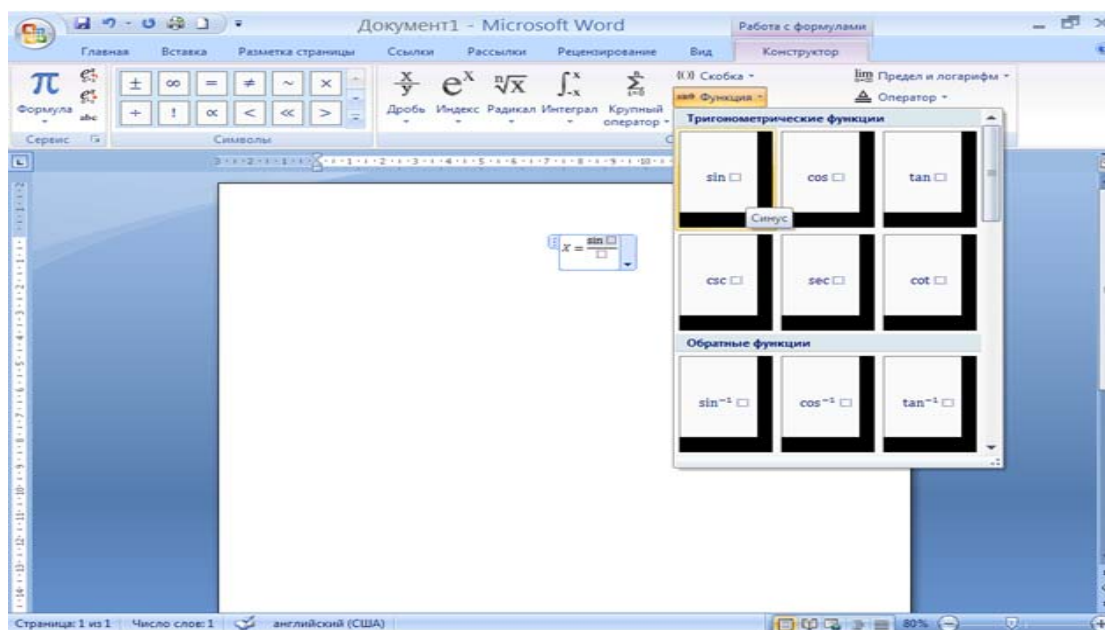


Рис. 1.25. В числитель дроби добавлена функция

Вы создали собственную формулу и можете сохранить её в галерее формул, чтобы в дальнейшем при необходимости быстро вставить в документ.

1. Щёлкните кнопкой мыши на треугольничке в правой части поля формулы, чтобы открыть контекстное меню.
2. В появившемся контекстном меню выберите команду **Сохранить как новую формулу**. На экране появится окно **Создание нового стандартного блока** (рис. .26.)

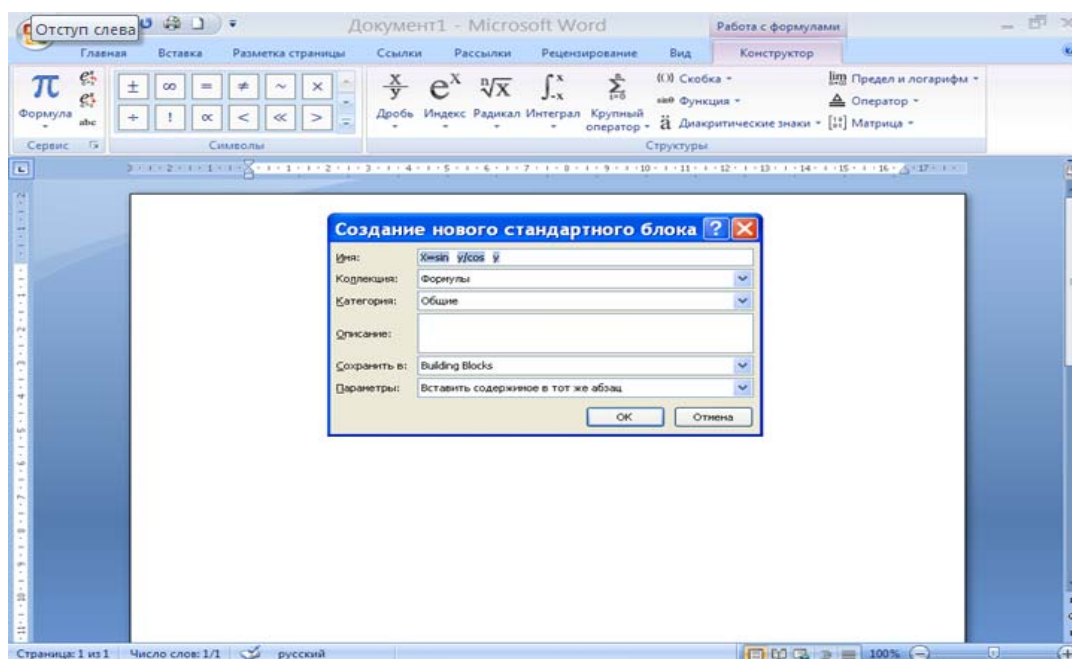


Рис. 1.26. Окно Создание нового стандартного блока

3. В поле **Имя** введите название сохраняемой формулы. Это может быть любое название, понятное вам, например **Моя первая формула**.
4. Нажмите кнопку **ОК**. Окно будет закрыто.
5. Нажмите кнопку **Формула** в группе **Сервис** контекстной вкладки **Конструктор**, перейдите в конец формулы и убедитесь, что созданная вами формула помещена в галерею.

Поэкспериментируйте с остальными компонентами галереи вкладки **Конструктор**, и вы убедитесь, что Microsoft Word содержит огромное количество символов и заготовок для создания любой – даже самой сложной – формулы.

Задание № 8. Используя мастер формул, наберите следующие математические формулы:

1.
$$y = \frac{1+x^2}{1+2x^2}$$

2.
$$\begin{aligned} 4x^2+5y^2 &= 3 \\ 5x+3y &= 1 \end{aligned}$$

3.
$$S = \frac{\sum_{i=1}^n x_i + 2i \sum_{i=1}^n y_i^2 + 5 \left(\sum_{i=1}^m \sum_{j=1}^m b_{ij} \right)^2}{3 + \sum_{i=1}^n y_i}$$

4.
$$A = \begin{pmatrix} 2 & 3 & 5 & 2 \\ 5 & 2 & 7 & 5 \\ 4 & 2 & 1 & 7 \\ 7 & 5 & 4 & 1 \end{pmatrix}, B = \begin{pmatrix} 3 \\ 2 \\ 3 \\ 2 \end{pmatrix}, Y = \begin{pmatrix} 3 \\ 1 \\ 2 \\ 2 \end{pmatrix}$$

5.

$$s = \sum_{i=1}^n x_i + 2i \sum_{i=1}^n y_i^2 + \left(\sum_{i=1}^m \sum_{j=1}^m b_{ij} \right) \left(2 + \sum_{i=1}^n x_i \right) - 2 \left(1 + \sum_{i=1}^n x_i y_i \right)$$

$$x = (1, 2, 7, 4), y = (1, 7, 2, 3), b = \begin{pmatrix} 4 & 1 \\ 2 & 5 \end{pmatrix}$$

$$6. \quad Y = \frac{1+x e^{-x}}{2+\sqrt{x^2+\sin^2(x)}}$$

$$7. \quad G = \begin{cases} \frac{1+x^2}{\sqrt{1+x^2}}, & x \leq 0 \\ 2x + \frac{\sin^2(x)}{2+x}, & x > 0 \end{cases}$$

$$8. \quad \begin{cases} \frac{1+|x|}{\sqrt[3]{1+x+x^2}}, & x \leq -1 \\ 2\ln(1+x^2) + \frac{1+\cos^4(x)}{2+x}, & x \in [-1; 0] \\ (1+x)^{\frac{x}{2}}, & x \geq 0 \end{cases}$$

$$9. \quad F = \begin{cases} \sqrt{1-\frac{2x}{1+x^2}}, & x \in [0; 1] \\ \frac{1+x+x^3}{1+x^2}, & x < 0 \\ 2|0.5 + \sin(x)|, & x \geq 1 \end{cases}$$

$$10. \quad y = 2 \sin(2\pi x) \cos(\pi x) + \sin(3\pi x);$$

$$z = \cos(2\pi x) \sin^2(\pi x) - \cos(4\pi x)$$

$$11. \quad s = \left(\sum_{i=1}^m a_i \right) + \sum_{i=1}^n \sum_{j=1}^n c_{ij} - \left(3 + \sum_{i=1}^n \sum_{j=1}^n c_{ij} \right) \left(1 + \sum_{i=1}^m a_i^2 \right),$$

$$A = (3, 3, 1, 3), c = \begin{pmatrix} 2 & 2 & 4 \\ 2 & 4 & 6 \\ 2 & 5 & 3 \end{pmatrix}.$$

§6. Инструмент Smart Art и создание фигур

Это новый инструмент в программе Microsoft Word версии 2007, он является удобным средством для построения красочных и наглядных схем, графиков, иерархических списков или диаграмм. Всего несколькими щелчками кнопкой мыши вы можете иллюстрировать структуру вашей организации, иерархию отделов, руководителей либо просто красочную титульную страницу главы или раздела с вводным текстом.

Построим простую схему иерархии управления университетом с помощью инструмента Smart Art.

1. Создайте новый документ.
2. Перейдите на вкладку **Вставка**.
3. Нажмите кнопку **Smart Art**, расположенную в группе **Иллюстрации**. На экране отобразится окно **Выбор рисунка Smart Art** с образцами готовых шаблонов, упорядоченных по категориям (рис. 1.27).
4. Выберите категорию **Иерархия**, щёлкнув кнопкой мыши на одноимённом пункте списка, расположенном в левой части окна. Набор образцов в окне будет обновлён.
5. Щёлкните кнопкой мыши на образце **Горизонтальная иерархия** (названия образцов появляются во всплывающей подсказке при наведении указателя мыши) и нажмите кнопку **ОК**. Окно закроется, а в документе появится выбранная вами схема Smart Art. При этом на панели инструментов появится контекстный инструмент **Работа с рисунками Smart Art** и его контекстные вкладки **Конструктор** и **Формат**.

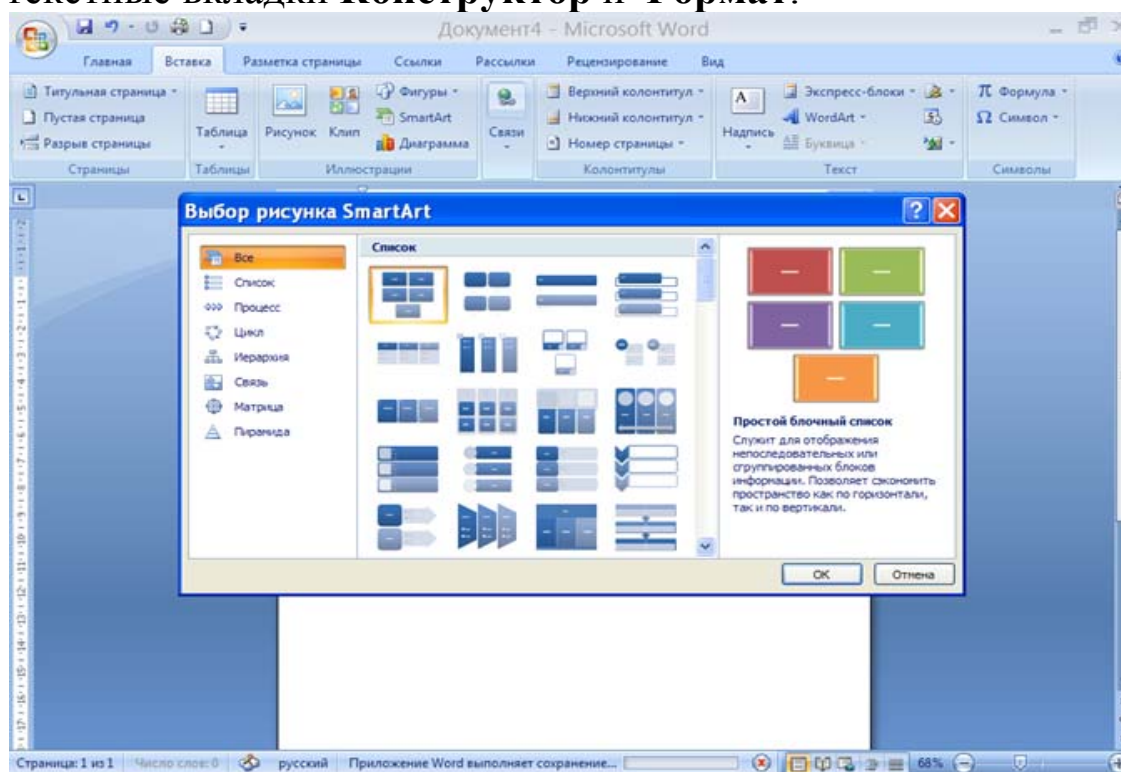


Рис. 1.27. Окно **Выбор рисунка Smart Art**

В каждом блоке схемы содержится надпись **Текст**. Эти надписи вам предстоит заменить.

1. Щёлкните кнопкой мыши на первом (левом) блоке. В нём появится курсор.
2. Введите слово «Ректор».

Обратите внимание, что размер символов текста в блоке изменяется в зависимости от количества текста. Таким образом, при большом количестве текста блок остаётся неизменным, а размер символов уменьшается, чтобы весь текст поместился.

3. В блоки, расположенные справа, введите название должностей в порядке подчинения.

Теперь ваш документ содержит наглядную схему управления в университете (рис. 1.28).

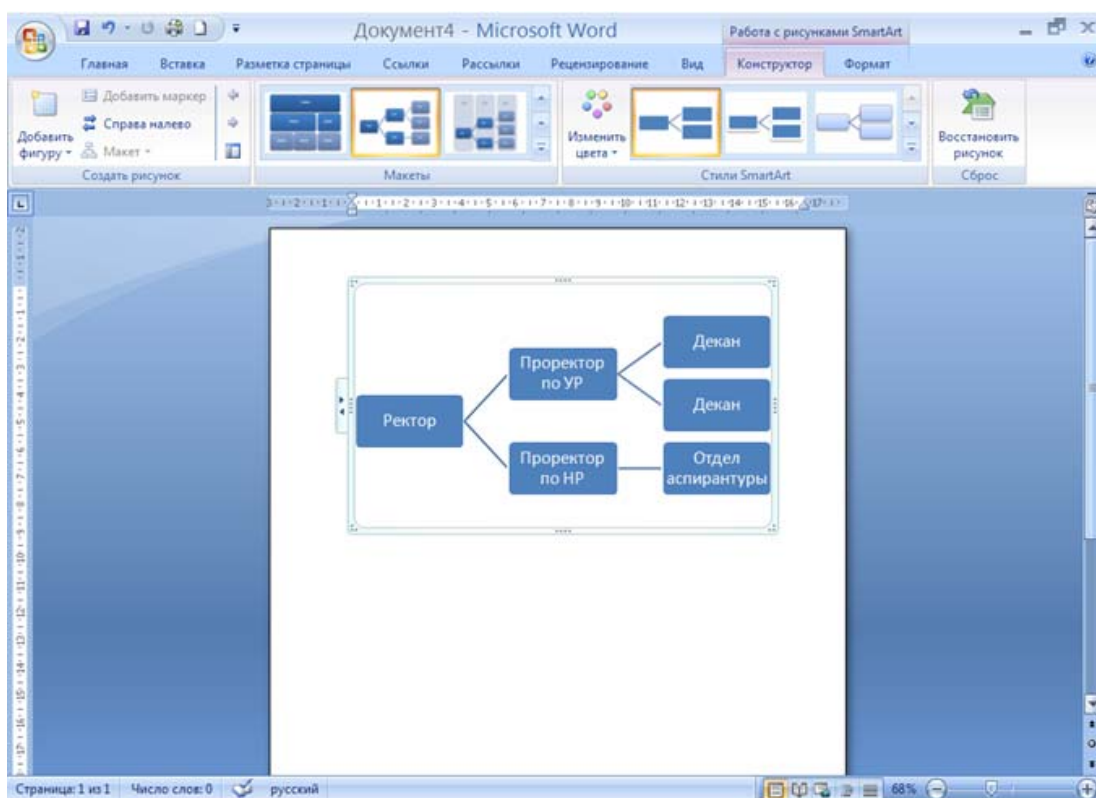


Рис. 1.28. Надписи в блоках изменены

Программа Microsoft Word содержит огромное количество заготовок различных геометрических фигур, которые вы можете добавить в свой документ. Каждая фигура обладает свойствами, доступными для изменения. Вы сможете создать собственную фигуру, если не найдете нужной среди шаблонов.

Создадим несколько геометрических фигур в документе.

1. Создайте новый документ.
2. Перейдите на вкладку **Вставка** ленты.
3. Нажмите кнопку **Фигуры**, расположенную в группе **Иллюстрации** вкладки **Вставка**. На экране появится галерея шаблонов геометрических фигур (рис. 1.29).

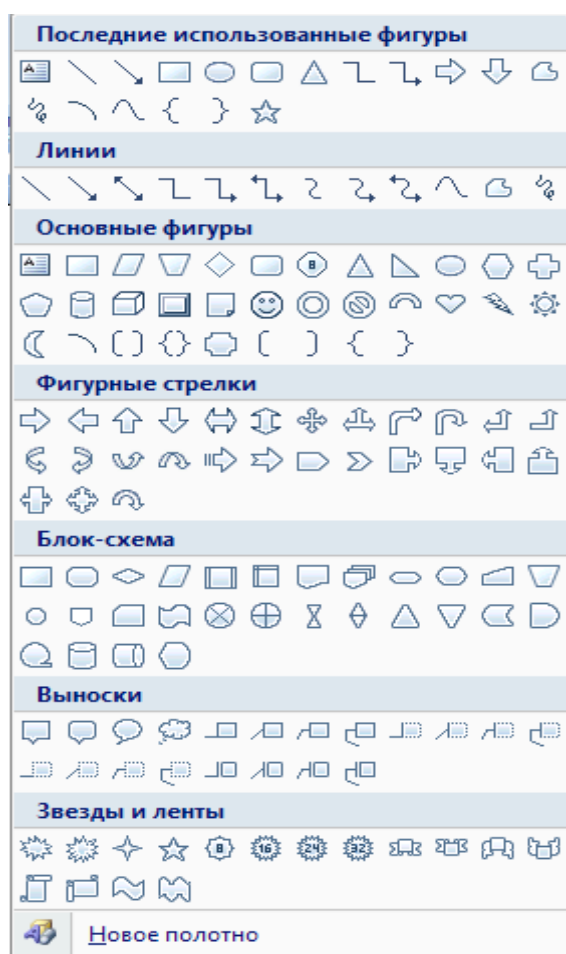


Рис. 1.29. Шаблоны геометрических фигур

Добавьте в документ трапецию.

1. Щелкните кнопкой мыши на значке Трапеция, расположенном в группе основные фигуры галереи. Указатель мыши примет вид крестика.

2. Установите указатель мыши на область документа и, нажав и удерживая левую кнопку мыши, переместите указатель по диагонали. Затем отпустите кнопку мыши. В документе появится изображение трапеции.

После добавления фигуры на панели инструментов появится контекстный инструмент Средства рисования и его контекстная вкладка Формат.

Добавьте в документ другую фигуру.

1. Нажмите кнопку **Фигуры** и в появившейся галерее выберите фигуру **Треугольник**, расположенную в группе **Основные фигуры**.
2. Установите указатель мыши на область документа и, нажав и удерживая левую кнопку мыши и кнопку «Ctrl», переместите указатель по диагонали. Отпустите кнопку мыши. Теперь в документе появилась еще одна фигура - треугольник (рис. 1.30).

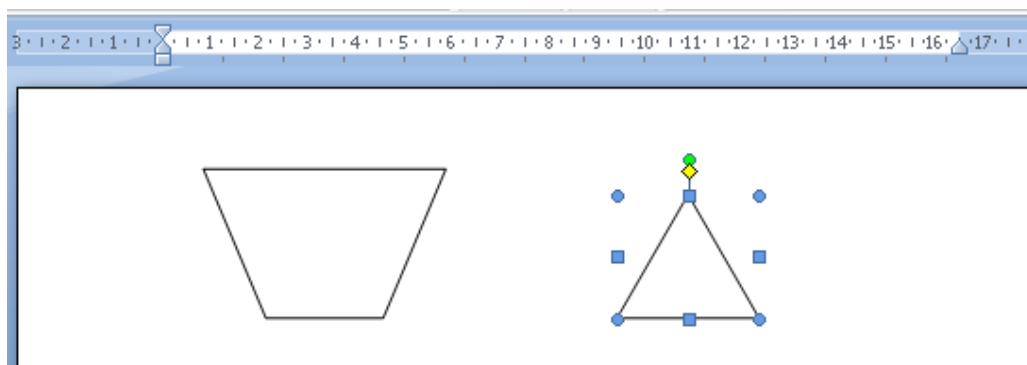


Рис. 1.30. Фигуры Трапеция и Треугольник добавлены в документ.

Таким образом, вы можете добавить в документ любую фигуру, содержащуюся в галерее **Фигуры**.

Задание №9. Используя инструмент Smart Art и инструмент создания Фигур создайте схемы документов, образцы которых, необходимо получить у преподавателя.

§7. Режимы отображения документа

Задание №10. Скопируйте все выполненные ранее задания по Word в один общий файл.

Теперь у вас есть загруженный в Microsoft Word файл, и вы можете поэкспериментировать с режимами отображения документа в окне программы.

Наиболее удобным для планирования и компоновки содержимого документа является режим, называемый *разметкой страницы*. В этом режиме вы видите документ в виде одной или нескольких страниц, относительные размеры которых соответствуют бумажным страницам при печати.

Режим разметки страницы устанавливается по умолчанию для всех документов, создаваемых с использованием основного шаблона.

1. Перейдите на закладку **Вид**. На ленте отобразятся элементы управления отображением документа.
2. Убедитесь, что кнопка **Разметка страницы** нажата. Если это не так, нажмите эту кнопку. Режим **Разметка страницы** будет включён.
3. Нажмите кнопку **Режим чтения**. Режим отображения документа будет изменён (рис. 1.31).

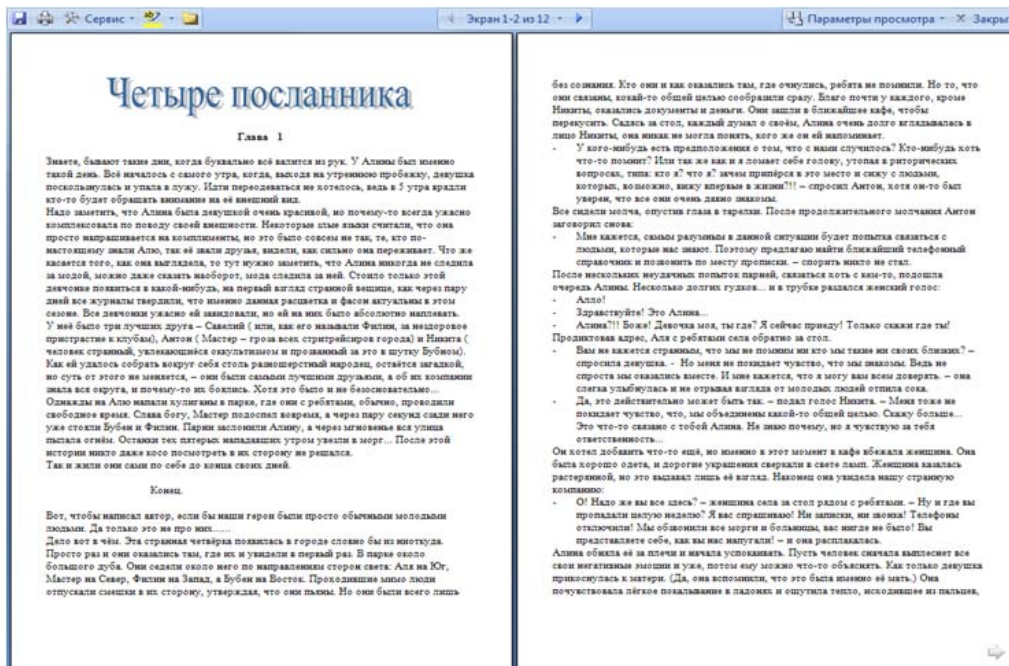


Рис. 1.31. Режим чтения

В режиме чтения на экране отображаются две страницы документа, подобно раскрытой книге. При этом лента, линейки и некоторые другие элементы интерфейса программы скрываются, чтобы максимально освободить экранное пространство. В нижней части каждой страницы расположены кнопки со стрелками для перехода к следующим страницам. Кнопки со стрелками не отображаются лишь на первой и последних страницах документа.

В режиме чтения вы можете только читать документ, вносить примечания или выделять цветом фрагменты текста. Править или вводить новый текст в этом режиме нельзя.

1. Нажмите кнопку **Закреть** в правом верхнем углу окна программы в режиме чтения, чтобы вернуть обратно в режим разметки.

2. Теперь нажмите кнопку **Структура**. Вид отображения документа изменится. (рис. 1.32).

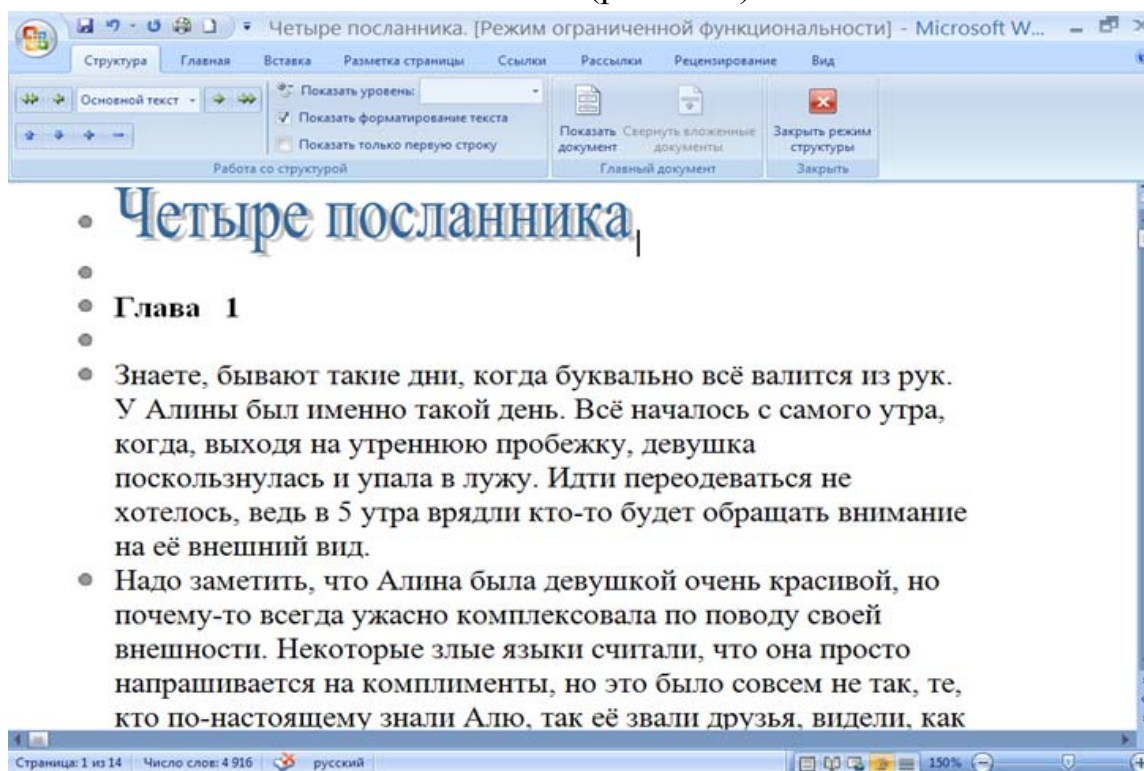


Рис. 1.32. Структура документа

В этом режиме документ отображается в структурном виде. Если текст содержит заголовки глав и разделов, то можно скрыть или отобразить текст, расположенный между заголовками.

Слева от каждого заголовка, названия главы или раздела располагается плюсики, дважды щёлкнув кнопкой мыши на котором,

вы можете скрыть или, наоборот, отобразить на экране текст, принадлежащий выбранному разделу или главе, или вложенные заголовки.

Режим **Структура** удобен для просмотра структуры создаваемого документа, позволяет обеспечить правильную организацию уровней заголовков. В этом режиме вы можете править или вводить новый текст.

С помощью элементов управления, расположенных в верхней части окна режима **Структура**, вы можете изменить уровень заголовка и отобразить или скрыть определённые уровни.

1. Нажмите кнопку **Заккрыть режим структуры**, чтобы вернуться в режим разметки страницы.
2. Теперь нажмите кнопку **Черновик**. Представление документа на экране будет изменено.

Режим **Черновик** предназначен для быстрого редактирования текста. Вид документа в этом режиме несколько упрощён. Некоторые элементы документа, например колонтитулы, в режиме **Черновик** не отображаются.

Нажмите кнопку **Разметка страницы**, чтобы перейти в одноимённый режим отображения документа.

Мы рассмотрели основные режимы отображения документа в окне программы Microsoft Word. На разных этапах вам, возможно, придётся переключать эти режимамы для удобства работы.

В данной главе вы познакомились с различными сервисными средствами программы Microsoft Word. Мы постарались охватить наиболее часто используемые функции и не вдавались в тонкости работы с ними. Здесь показаны лишь основные приёмы работы с этими средствами. Если вы захотите изучить инструменты более тщательно, то можете обратиться к справочной системе программы Microsoft Word.

§8. Верстка многостраничных документов

Создание приглашения.

В процессе работы над этим заданием Вы должны набрать, сохранить и отпечатать приглашение на свадьбу.

Приглашение необходимо выполнить в виде книжки, состоящей из четырёх страниц. Создание на компьютере заготовки для будущей книги называют версткой. Одной из главных целей верстки является такое размещение страниц на листе, при котором при сборке этих листов в книгу страницы располагались бы в нужном порядке.

Основные этапы работы:

1. Установка размеров страницы и ширины полей, выбор типа ориентации страницы (альбомная, формат А5)
2. Установка режима создания документа в двух колонках.
3. Набор текста приглашения с использованием элементов верстки.
4. Сохранение документа на диске.
5. Печать приглашения на двух сторонах листа.

Необходимость создания документа, который нужно отпечатать с двух сторон листа по две колонки на каждой стороне, т.е. осуществить верстку, накладывает свои требования. Рассмотрим перечисленные выше этапы подробнее.

Установите ввод текста двумя колонками. Для этого пройдите в главное меню путь «Разметка страницы» → «Колонки» и в появившемся окне щелкните мышью по пиктограмме с изображением двух колонок. (рис. 1.33).

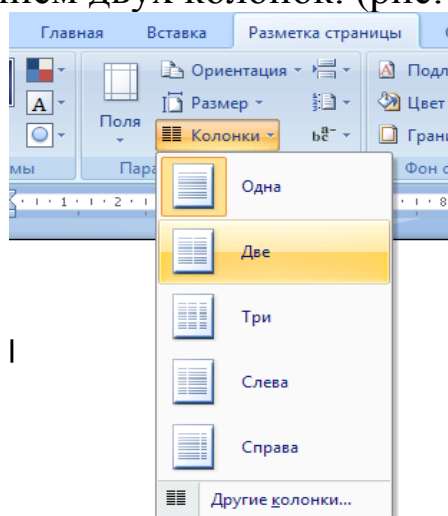


Рис. 1.33. Настройка колонок

Щелкните по кнопке **ОК**, и первая страница будущего макета появится на экране.

В левой колонке будет находиться курсор. Чтобы сделать доступной для курсора правую колонку первой страницы, необходимо заполнить левую колонку, и тогда курсор переместится в правую колонку. Теперь мы можем перемещать его из одной колонки в другую с помощью мыши или клавиш [←] [→].

Создадим вторую страницу макета приглашения. Не забывайте, что документ должен состоять из двух страниц, которые нужно будет отпечатать с двух сторон одного и того же листа. Для создания второй страницы необходимо пройти путь на ленте вкладка **Вставка** → **Пустая страница**. Появится вторая страница, уже разбитая на две колонки. После заполнения правой колонки первой страницы, курсор переместится в левую колонку второй страницы.

Теперь проверим, не создали ли мы случайно лишнюю (третью) страницу. Если документ состоит из двух страниц, то в левой части информационной строки на третьей позиции будет находиться информация 1/2 (первая страница двухстраничного документа) или 2/2 (вторая страница двухстраничного документа). Если страниц больше, то информация будет иметь вид 1/3, 2/3, 2/4 и т.п. Лишние страницы следует удалить, выделяя имеющиеся на них пустые строки и удаляя их клавишей **Backspace**.

Чтобы сохранить все сделанные установки, осуществите начальное сохранение документа в своей личной папке на жестком диске вашего компьютера. Итак, в нашем распоряжении две страницы, каждая из которых содержит по две колонки. Выполняйте промежуточные сохранения документа после заполнения каждой колонки.

В левую колонку первой страницы вставьте рисунок. Ниже приведена технология вставки рисунка в среде Word 2007, также можно вставить рисунок из среды Интернет. В более ранних версиях технология может быть другой. Но первый шаг “Вставка” → ”Рисунок” будет тот же.

Поместите указатель мыши в центр левой колонки первой страницы и выполните однократный щелчок левой клавишей. В 1-й строке колонки должен появиться курсор. Если курсор поя-

вился ниже, значит, Вы случайно ввели в колонку несколько пустых строк. Выделите эти строки и удалите их клавишей «Delete».

С помощью клавиши Enter продвиньте курсор на 3 строки вниз по левой колонке, затем с помощью клавиши ↑ возвратите его на 2-ю строку. Перечисленные манипуляции необходимы для того, чтобы после вставки рисунка сверху и снизу от него были пустые строки, что в дальнейшем позволит перемещать рисунок по вертикали. Это связано с форматом предлагаемых вам рисунков.


Пройдите в главном меню путь **Вставка** → **Рисунок** → **Картинки**. Появляется окно «Вставка картинки», в котором должна быть открыта вкладка «Рисунки» с изображениями тем. Выполните щелчок на рисунке с подходящей для вас темой. Появится окно с клипами. Выделите подходящий клип. Рядом с клипом появится контекстное меню. Щелкните в нем по первой кнопке **Вставить клип**, после чего закройте окно клипов (кнопка  в верхнем правом углу). В результате клип будет помещен в ваш документ, начиная со строки, на которой находился курсор перед обращением в меню **Вставка**.

Рисунок можно увеличить или уменьшить пропорционально. Для этого его необходимо выделить (поместить указатель мыши на поле рисунка и щелкнуть левой клавишей). Вокруг рисунка образуется рамка с кнопками. Поместите указатель мыши на кнопку в любом углу этой рамки и медленно перемещайте его там, пока не найдете такое положение указателя, в котором он приобретет вид наклонной обоюдоострой стрелки. В этот момент нажмите левую клавишу мыши и начните движение указателя под углом внутрь рисунка, чтобы уменьшить его, или наружу, чтобы его увеличить.

Рисунок можно исказить, «ухватив» его за кнопку в центре одной из сторон рамки, и вытянуть по вертикали или по горизонтали

Для перемещения рисунка по горизонтали нужно его выделить и воспользоваться кнопками выравнивания текста (по левому краю, по центру, по ширине), а также кнопкой **Отступ первой строки** на горизонтальной линейке.

Другие форматы рисунков позволяют манипулировать с ними проще, но эти форматы требуют значительно больше памяти, с чем при создании текстовых документов иногда приходится считаться.

В правую колонку первой страницы введите текст приглашения.

Текст приглашения может быть любым, торжественным или шутливым, но не короче 10 строк. Например:

Дорогие Юля и Миша!

Просим вас принять участие в торжествах по поводу нашего бракосочетания, которое состоится в воскресенье, 7 июля 2007 года.

Официальная часть начнется в 10 утра во Дворце бракосочетания № 1 "Грибоедовский" по адресу: Москва, Малый Харитоньевский пер., д. 10. Находится в пяти минутах ходьбы от метро, по Мясницкой улице в сторону Садового кольца до пересечения с Малым Харитоньевским переулком. Проезд до станции метро Чистые пруды.

Банкет намечен на 19 часов в ресторане «Веселые ребята», Сиреневый бульвар, 31.

Ждем вас.

Юля Гулькина и Михаил Соколов.

Для набора поместите с помощью мыши курсор в правую колонку первой страницы. Если колонка содержит пустые строки, удалите их.

Режим выравнивания текста, тип и начертание шрифта выберите по собственному усмотрению.

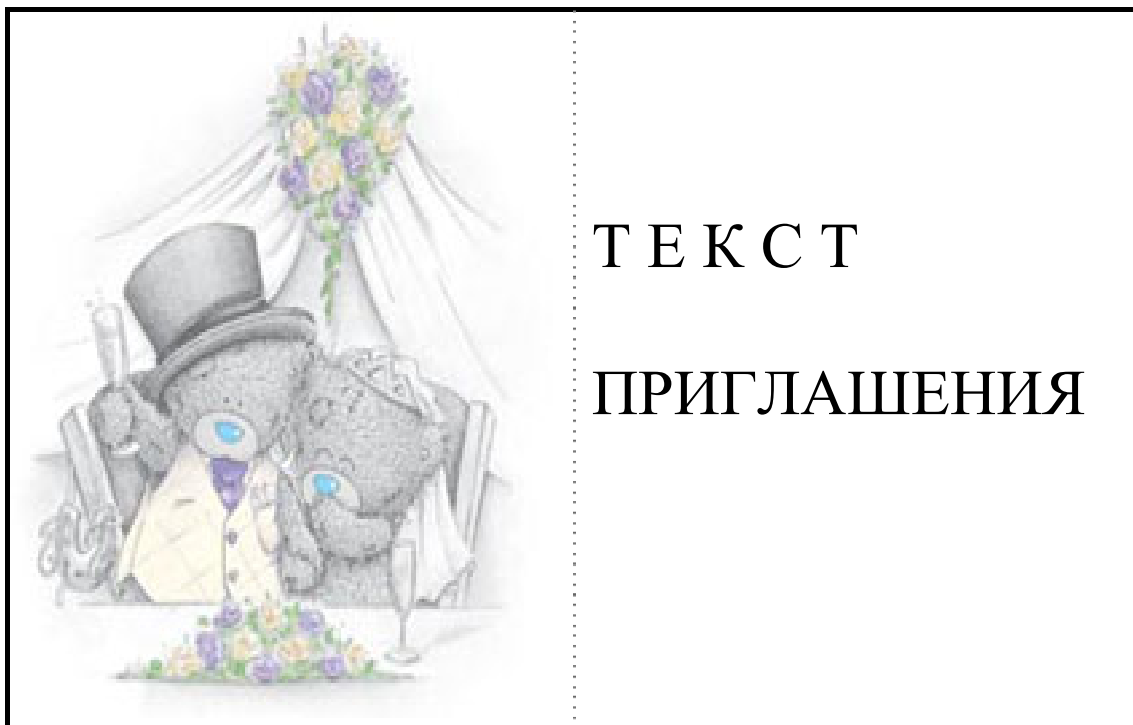
Размер шрифта установите не более 12 для Times New Roman и не более 10 для Arial, иначе текст не поместится. По этой же причине начинайте ввод с первой строки и не оставляйте пустых строк. В остальном технология ввода та же, что и при создании первого варианта приглашения.

В левую колонку второй страницы необходимо поместить реквизиты автора (группа, фамилия И.О.). Можно использовать ту же технологию, что и при вводе текста приглашения.

В правую колонку второй страницы введите текст титульного листа. Можно рекомендовать выравнивание текста **По центру**, размер шрифта не менее 24. Текст можно разместить симметрич-

но относительно центра колонки. Макет приглашения показан на рис. 1.34.

Первая страница (печатается на внутренней стороне листа)



Вторая страница (печатается на наружной стороне листа)

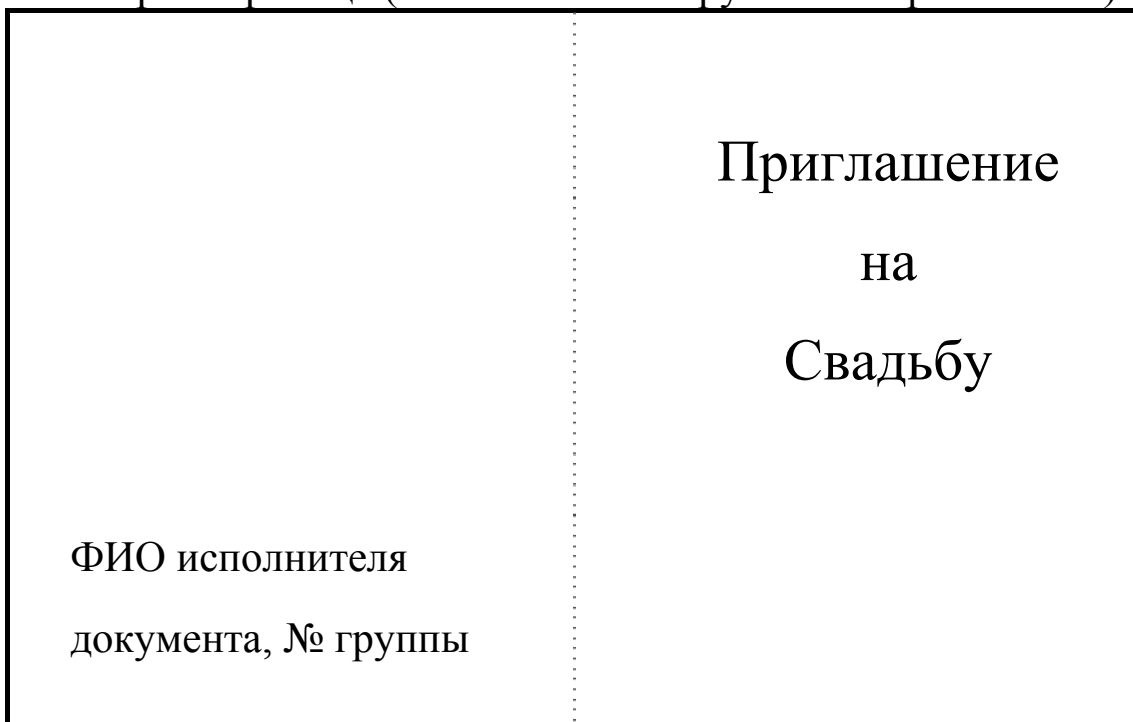


Рис. 1.34. Вариант приглашения на свадьбу.

Макет приглашения (после печати лист складывается вдвое по линии, обозначенной пунктиром).

Печать приглашения.

Способ размещения листа формата А5 в принтере зависит от типа принтера.

Убедитесь, что документ закончен и не содержит ошибок и опечаток.

1. Нажмите кнопку **Office** и в появившемся меню выберите команду **Печать - Печать**. На экране появится одноименное окно (рис. 1.35).

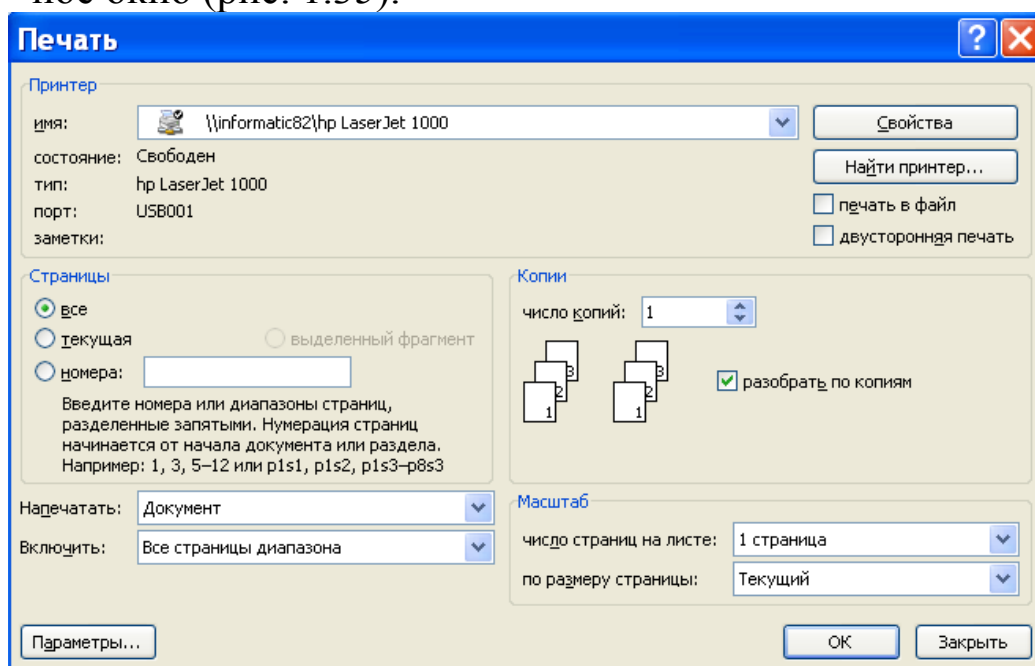


Рис. 1.35. Печать документа

2. В раскрывающемся списке имя области **Принтер** выберете принтер, на котором хотите распечатать документ.
3. В области **Страницы** установите переключатель в одно из следующих положений:
 - текущая – будет распечатана только первая страница приглашения, на которой в данный момент находится курсор.
4. Нажмите кнопку **ОК**. Окно закроется, а выбранный принтер начнет распечатывать приглашение.

Лист с отпечатанной лицевой стороной приглашения, не переворачивая, снова поместите в принтер. Установите курсор на вторую страницу документа и повторите предыдущие пункты инструкции. Отпечатается внутренняя сторона приглашения. Сложите приглашение пополам в виде открытки.

Задание №11. Получите у преподавателя номер и создайте одно из перечисленных ниже приглашений в виде книжки. Приглашения снабдите соответствующим рисунком. Не забудьте в левой колонке лицевой стороны приглашения указать свою группу и фамилию.

Номер варианта	Характер текста
0	Приглашение на именины
1	Приглашение на золотую свадьбу
2	Приглашение в суд в качестве свидетеля
3	Приглашение на похороны
4	Приглашение на дружескую вечеринку
5	Приглашение на собрание гаражного кооператива
6	Приглашение на учредительное собрание акционерного общества
7	Приглашение на уборку территории молодежного жилищного кооператива (МЖК)
8	Вызов на дуэль
9	Приглашение на кинофестиваль
10	Приглашение на юбилей
11	Приглашение на день рождения
12	Приглашение на выпускной вечер
13	Приглашение на встречу однокурсников
14	Приглашение на выборы
15	Приглашение на выставку компьютерных технологий
16	Приглашение на экскурсию в музей
17	Приглашение на открытие ресторана
18	Приглашение в клуб на Halloween
19	Приглашение на день открытых дверей в ВУЗ
20	Приглашение на чемпионат по теннису
21	Приглашение съёмки телесериала
22	Приглашение на дискотеку
23	Приглашение на фотосессию
24	Приглашение на пикник
25	Приглашение на корпоративную вечеринку
26	Приглашение на званый ужин
27	Приглашение на работу
28	Приглашение на литературный вечер

29	Приглашение на премьеру нового фильма
30	Повестка в военкомат

Задание № 12.

На листе А4 создайте рекламу любого вида товара или услуг используя следующие элементы: рисунки, объекты Word Art, колонтитулы, текст рекламы, набранный в документе Word. Варианты задания для создания рекламы необходимо получить у преподавателя.

Номер варианта	Характер текста
0	Реклама туристического агентства
1	Реклама книжной ярмарки
2	Реклама книжного магазина
3	Реклама зоомагазина
4	Реклама агентства недвижимости
5	Реклама концерта
6	Реклама музыкальной группы
7	Реклама фильма
8	Реклама музея
9	Реклама рыболовной базы
10	Реклама охотничьего хозяйства
11	Реклама магазина одежды
12	Реклама магазина обуви
13	Реклама строительного рынка
14	Реклама зоопарка
15	Реклама театра
16	Реклама торгового центра
17	Реклама развлекательного центра
18	Реклама ночного клуба
19	Реклама ресторана
20	Реклама кафе
21	Реклама фитнес-центра
22	Реклама дизайнерского бюро
23	Реклама университета
24	Реклама автомобиля российского производства
25	Реклама автомобиля иностранного производства
26	Реклама новой школы
27	Реклама спортивной школы
28	Реклама питомника для собак
29	Реклама пчеловодческого хозяйства
30	Реклама фермерского хозяйства

Алгоритм верстки отчета по теме Word в виде книжки.

Отчёт должен содержать все задания, выполненные по теме Word.

1. Число страниц в документе должно быть кратно четырем, т.е. равняться $4 \cdot n$. Если это условие не выполнено, то следует дополнить документ пустыми страницами до $4n$.
2. Выполняем команду: Кнопка **Office** > **Печать** > Предварительный просмотр и печать документа. (рис. 1.36).

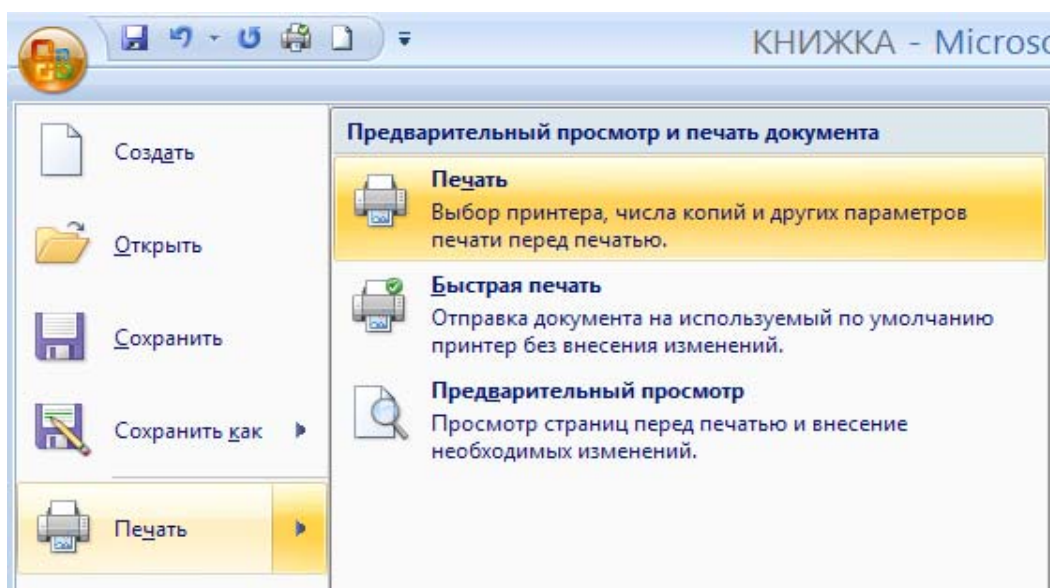


Рис. 1.36. Выбор режима печати

3. Появляется диалоговое окно. (рис. 1.37).

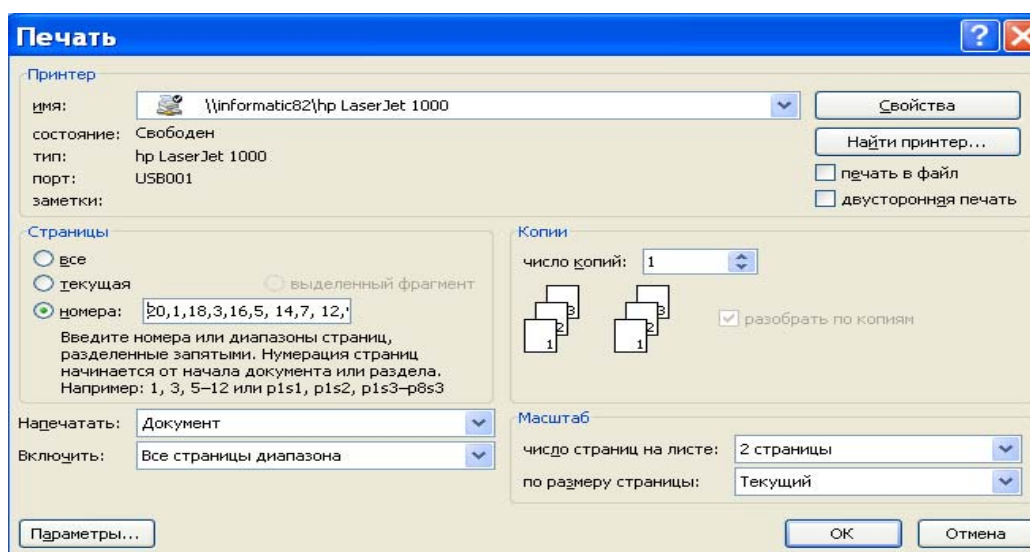


Рис. 1.37. Окно печати

В этом диалоговом окне на вкладке **Масштаб** устанавливаем «2 страницы». Кнопка **Office - Печать – параметры - дополнительно - Печать**, отключаем «печатать в обратном порядке» (снимаем галочку). На вкладке **Страницы** устанавливаем номера страниц в следующем порядке: $4n,1$; $4n-2,3$ и т.д. n -пар номеров страниц. Например, у нас имеется документ, состоящий из 20 страниц. Тогда $n=5$: $20,1$; $18,3$; $16,5$; $14,7$; $12,9$. Нажимаем кнопку **Enter**. Документ распечатается с одной стороны листа бумаги.

4. Вынимаем листы из принтера, вставляем их обратно с чистой стороны, не поворачивая.
5. Выполняем команду: Кнопка **Office > Печать > Предварительный просмотр и печать документа** (рис 1.36).
6. Появляется диалоговое окно. На вкладке **Масштаб** устанавливаем **2 страницы**. Кнопка **Office - Печать – параметры - дополнительно - печать** включаем **печать в обратном порядке**, ставим галочку (рис. 1.38).

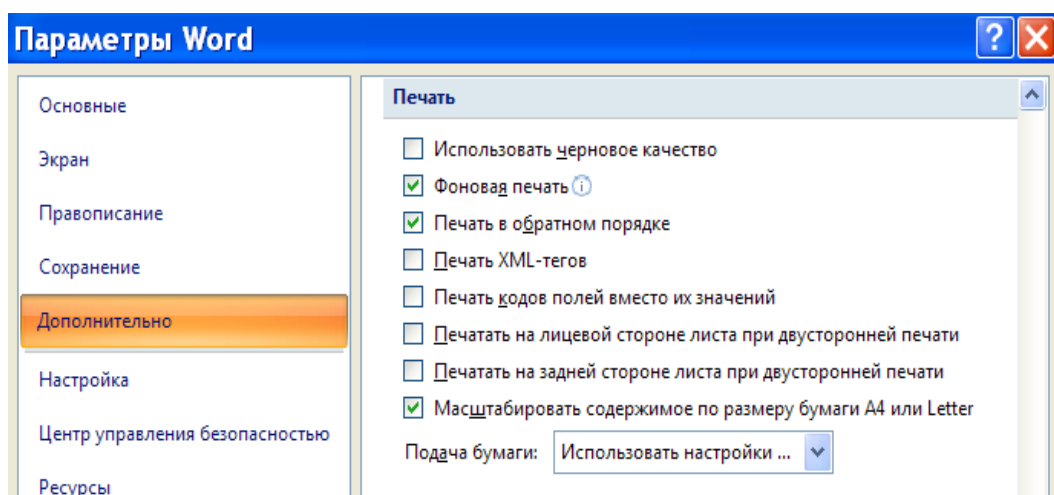


Рис. 1.38. Окно **Параметры Word**

7. На вкладке «Страницы» устанавливаем номера страниц в следующем порядке: $4n-1,2$; $4n-3, 4$ и т.д. Для вышеприведенного документа получается 5 пар $19,2$; $17,4$; $15,6$; $13,8$;

11,10. Нажимаем кнопку **Enter**. Документ распечатается с другой стороны листа бумаги.

Задание №13. Скопировать все выполненные лабораторные работы по Word в один файл «Отчёт по Word». Сформировать оглавление позволяющее переходить к любой лабораторной работе. Сделать отчёт в виде многостраничного документа, распечатать и оформить в виде книжки.

§9. Работа с документами на электронных носителях, в которых содержатся гиперссылки

Гиперссылки могут быть полезными при работе с электронной версией документа. Вы можете связать любой документ с другим документом, адресом электронной почты или веб-страницей. Таким образом, пользователь, работая с документом Microsoft Word, может просто щёлкнуть кнопкой мыши на фрагменте текста, отмеченном гиперссылкой, чтобы перейти к другой части загруженного документа, загрузить другой документ, веб-страницу или отправить сообщение по электронной почте. Мы рассмотрим каждый из этих трёх видов гиперссылок.

Гиперссылки на файл или веб-адрес.

Создадим гиперссылку для связи с файлом.

1. Введите фрагмент текста (слово или выражение).
2. Перейдите на вкладку **Вставка**.
3. Нажмите кнопку **Гиперссылка** в группе **Связи**. На экране появится окно **Вставка гиперссылки** (рис. 1.39).

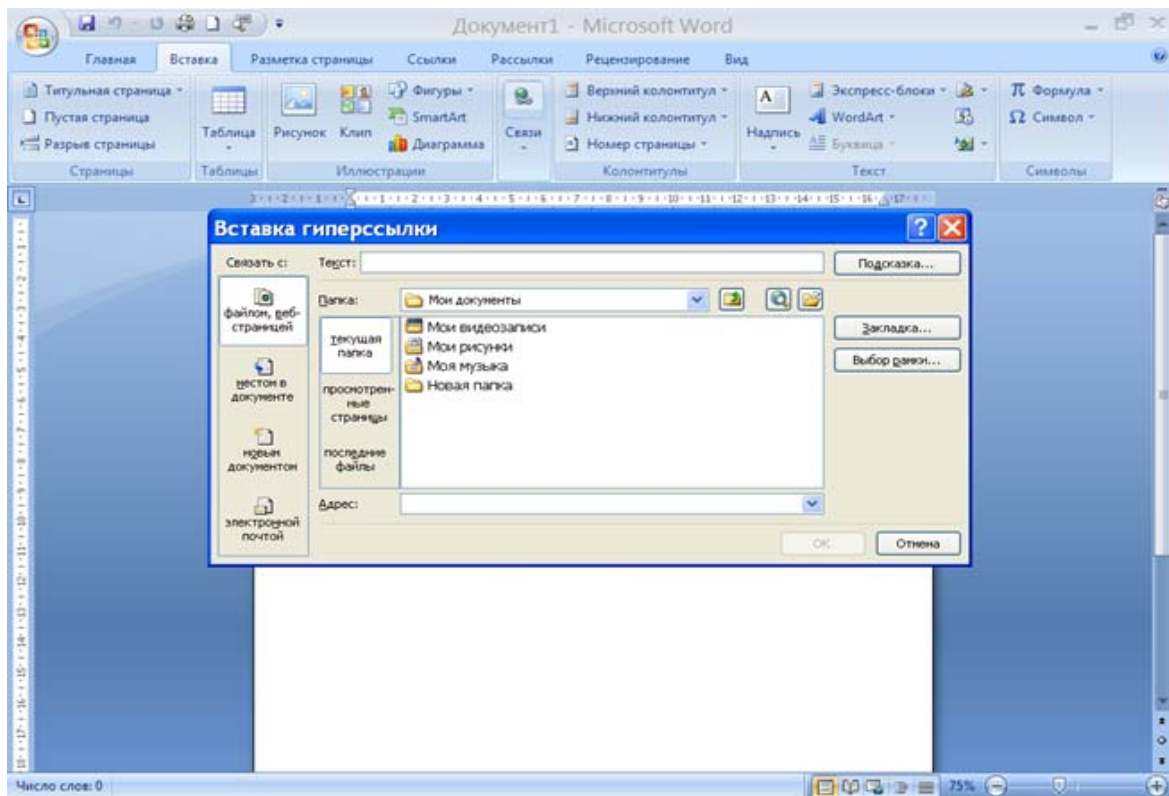


Рис. 1.39. Окно вставки гиперссылки

4. В группе **Связать с** щёлкните кнопкой мыши на пункте **файлом, веб-страницей**.
5. В раскрывающемся списке **Папка** выберите папку, содержащую файл, с которым вы хотите связать гиперссылку.

Вы можете создать гиперссылку не только на файл документа Microsoft Word, но и на любой другой, например, видео- или звуковой файл. При переходе по созданной вами в документе ссылке автоматически будет загружено приложение, нужное для открытия файлов этого типа. Например, если гиперссылка связана с видеофайлом, то он автоматически будет открыт в программе для воспроизведения видеофайлов, например, в **Проигрывателе Windows Media**.

6. В списке, расположенном чуть ниже выделенного фрагмента, выберите файл, который хотите связать с гиперссылкой.
7. Нажмите кнопку **ОК**. Окно будет закрыто, а форматирование выделенного фрагмента текста изменится: цвет

шрифта стал синий, а выделенный фрагмент подчёркнут. Это означает, что данный фрагмент текста связан с другим объектом гиперссылкой.

8. Установите указатель мыши над созданной гиперссылкой. На экране появится всплывающая подсказка, в которой указан путь к связанному со ссылкой файлу (рис. 1.40).
9. Нажмите клавишу Ctrl. Вид указателя изменится.
10. Удерживая нажатой клавишу Ctrl, щёлкните кнопкой мыши на гиперссылке. Приложение, ассоциированное с открытием связанного файла, будет запущено, а сам файл загрузится в эту программу.

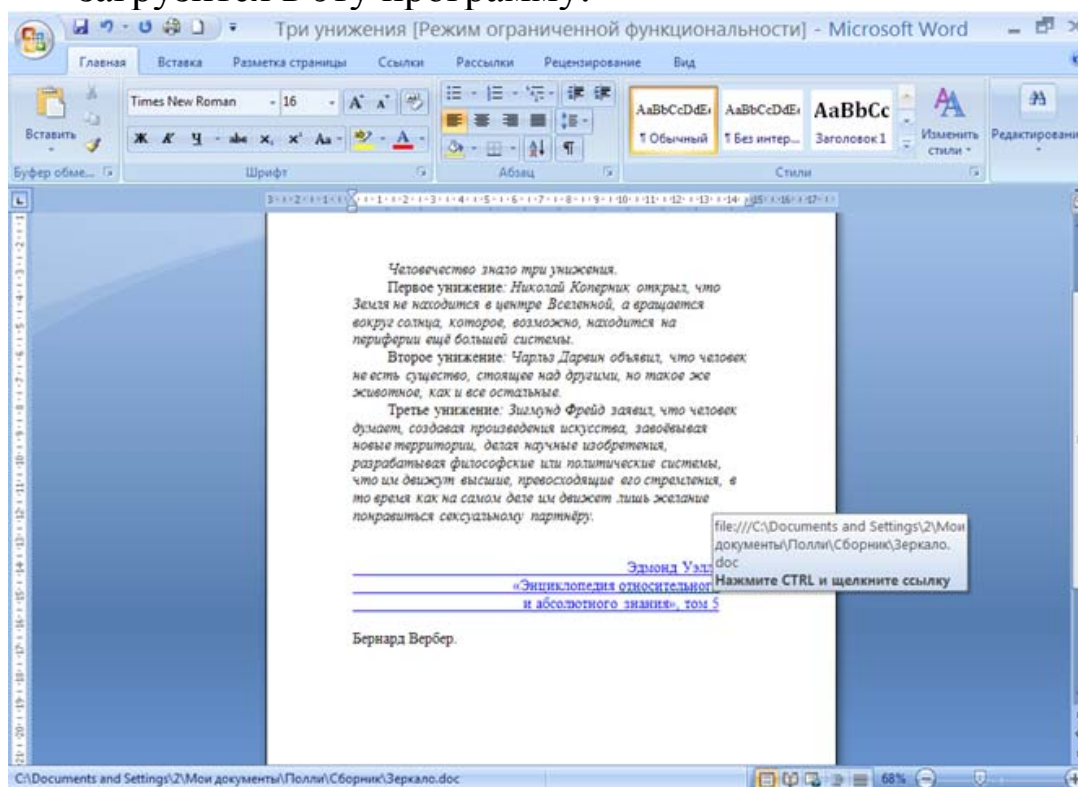


Рис. 1.40. Связь с гиперссылкой

Гиперссылки такого рода могут использоваться в различных областях. Например, если в документе встречается иностранное слово, вы можете связать его с соответствующим звуковым файлом, чтобы продемонстрировать правильное произношение этого слова.

Если в открывающемся списке **Адрес** окна **Изменение гиперссылки** ввести адрес веб-страницы, то при переходе по такой ссылке будет запущена *программа обозревателя Интернета*, а в неё будет загружена указанная веб-страница. При этом на компьютере должна быть установлена связь с Интернетом.

Гиперссылка на адрес электронной почты.

Часто при составлении деловых документов (договоров, коммерческих предложений и т.п.) в тексте указываются адреса электронной почты. Читатель такого документ может запустить установленную на компьютере программу для приёма и отправки электронной почты (например, Outlook Express), вручную создать и отправить письмо по указанному адресу. Использование гиперссылок может частично автоматизировать этот процесс.

Создадим гиперссылку на адрес электронной почты.

1. Введите в документ текст гиперссылки, например, **Свяжитесь с нами по этому адресу.**
2. Выделите введённый текст и нажмите кнопку **Гиперссылка** в группе **Связи**. На экране появится окно **Изменение гиперссылки**.
3. В группе **Связать с** выберите пункт **электронной почтой**. Содержание окна изменится (рис. 1.41).

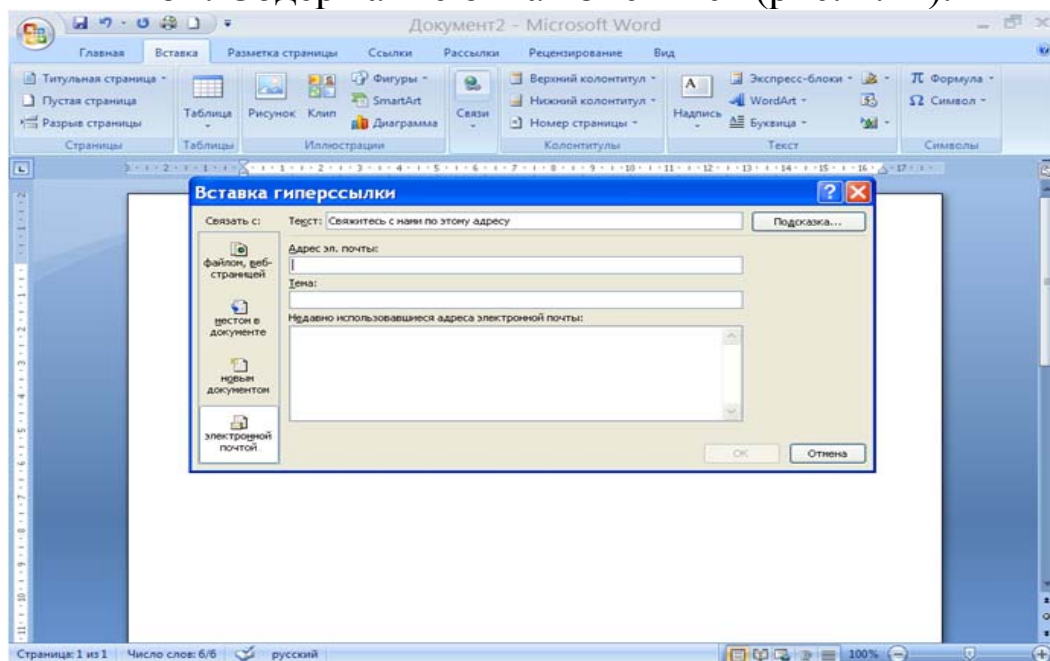


Рис. 1.41. Окно Вставка гиперссылки для указания адрес электронной почты

4. В поле **Адрес эл. почты** введите адрес, с которым хотите связать гиперссылку.
5. В поле **Тема** введите текст темы сообщения.
6. Нажмите кнопку **ОК**. Окно закроется, а выделенный текст будет помечен как гиперссылка.

7. Удерживая нажатой клавишу **Ctrl**, щёлкните кнопкой мыши на созданной гиперссылке. Программа, ассоциированная с отправкой электронных сообщений (например, Outlook Express), будет запущена. При этом автоматически будет создано новое сообщение, а в строке **Адрес** и **Тема** уже будут присутствовать данные, указанные вами в окне **Изменение гиперссылки**. Всё, что вам нужно только ввести текст сообщения и отправить его. Это удобно, поскольку тема письма уже заранее определена и адресат будет знать, по какому поводу пришло сообщение. Кроме того, вы не ошибётесь при вводе адреса электронной почты, поскольку программа «Microsoft Word» сама укажет адрес почтовой программе.

Гиперссылка на другой фрагмент текста.

Создайте ссылку на раздел документа.

1. Выделите в документе текст гиперссылки.
2. Нажмите кнопку **Гиперссылка** в группе **Связи**. На экране появится окно **Изменение гиперссылки**.
3. В области **Связать с** выберите пункт **местом в документе**. Содержание окна будет изменено. В поле **Выберите место в документе** отображается структура документа (рис. 1.42).

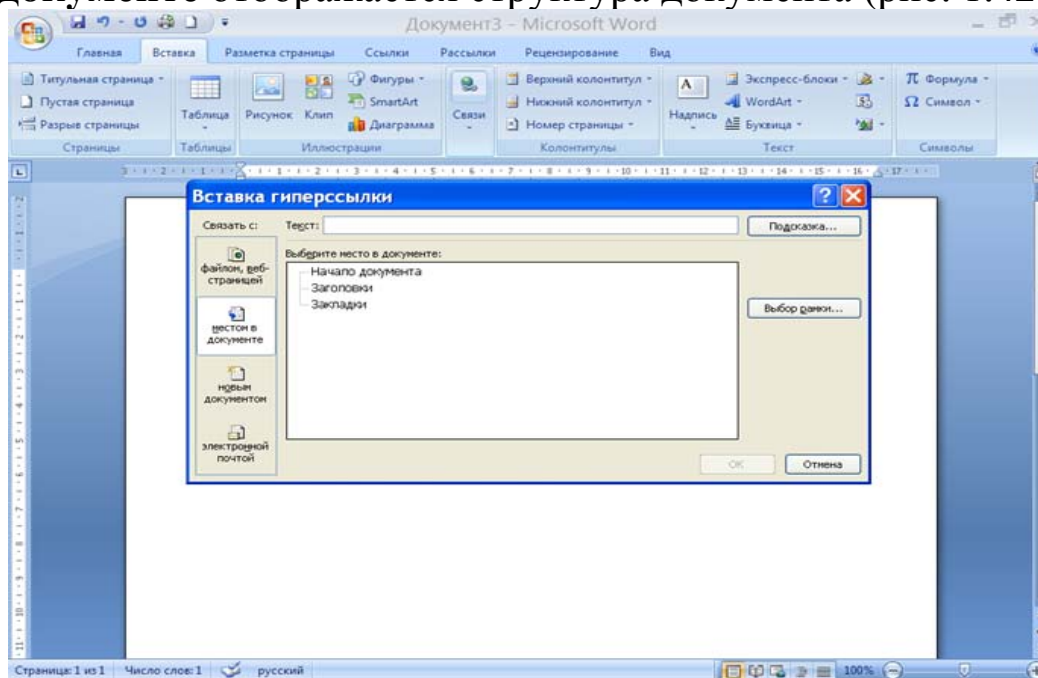


Рис. 1.42. Окно Вставка гиперссылки для указания фрагмента документа

4. Щёлкните кнопкой мыши на названии раздела (главы) документа в поле **Выберите место в документе**.
5. Нажмите кнопку **ОК**. окно закроется, а выделенный фрагмент будет помечен, как гиперссылка.
6. Нажав и удерживая клавишу «Ctrl», щёлкните кнопкой мыши на созданной гиперссылке. Текст документа будет прокручен до начала главы или раздела, на который указывает созданная гиперссылка.

Гиперссылки на место в документе могут быть полезны при составлении оглавления или предметного указателя. Вы можете создать гиперссылки для каждого пункта оглавления, указывающие на соответствующую главу, а читатель может быстро прокрутить документ до начала нужной главы, просто щёлкнув кнопкой мыши, удерживая клавишу «Ctrl», на пункте оглавления.

Если в документе названия всех разделов набраны стилем Заголовков, то, чтобы автоматически получить оглавление с гиперссылками достаточно выполнить **Вставка, Ссылка, Оглавление... ОК**.

Задание 14. Добавьте в общий файл заданий по теме «Word» задания №11 и №12. Скопируйте все выполненные задания по теме «Word» в один файл. Перед каждым заданием стилем Заголовков напишите его название и номер. Создайте оглавление, позволяющее переходить к нужному заданию. Введите ссылки внутри текста, позволяющие возвращаться из конца каждой работы в её начало.

§10. Microsoft Power Point

Microsoft Power Point – удобное средство создания презентаций для сопровождения выступления на конференции или на семинаре и для других видов представления информации. Эта программа позволяет включать в презентацию различные графические объекты, видеосюжеты, звуковое оформление, анимацию.

На рис. 1.43 представлены форматы файлов, используемые для сохранения презентаций.

Тип файла	Расширение	Используется для сохранения
Презентация	.ppt	Обычной презентации Microsoft PowerPoint
Метафайл Windows	.wmf	Слайда в виде графики
Рисунок в формате GIF	.gif	Слайда в виде графики для использования на веб-страницах
Рисунок в формате JPEG	.jpg	Слайда в виде графики для использования на веб-страницах
Рисунок в формате Portable Network Graphics	.png	Слайда в виде графики для использования на веб-страницах
Структура, RTF	.rtf	Содержимого презентации в виде документа структуры
Шаблон презентации	.pot	Презентации в виде шаблона
Демонстрация PowerPoint	.pps	Презентации, которая всегда будет открываться в режиме показа слайдов
Веб-страница	.htm; html	Веб-страницы в виде папки с .htm-файлами и всеми вспомогательными файлами
Веб-архив	.mht; mhtml	Веб-страницы в виде отдельного файла, включающего все вспомогательные файлы

Рис. 1.43. Форматы файлов, используемые для сохранения презентаций

Рассмотрим процесс создания презентации.

1. *Начало работы*

При запуске **Power Point** автоматически создается новая презентация, состоящая из одного слайда (рис. 1.44).

2. *Создание слайдов*

Презентация обычно состоит из нескольких слайдов. Добавить новые слайды можно следующим образом: **Главная** → **Создать слайд** → **Выбрать вид слайда**, например **Заголовок и объект** (рис. 1.44). Если вид следующего слайда должен быть таким же, как у предыдущего, можно применить команду **Дублировать выделенные слайды** (под предлагаемыми видами).

Еще один способ добавления слайда – находясь в поле слайдов, выбрать команду **Создать слайд** в контекстном меню.

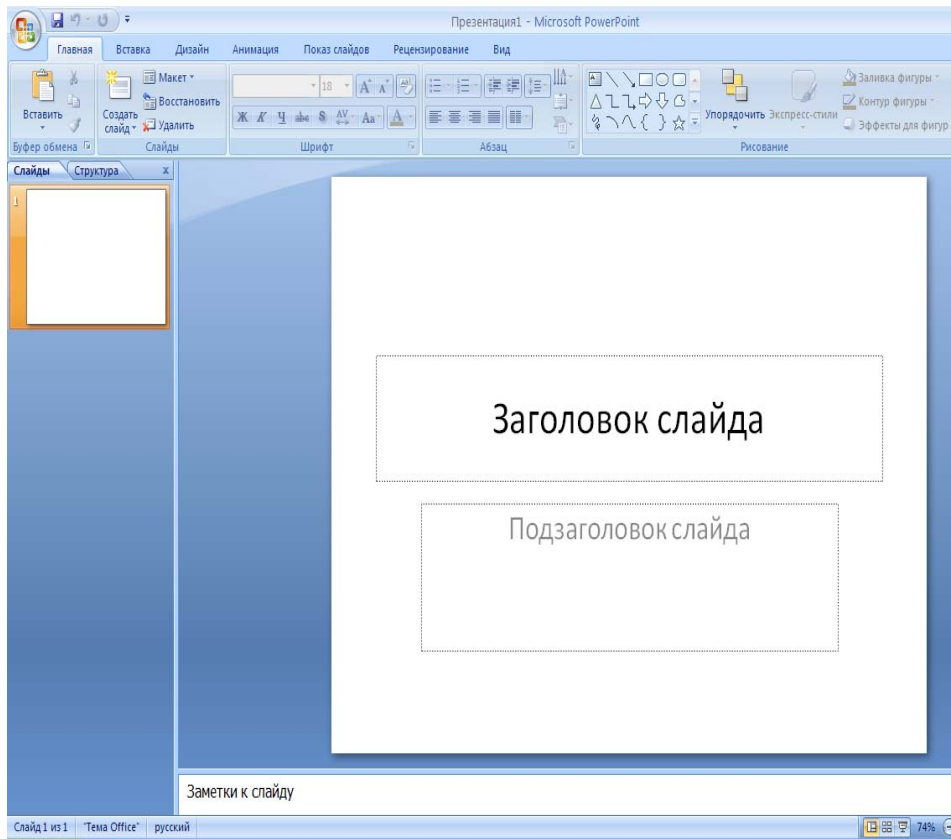


Рис. 1.44. Поле слайдов

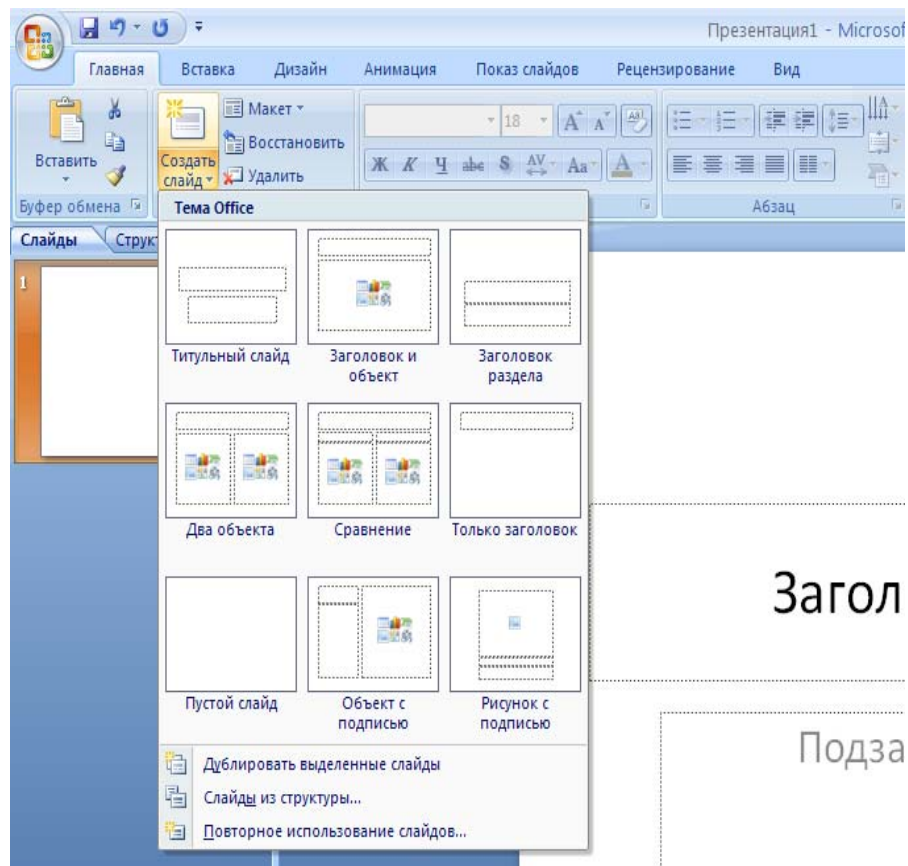


Рис. 1.45. Добавление новых слайдов

3. Заполнение слайдов информацией

Информацию в слайды можно перенести с помощью буфера обмена из документов, созданных в Word, либо вставить отдельные слайды из других презентаций.

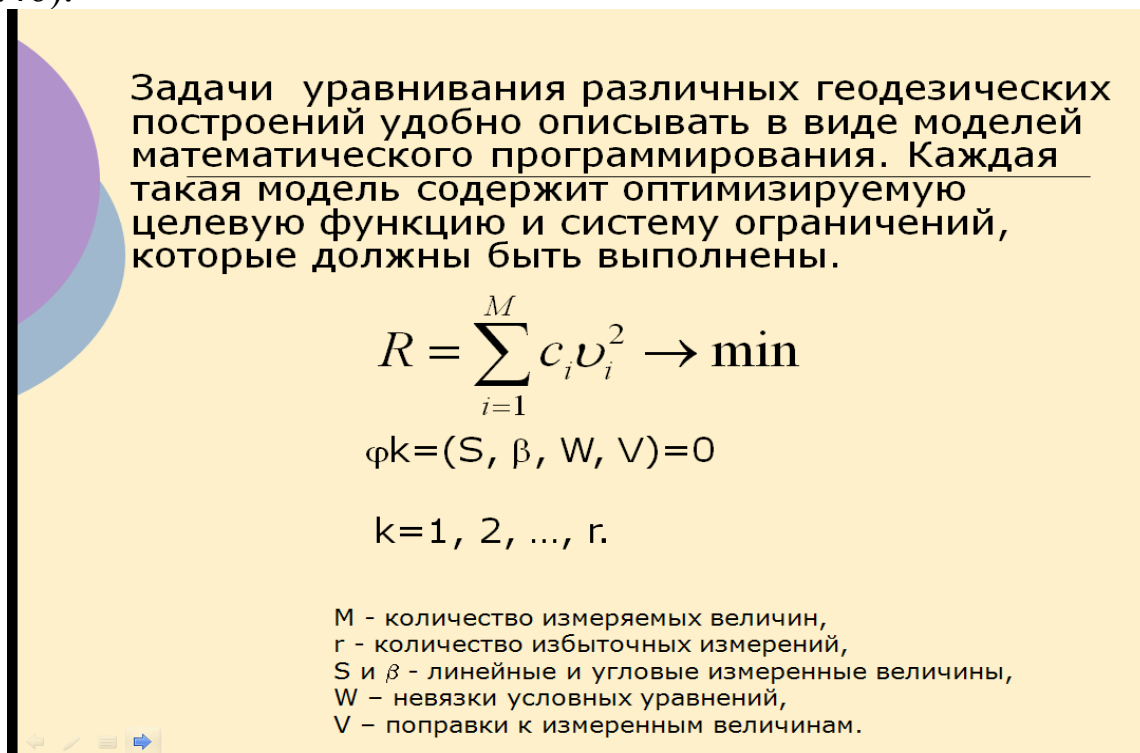
Форматирование и редактирование текстовой и графической информации, вставка объектов выполняются так же, как в Word.

4. Показ презентации

Предварительно можно настроить режим показа презентации вручную или по времени (**Показ слайдов** → **Настройка демонстрации**). При установке показа **По времени** сначала в репетиционном режиме определяется время выполнения презентации и при завершении сохраняется (**Показ слайдов** → **Настройка времени**).

Показ можно начать либо с первого слайда (**Показ слайдов** → **С начала**), либо с текущего - (**Показ слайдов** → **С текущего слайда**), либо в произвольном порядке (**Показ слайдов** → **Произвольный показ**).

При ручном показе для перехода к другому слайду нужно щелкнуть в любом месте слайда, либо подвести курсор в левый нижний угол и щелкнуть по нужной стрелке (вперед, назад) (рис. 1.46).



Задачи уравнивания различных геодезических построений удобно описывать в виде моделей математического программирования. Каждая такая модель содержит оптимизируемую целевую функцию и систему ограничений, которые должны быть выполнены.

$$R = \sum_{i=1}^M c_i v_i^2 \rightarrow \min$$
$$\varphi_k = (S, \beta, W, V) = 0$$
$$k = 1, 2, \dots, r.$$

M - количество измеряемых величин,
r - количество избыточных измерений,
S и β - линейные и угловые измеренные величины,
W - невязки условных уравнений,
V - поправки к измеренным величинам.

Рис. 1.46. Режим показа презентации

Несколько слайдов готовой презентации показаны на рис. 1.47.

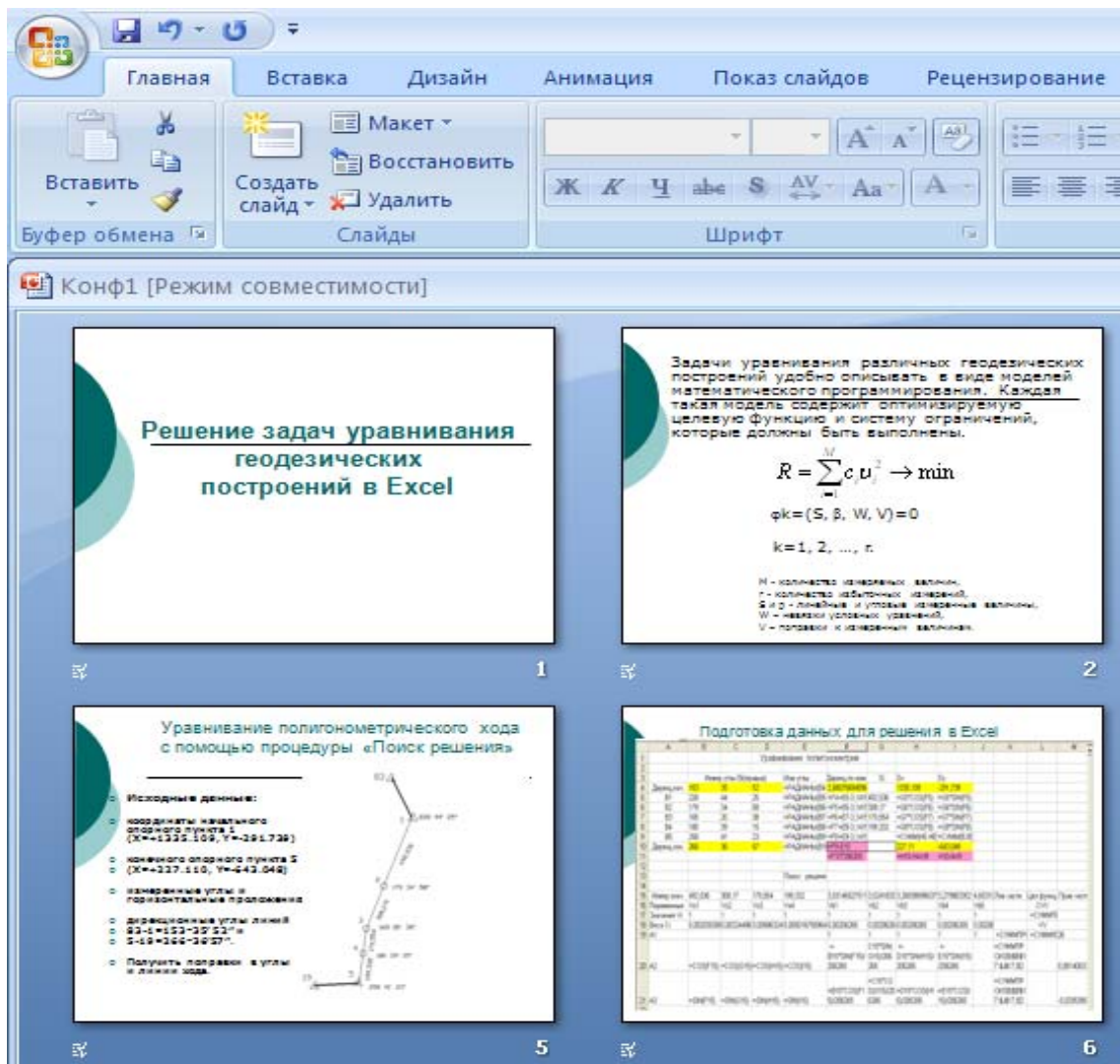


Рис. 1.47. Пример нескольких слайдов готовой презентации приведен

Задание №15. Создать презентацию из всех своих лабораторных работ. На первом слайде указать название «Отчет по теме «Текстовый редактор Word», ФИО и номер группы автора презентации, ФИО преподавателя, число, месяц, год выполнения работы.

лом (desktop). На рабочем столе размещены значки наиболее часто используемых программ, папок, документов. Под значком находится название элемента, если название длинное, то оно прерывается многоточием. Щелкнув два раза по значку, можно быстро запустить программу или открыть документ.

Некоторые значки появляются на рабочем столе в момент установки той или иной программы. Другие нужно создавать, например, ярлыки. Если у значка в левом нижнем углу есть стрелочка — это ярлык. Ярлык — это ссылка на программу.

Внизу находится панель задач (taskbar), которая используется для информирования пользователя о работе компьютера и обеспечения быстрого запуска и закрытия программ. На панели задач находится кнопка Пуск (Start) и кнопки каждой открытой в данный момент программы. В правой части панели задач находится область заметок. По умолчанию в области заметок выводится время и индикатор регистра языка, но могут появляться другие значки. Если все значки не видны, в области заметок появляется круглая кнопка со стрелкой. Щелкнув по ней, можно раскрыть список тех значков, которые не видны. Если навести указатель мыши на значок, то через некоторое время появляется подсказка о назначении этого значка.

На панели задач также находится панель инструментов быстрого запуска, на ней располагаются значки некоторых программ.

Упражнение 1. Определите назначение значков, находящихся на панели инструментов быстрого запуска и в области заметок панели инструментов.

Строение окна папки

Все окна Windows имеют похожее строение (рис. 2.2). Вверху заголовков, где в данном случае выведено название папки. В правой части строки заголовка находятся три кнопки. Красная кнопка с крестиком — кнопка закрытия окна. Кнопка, где изображен прямоугольник с утолщенным верхом — развернуть. Если щелкнуть по кнопке Развернуть, она изменится на кнопку, где изображены два прямоугольника — это кнопка Восстановить. Кнопка, на которой изображен знак подчеркивания — это кнопка Свернуть.

Под строкой заголовка находится главное меню. Здесь есть все команды, которые можно выполнить с содержимым папки. В зависимости от того, как настроен вид просмотра папки, под

главным меню может располагаться панель инструментов. Краткую справку о назначении кнопок можно получить, задержав указатель мыши на кнопке на несколько секунд.

Кнопка Назад (Back) возвращает пользователя в предыдущее место. Кнопка Вперед (Forward) переместит вас вперед, если вы возвращались Назад. Кнопка Вверх (Up) переместит вас на один уровень вверх. Кнопка Поиск (Search) позволяет по имени или части имени найти файл или папку на компьютере. Кнопка Папки (Folders) открывает список команд позволяющих менять настройки окна.

Файлы и папки в окне можно осматривать разными способами, для этого нужно изменить вид папки. Как это сделать? На панели инструментов есть кнопка Вид (Views). Меню Вид(Views) содержит команды по изменению вида просмотра файлов и папок.

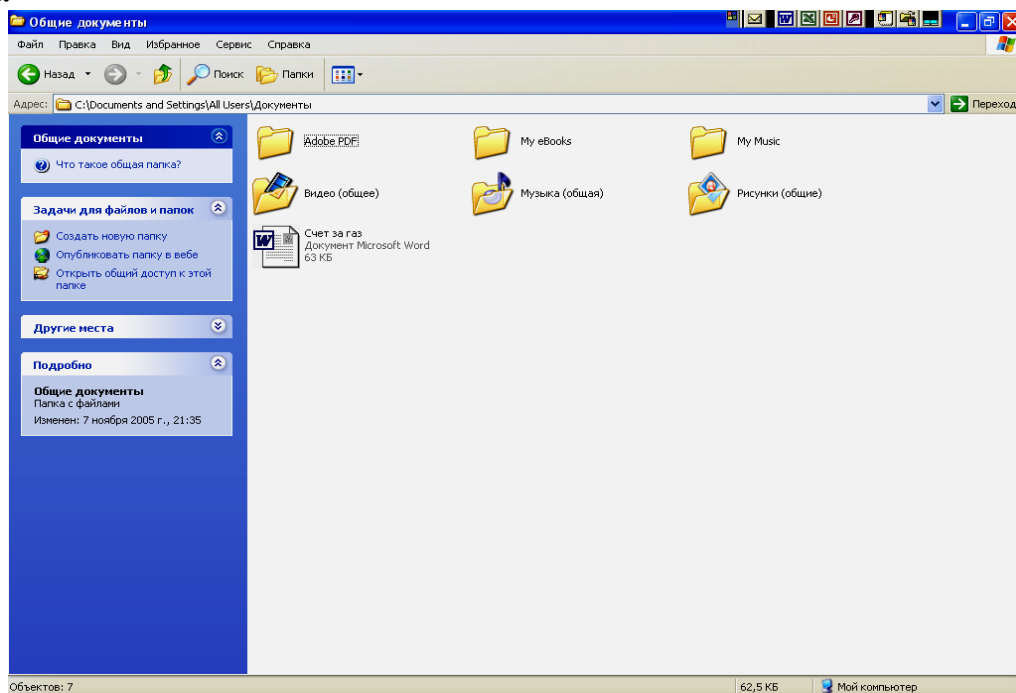


Рис. 2.2 Строение окна папки

Вид Значки (Icons) отображает значки и имя файла для каждого файла в текущей папке.

Вид Список (List) показывает список файлов и папок в текущей папке без всякой дополнительной информации за исключением имени файла или папки и маленького сопровождающего значка.

Вид Эскизы страниц (Thumbnails) показывает до четырех небольших презентаций файлов, содержащихся в каждой папке.

Вид Таблица (Details) показывает список файлов и папок, а также их свойства. Свойства, отображаемые по умолчанию: Имя (Name), Размер (Size), Тип (Type), и Дата изменения (Date Modified).

В Таблице пользователь может показать ряд других свойств, например, Автор (Author), Защита (Protected), Модель камеры (Camera Model) и др. Это можно сделать через меню контекстное меню, которое появляется при щелчке по правой кнопке мыши на заголовке таблицы.

В режиме Таблицы очень удобно сортировать файлы. Например, если вы хотите отсортировать файлы в порядке возрастания размеров, достаточно один раз щелкнуть по заголовку Размер (Size). Второй щелчок там же приведет к упорядочиванию файлов в порядке убывания размеров.

Упражнение 2. Щелкните по значку Мой Компьютер на Рабочем столе. Откройте папку Мои документы. Установите вид Таблица. Отсортируйте файлы по именам в порядке, обратном алфавитному.

В Windows XP, при любом установленном виде, информацию о типе файла и его размере можно получить, задержав на несколько секунд указатель мыши на его имени.

Под панелью инструментов располагается адресная строка, в которой выведено полное имя (с указанием пути) открытой папки, например W:\Кафедра информатики \ Практикум по информатике.

В самом низу находится статусная строка. В нее выводится информация о выделенном объекте. Если статусной строки нет, ее можно установить с помощью команды **Статусная строка (Status Bar)** из меню **Вид (Views)**.

Корзина

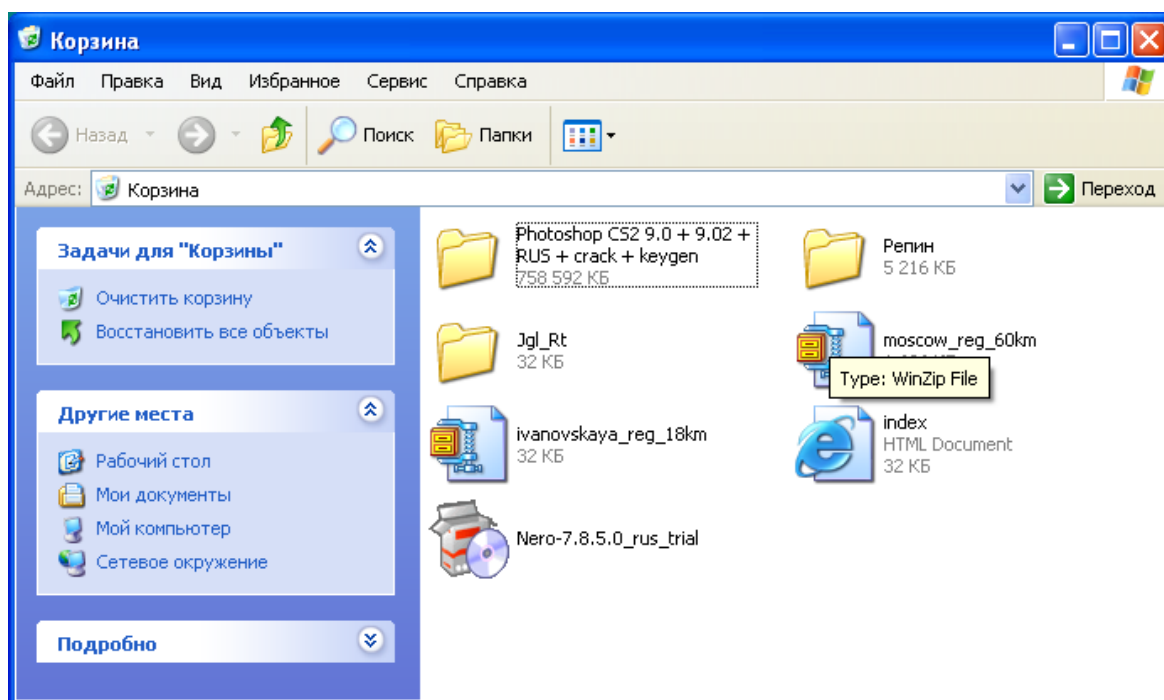


Рис. 2.3. Корзина

Один из значков на рабочем столе — Корзина (Recycle Bin). Корзина служит для временного хранения удаленных файлов. Корзина является «страховкой» для пользователя. Если пользователь случайно удалит нужные ему файлы, он может зайти в Корзину и их восстановить.

Если в Корзине установлен параметр — «самоочищающаяся», то файлы сразу пропадают из корзины и их восстановить нельзя.

Упражнение 3. Откройте Корзину. Если в ней есть файлы, попробуйте один из них восстановить. Если у Корзины установлен параметр «самоочищающаяся», отмените его.

Епощлиoadьидвли

§2. Общие сведения о программе Проводник

В Windows XP, так же как и предыдущих версиях, есть программа Проводник (Windows Explorer). Проводник предназначен для просмотра всех файлов, папок, дисководов и периферийных устройств, которые есть на компьютере. Когда вы открываете окно Мой компьютер, на самом деле это работает программа Про-

водник. Проводник можно вызвать, щелкнув правой кнопкой мыши по значку Мой компьютер и из контекстного меню выбрать Проводник. Или используя кнопку Пуск→ Все программы →Стандартные →Проводник.

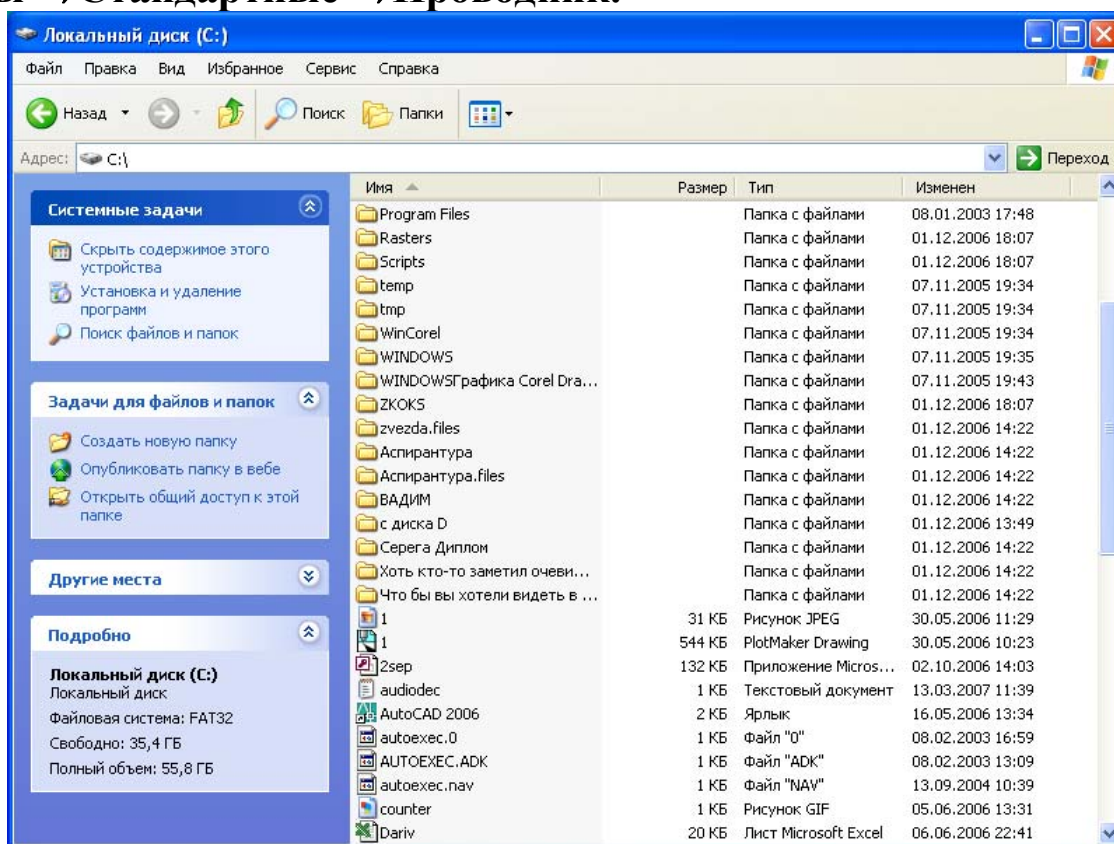


Рис. 2.4 Проводник

Проводник может работать в двух видах:
— вид Задачи (Tasks) отображает ссылки на задания и месторасположения, которые на данный момент имеют отношение к просматриваемой папке (рис. 2.4). Задания и месторасположение автоматически обновляются в зависимости от содержимого папки. Вид Задачи установлен по умолчанию. Если установлен вид Папки, нужно щелкнуть по кнопке с крестиком в заголовке панели Папки и тогда вы вернетесь к виду Задачи;
— вид Папки (Folders) отображает иерархическую структуру файлов, папок и вложенных папок, а также дисководов и периферийных устройств компьютера (рис. 2.5). Если компьютер подключен к сети, то видны сетевые диски и другие компьютеры. Вид Папки устанавливается щелчком на кнопке Папки панели инструментов.

Проводник позволяет вам не только видеть, что есть на вашем компьютере, но и обслуживать файловую систему, т.е. производить такие действия, как создание папок, ярлыков, удаление объектов (файлов, папок, ярлыков), переименование, копирование, перемещение.

Упражнение 4. Откройте Проводник и установите для него вид Папки.

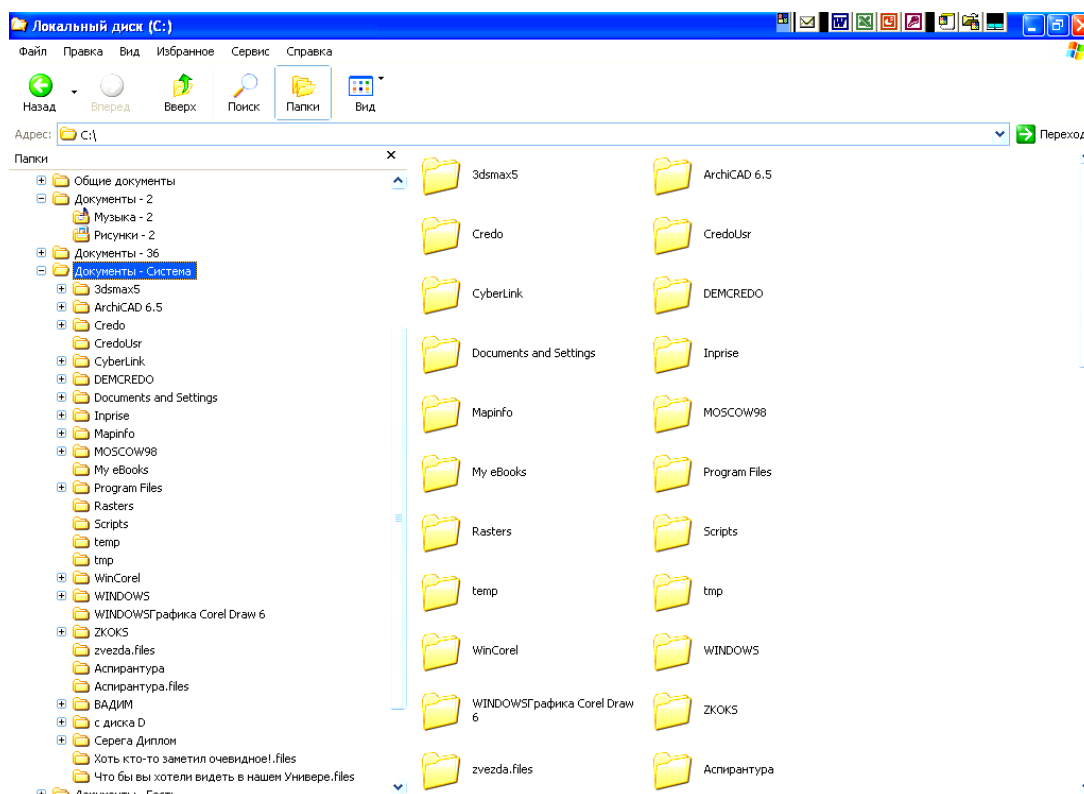


Рис. 2.5 Вид папки (C:) в Проводнике

§3 Сервисное обслуживание файловой системы

Создание папок

С помощью Главного меню — в меню Файл воспользоваться командой **Создать**→**Папку**. Стандартное имя Новая папка заменить на свое.

С помощью правой кнопки мыши (контекстное меню) — из контекстного меню выбрать **Создать**→**Папку**. Стандартное имя Новая папка заменить на свое.

Упражнение 5. На диске A: или своем съемном диске создать папку с названием «Фамилия И.О. студента» (т.е. указать свою

фамилию). На диске W: создать папки: Номер группы и в ней свою личную папку.

Создание ярлыков

Ярлык — это значок (обычно на рабочем столе), с помощью которого можно быстро получить доступ к нужному файлу или программе, не блуждая по иерархии папок.

В Windows XP рекомендуется создавать ярлыки с помощью Мастера (Create Shortcut Wizard). Мастер — это набор диалоговых окон, представляющих собой последовательность действий, которые надо выполнить, чтобы получить нужный результат. Для создания Ярлыка (Shortcut) щелкните правой кнопкой мыши на свободном месте рабочего стола. В контекстном меню выберите команду **Создать→Ярлык**. Откроется окно Мастера, которое называется «Создание ярлыка». В нем нужно указать размещение объекта. Это легко можно сделать с помощью кнопки Обзор, которая показывает иерархию папок. В следующем окне предлагается задать имя ярлыка. Если вас устраивает имя, данное по умолчанию (обычно соответствует имени файла), то можно просто нажать кнопку Готово. Ярлык появится на том месте рабочего стола, где вы щелкнули мышью.

Но можно воспользоваться командой **Создать ярлык**, которая есть в меню Вид каждого окна Windows.

Получить доступ к файлу с помощью Ярлыка, можно два раза щелкнув по стрелочке, расположенной в низу слева у значка Ярлыка.

Если вам не нравится сам значок, его можно заменить. Для этого щелкните по значку Ярлыка правой кнопкой и из контекстного меню выберите команду **Свойства**. В открывшемся окне на вкладке Ярлык нажмите кнопку Сменить значок. В окне Смена значка будут выведены разные значки. Вы можете выбрать понравившийся вам значок и нажать кнопку ОК. Значок Ярлыка на рабочем столе будет заменен.

Упражнение 6. Создайте на рабочем столе ярлык для файла «Практикум по информатике», который находится на диске I: в папке «Кафедра информатики». Установите другой значок для Ярлыка.

Выделение файлов и папок

Выделение одного файла — навести указатель мыши на название файла (папки) и щелкнуть левой клавишей мыши.

Многие операции по обслуживанию файловой системы осуществляются с группами файлов и папок.

Выделение группы смежных файлов — навести указатель мыши на название первого файла (папки) из группы выделяемых, щелкнуть на нем левой клавишей мыши, удерживая клавишу **Shift**. Далее, удерживая клавишу **Shift**, щелкнуть левой клавишей мыши по последнему файлу (папке) из группы. Можно просто, удерживая левую клавишу мыши, обвести рамкой группу выделяемых файлов (папок).

Выделение группы несмежных файлов — удерживая клавишу **Ctrl**, щелкнуть левой клавишей мыши по всем файлам (папкам), которые нужно выделить.

Выделить все файлы в открытой папке можно с помощью команды **Выделить все** из меню Правка или нажать сочетание клавиш **Ctrl+A**.

Упражнение 7. Выделить все файлы на своей дискете или съемном диске (Flash). В папке «Мои документы» потренируйтесь выделять группы смежных файлов и тех, которые располагаются вразброс.

Поиск файлов и папок

В Windows XP есть Помощник по поиску (Search Companion), который помогает искать все типы объектов, включая файлы, принтеры и компьютеры. Помощник по поиску представляет собой анимированного героя, по умолчанию — это собачка, но можно выбрать волшебника, экскурсовода или спортсмена, который в дружественной форме общается с пользователем, помогая ему найти нужные файлы и папки. Анимированного героя можно отключить.

Помощник по поиску включает свойство Indexing Service, которое индексирует файлы на компьютере, тем самым ускоряя процесс поиска.

Помощника по поиску можно вызвать через меню Пуск → Поиск. На экране появится окно, представленное на рис. 2.6.

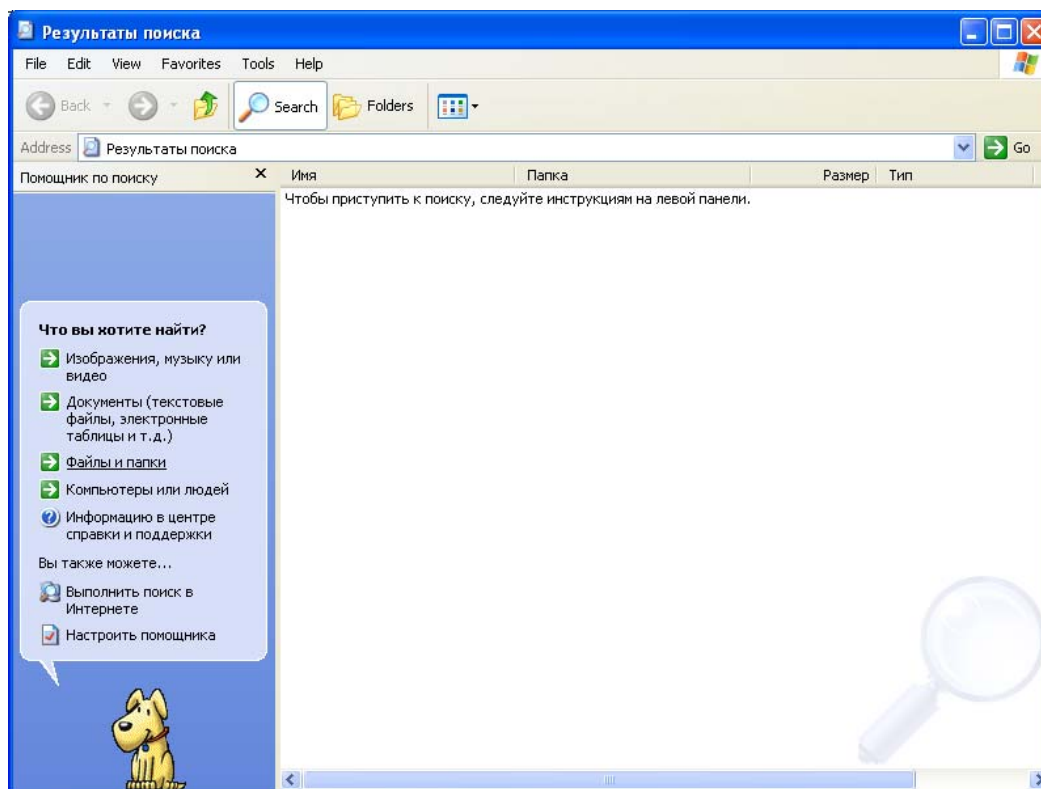


Рис. 2.6 Окно результаты поиска

Если вы собираетесь искать файл или папку, то на левой панели выберите команду Файлы и папки. В следующем окне нужно указать часть имени файла или имя файла целиком, можно указать слово или фразу в файле, а так же выбрать диски, где нужно осуществить поиск. Задать часть имени файла можно согласно следующим правилам. Звездочка (*) – это маска, представляющая любое количество символов, включая ноль. Знак вопроса (?) обозначает один любой символ. Например, указав *.bmp и нажав кнопку Поиск, мы запустили поиск всех файлов, у которых расширение .bmp . Результаты поиска будут выведены в следующее окно.

Упражнение 8. Найти на компьютере файлы с расширением .jpg.

Переименование файла

Переименование файла (папки) — это одиночная операция, осуществлять ее для группы файлов (папок) нельзя.

С помощью Главного меню — в меню Файл выбрать команду **Переименовать**.

С помощью правой кнопки мыши (контекстное меню) — в контекстном меню выбрать команду **Переименовать**.

Два щелчка на имени файла — вокруг имени появляется рамка. Внутри рамки можно изменять имя файла или папки.

Упражнение 9. Переименуйте папку «Фамилия И.О. студента» на своей дискете в «Номер группы - номер студента по списку».

Копирование файлов

Предварительно копируемый файл или группа файлов должны быть выделены.

С помощью Главного меню — в меню Правка выбрать команду **Копировать**. Перейти в ту папку куда нужно скопировать и в меню Правка выбрать команду **Вставить**.

С помощью левой кнопки мыши — при буксировании файла, с нажатой левой кнопкой мыши в папку на другой диск, файл копируется. Если папка находится на том же диске, то происходит перемещение.

С помощью правой кнопки мыши (контекстное меню) — выделить файл. Щелкнуть правой кнопкой мыши и в появившемся контекстном меню выбрать команду **Копировать**. Перейти в папку, куда требуется скопировать файл, щелкнуть правой кнопкой и в появившемся контекстном меню выбрать команду **Вставить**.

С помощью клавиатуры — нажать сочетание клавиш **Ctrl+C**. Перейти туда, куда требуется скопировать файл и нажать сочетание клавиш **Ctrl+V**.

Упражнение 10. Выделите все файлы, созданные вами в Word и скопируйте их в папку «Мои документы».

Перемещение файлов

Предварительно перемещаемый файл или группа файлов должны быть выделены.

С помощью Главного меню — в меню Правка выбрать команду **Вырезать**. Перейти в ту папку куда копировать и в меню Правка выбрать команду **Вставить**.

С помощью левой кнопки мыши — при буксировании файла, с нажатой левой кнопкой мыши в папку на этом же диске, файл перемещается. Если папка находится на другом диске, происходит копирование.

С помощью правой кнопки мыши (контекстное меню) — выделить файл. Щелкнуть правой кнопкой мыши и в появившемся контекстном меню выбрать команду **Вырезать** или **Отправить**. Перейти в папку, куда требуется скопировать файл, щелкнуть правой кнопкой и в появившемся контекстном меню выбрать команду **Вставить**.

С помощью клавиатуры — нажать сочетание клавиш **Ctrl+X**. Перейти туда, куда требуется скопировать файл и нажать сочетание клавиш **Ctrl+V**.

Упражнение 11. Выделите три файла в папке «Мои документы» и переместите их на свой диск.

Удаление файлов

Предварительно удаляемый файл или группа файлов должны быть выделены.

С помощью Главного меню — в меню Правка выбрать команду **Вырезать** или в меню Файл команду **Удалить**.

С помощью левой кнопки мыши — удерживая левую кнопку, отбуксировать файл в Корзину.

С помощью правой кнопки мыши (контекстное меню) — в контекстном меню выбрать команду **Удалить**.

С помощью клавиатуры — нажать сочетание клавиш **Ctrl+X**.

Замечание.

Удаление файла, минуя Корзину — одновременно нажать клавиши **Shift + Del**.

Упражнение 12. Перемещенные файлы из предыдущего упражнения выделите и удалите.

Замечание.

Часто при копировании информации на диск или скачивании музыки и фильмов из Интернета возникает вопрос: «А хватит ли у меня места, чтобы все это разместить у себя на компьютере?»

Чтобы ответить на этот вопрос, нужно в папке «Мой компьютер» подсветить нужный диск, щелкнуть правой кнопкой мыши и выбрать команду Свойства. В открывшемся окне указано, сколько занято места и свободно на выбранном диске.

Упражнение 13. Определите, сколько свободного места осталось на вашем диске и диске D: компьютера.

§4 Программы архивации

Для пересылки файлов по Интернету часто возникает необходимость в их сжатии, т.е. уменьшении размера. Раньше проблема сжатия файла была актуальна при записи файлов на магнитные носители, потому что у них был недостаточный объем. Сейчас с появлением CD и DVD дисков острота этой проблемы значительно уменьшилась.

В Windows XP широко используются две программы сжатия данных — это WinRAR и WinZip. Если вы выделите файл или группу файлов и щелкнете правой кнопкой мыши, то в открывшемся контекстном меню вы найдете команду **Добавить в архив**. Щелчок по этой команде приведет к появлению диалогового окна, изображенного на рис. 2.7.

Современные программные средства создания и обслуживания архивов часто называют диспетчерами архивов. К основным функциям программ архивации относятся: создание новых архивов, добавление файлов в имеющийся архив, создание самораспаковывающихся архивов, извлечение файлов из архивов, защита архивов от просмотра и несанкционированной модификации и др.

Рассмотрим создание архивного файла на примере программы WinRAR. В поле Имя архива задать полное имя архивного

файла (с указанием диска и папки), если вас не устраивает данное по умолчанию (см. рис. 2.5). Выбрать формат архива, в нашем случае RAR, метод сжатия и метод обновления. Если вы хотите создать самораспаковывающийся архив, то поставьте галочку напротив SFX- архив. Далее нажать ОК. После того, как архивация закончится, в указанной вами папке появится файл со значком, изображающем стопку книг (если самораспаковывающийся архив, то на книги налажена толстая стрелка) и именем, которое вы указали.

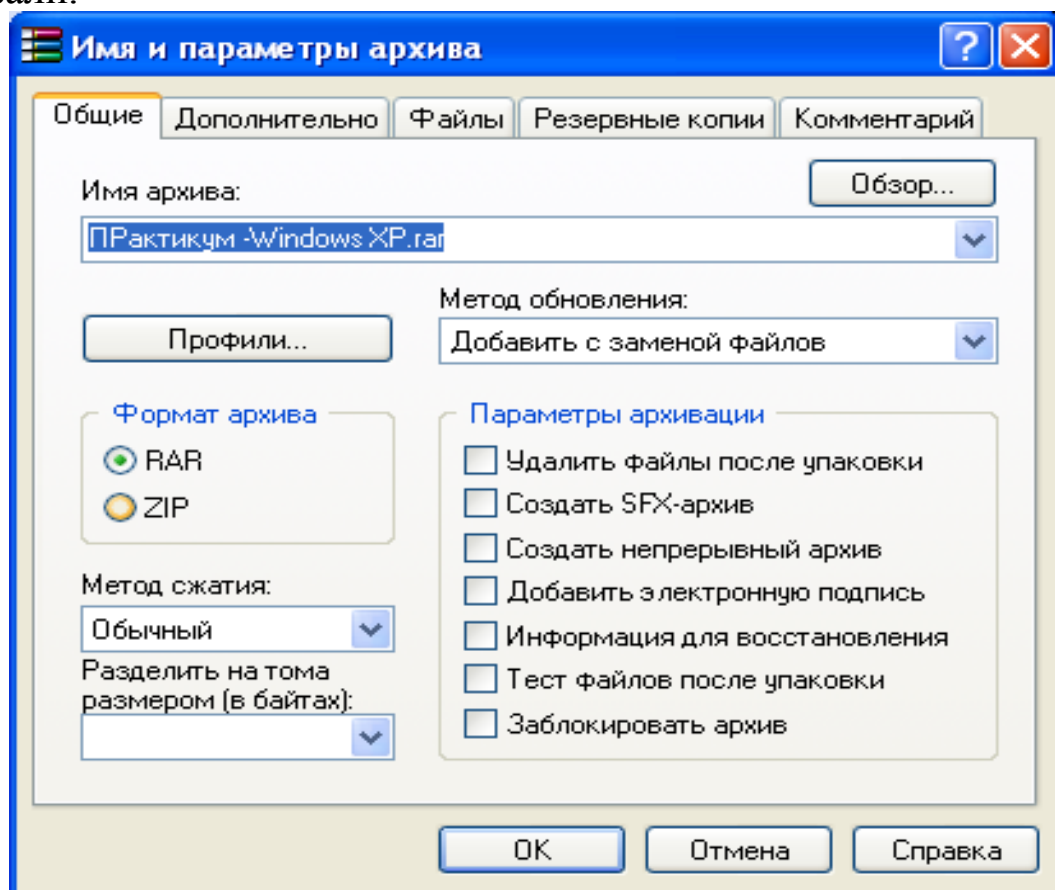


Рис. 2.7. Окно Имя и параметры архива

Распаковать архив просто, особенно самораспаковывающийся. Для самораспаковывающегося архива достаточно два раза щелкнуть по нему и в появившемся окне указать папку назначения. Если щелкнуть по изображению обычного архивного файла, то откроется окно программы WinRAR, в котором будут выведены названия заархивированных файлов в этом архиве. Для извлечения файлов, достаточно щелкнуть по кнопке Извлечь и указать Путь и параметры извлечения.

Упражнение 14. Создайте архивный файл, из файлов находящихся на диске D: в корневом каталоге.

§5. Несколько слов о Windows Vista

Для установки операционной системы Windows Vista, компьютер должен удовлетворять следующим минимальным требованиям:

- процессор 800 МГц;
- 512 Мб оперативной памяти;
- графический адаптер, совместимый с DirectX 9;
- жесткий диск объемом 20 Гб с 15 Гб свободного дискового пространства.

Для обеспечения большей производительности и поддержки расширенных возможностей, включая визуальные эффекты Windows Aero, компьютер должен отвечать следующим минимальным требованиям:

- процессор 1 ГГц;
- 1 Гб оперативной памяти;
- графический адаптер, совместимый с DirectX 9 с драйвером Windows Vista Display Driver Model (WDDDM), 128 МБ графической памяти; Pixel Shader 2.0;
- жесткий диск объемом 40 Гб со свободным дисковым пространством не менее 15 Гб;
- встроенный или внешний привод DVD-ROM;
- возможность подключения акустических систем;
- возможность доступа в Интернет.

Версия Windows Vista в отличие от Windows XP отличается не только новыми визуальными эффектами. По утверждению компании Microsoft, в ней выведены на новый уровень вычислительные возможности. Помимо всевозможных программ, помогающих следить за состоянием и защищенностью компьютера, в фоновом режиме выполняются процессы, сдерживающие действия вредоносных программ и злоумышленников, а также выводятся предупреждения в случае попытки программы или пользователя внести какие-либо изменения, которые могут повлиять на стабильность работы системы.

В Windows Vista появилось множество функций повышающих эффективность работы. Поиск файлов стал еще проще и быстрее: нужно просто ввести слово, присутствующее в искомом файле, и Windows Vista мгновенно отобразит все файлы в системе, содержащие это слово. Получение информации, необходимой для выполнения задачи, теперь не представляет никакой сложности: нужно просто щелкнуть на нужной ссылке справки в окнах Windows или диалоговых окнах, чтобы изучить необходимые возможности.

Замечание.

Те минимальные действия, которые вы освоили при изучении Windows XP, сохранились и в Windows Vista.

Задания для самостоятельного решения

Если вы делаете свое задание дома, то проиллюстрируйте ваши действия распечатками.

Вариант 1.

1) Создайте на диске A: папку с названием вашего факультета, а в ней подпапку Word.

2) Скопируйте в нее все файлы, созданные вами в редакторе Word.

3) Создайте в корневом каталоге папку «Архив» и внутри ее создайте архив файлов, созданных в Word.

4) Установите вид Таблица в окне папки в Проводнике.

5) Раскройте папку «Word» и отсортируйте файлы по дате последнего изменения.

Вариант 2.

1) Создайте на диске W: папку под названием «Моя библиотека», в ней подпапку «Мой факультет».

2) Переместите в папку «Мой факультет» все файл, созданные вами в Word.

3) Один из файлов, созданных в редакторе Word, переименуйте так, чтобы первой буквой была буква А.

4) На рабочем столе создайте ярлык для папки «Моя библиотека».

5) Скопируйте переименованный в предыдущем задании файл в папку «Мои документы» на компьютере.

Вариант 3.

1) В папке «Мои документы» на компьютере создайте папку «Фамилия И.О.» (подразумевается ваша фамилия).

2) В папке «Фамилия И.О.» создайте подпапки «Access», «Excel», «Delphi».

3) В папку «Word» скопируйте файлы, созданные вами в редакторе Word.

4) Один из файлов удалите в Корзину.

5) В Проводнике откройте папку «Корзина». Если в папке «Корзина» установлен Параметр «самоочищающаяся», то отмените его.

Вариант 4.

1) На диске D: создать папку «Архив\Фамилия студента».

2) На своей дискете, заархивировать файлы, созданные в редакторе Word.

3) Переместите архивный файл в папку «D:\Архив\Фамилия студента».

4) Переименовать файлы, созданные вами в редакторе Word, таким образом, чтобы первая буква в названии была W.

5) В Проводнике откройте вашу дискету в режиме Таблица и отсортируйте файлы по размеру в убывающем порядке.

Вариант 5.

1) На своей дискете создать папки: «Word», «Windows», «Excel», «Access», «Delphi».

2) В папке «Delphi» создайте подпапку «Практика».

3) Переместите в папку «Word» все созданные ранее вами файлы.

4) Откройте папку «Word» в Проводнике. Отсортируйте файлы по дате последнего изменения в убывающем порядке.

5) Индивидуальные задания по Word скопируйте в папку «Практика».

Вариант 6.

1) На своей дискете создайте папку «Документы».

2) В папке «Документы» создайте две подпапки: «Лабораторные работы», «Индивидуальные задания».

3) Скопируйте в папку «Лабораторные работы», выполненные вами в редакторе Word лабораторные работы, а в папку «Индивидуальные задания» – индивидуальные работы.

- 4) Удалите из корневого каталога своего диска все файлы, созданные в редакторе Word.
- 5) На рабочем столе создайте ярлыки для папок: Лабораторные работы, Индивидуальные задания.

Вариант 7.

- 1) На диске D: вашего компьютера создайте папку «Индивидуальные работы в Word» и в нее скопируйте с вашей дискеты индивидуальные задания по Word.
- 2) В Проводнике откройте папку «Индивидуальные работы в Word». Отсортируйте файлы по имени в порядке обратном алфавитному.
- 3) На своей дискете создайте папку «Архив лабораторных работ» и заархивируйте в нее все лабораторные работы.
- 4) На рабочем столе создайте ярлык для папки «Индивидуальные работы в Word».
- 5) Индивидуальную работу №5 по Word в папке «Индивидуальные работы в Word» переименуйте так, чтобы первой буквой в названии была первая буква вашей фамилии.

Вариант 8.

- 1) На диске A: создайте папки «Проба пера», «Мои новые документы», «Задачи».
- 2) В папке «Проба пера» создайте папку «Редактор текстов» и скопируйте в нее все файлы из корневого каталога диска A:, созданные в редакторе Word.
- 3) Удалите все файлы из корневого каталога диска A:.
- 4) Какой объем памяти занимает папка «Редактор текстов»?
- 5) Переименуйте один из ваших файлов так, чтобы первая буква совпала с первой буквой вашей фамилии.

Вариант 9.

- 1) В Проводнике откройте свою дискету и определите, сколько свободного места осталось на диске.
- 2) Создайте папки: «Word», «Delphi», «Excel», «Access».
- 3) На дисках C: и D: найдите файлы с расширением .pas.
- 4) На дисках C: и D: найдите файлы с расширением .xls и скопируйте их в папку «Excel».
- 5) Определите, сколько свободного места осталось на Вашем диске.

Вариант 10.

- 1) На своей дискете создайте папки: «Мои работы по Word», «Копия Мои документы».
- 2) В папку «Мои работы по Word» переместите все свои лабораторные работы и индивидуальные задания по Word.
- 3) Из папки «Мои документы» на компьютере скопируйте несколько файлов (не менее 5) в папку «Копия Мои документы».
- 4) Откройте в Проводнике папку «Копия Мои документы» в режиме Таблица и отсортируйте файлы по размеру.
- 5) Определите, сколько свободного места осталось на вашем диске.

Вариант 11.

- 1) На диске D: компьютера создайте папку «Мой Word» и скопируйте в нее свои лабораторные работы по Word.
- 2) В папке «Мои документы» компьютера создайте папку «Индивидуальные работы» и скопируйте в нее свои индивидуальные задания по Word.
- 3) Откройте папку «Мои документы» в Проводнике в режиме Таблица и отсортируйте файлы по размеру в порядке убывания.
- 4) Создайте на диске D: папку «Архив» и заархивируйте в нее все свои файлы, созданные в Word.
- 5) Определите, какой объем памяти занимает папка «Мои документы».

Вариант 12.

- 1) На диске D: компьютера создайте следующую структуру папок: «Городской кадастр» (в ней: «Word1», «Excel1», «Практика1»); «Земельный кадастр» (в ней: «Word2», «Excel2», «Практика2»); «Архитектура» (в ней: «Word3», «Excel3», «Практика3»); «Землеустройство» (в ней: «Word4», «Excel4», «Практика4»); «Юриспруденция» (в ней: «Word5», «Excel5», «Практика5»);
- 2) В папке, которая начинается со слова Word и относится к факультету, где вы учитесь, создайте папку «Ф.И.О. студента» (ваша фамилия) и скопируйте в нее свои файлы, созданные в редакторе Word.

- 3) В папке, начинающейся со слов «Практика» и относящейся к вашему факультету, создайте папку «Архив» и заархивируйте в нее свои файлы, созданные в редакторе Word.
- 4) Сколько свободного места осталось на диске D:?

Глава 3. Работа в среде табличного процессора Excel 2007

§1. Вид окна Excel 2007

Одним из наиболее широко распространенных приложений пакета Microsoft Office является табличный процессор Excel 2007. Файл, сформированный в этом приложении, может иметь любое имя, соответствующее ограничениям, накладываемым операционной системой. Его расширение «xls» устанавливается автоматически. После запуска программы в верхней части экрана появляется окно интегрированного пакета Microsoft Office, включающее строку заголовка, меню и области активной вкладки. Например, на рис. 3.0 при активной вкладке *Главная* видим области *Буфер обмена*, *Шрифт*, *Выравнивание*, *Число*, *Стили*, *Ячейки*, *Редактирование* и соответствующие им кнопки панелей инструментов.

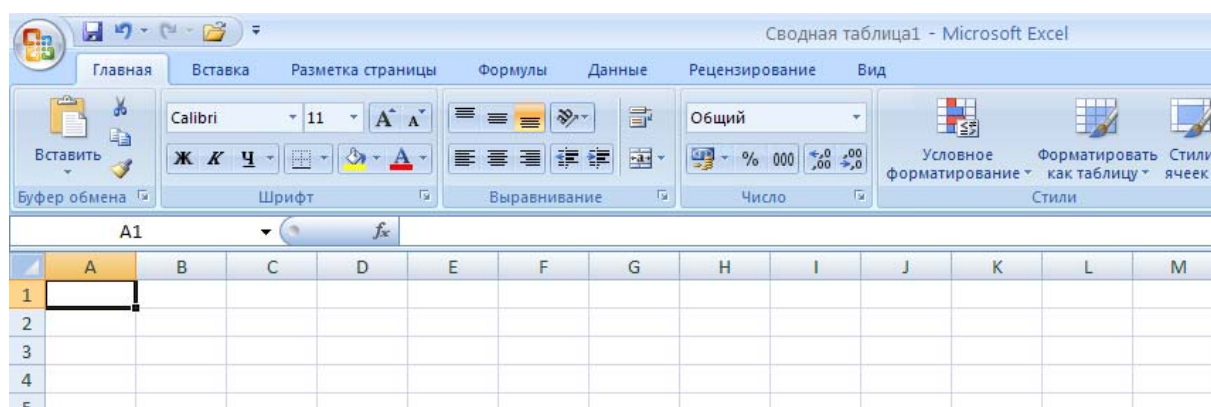


Рис. 3.0

Под областями располагается *информационная строка*. В её левой части находится *поле имени*, где записывается адрес или

имя выделенной ячейки, а в правой части — *поле формул*, содержащее информацию, внесенную в выделенную ячейку. В рабочей части окна расположена сетка, каждая ячейка которой характеризуется номерами строки и столбца, отсчитанными от верхнего левого угла. При этом по умолчанию строки маркируются цифрами, а столбцы — буквами или комбинациями букв латинского алфавита. Активное окно табличного процессора Excel 2007 называется рабочим листом, а создаваемый программой файл или документ — рабочей книгой. Если требуется сослаться на ячейку таблицы, расположенной на другом листе, то её адрес записывают в виде *Имя_листа!Адрес_ячейки*. Можно сослаться на информацию, содержащуюся в другой рабочей книге. Такая ссылка называется внешней и имеет вид *[Книга5]Имя_листа!Адрес_ячейки*. Имя книги заключается в квадратные скобки. При использовании внешней ссылки должны быть открыты окна обеих используемых книг.

Пользователь имеет возможность удалить ячейку, выбрав *Главная* → *Ячейки* → *Удалить* или вставить ячейку при помощи команды *Вставить*. Те же команды дают возможность вставлять и удалять столбцы и строки.

При желании сетку можно сделать невидимой. Для этого надо выбрать *Вид* → *Показать или скрыть* → *Сетка* (убрать флажок).

Рабочая книга может содержать несколько рабочих листов. Пользователь имеет возможность добавлять листы с помощью кнопки *Вставить лист* в нижней части окна за последним листом.

§2. Выделение ячеек, редактирование информации в ячейках

Ввод и обработка информации возможны только в *выделенных ячейках*. Для того чтобы выделить ячейку, необходимо установить на неё курсор и щёлкнуть левой клавишей мыши. Адрес выделенной ячейки отображается в поле имени, расположенном в верхней части экрана непосредственно над таблицей слева. После ввода информации в ячейку необходимо либо нажать клавишу

«**Enter**», либо перевести курсор в соседнюю ячейку клавишами управления, что свидетельствует о завершении ввода информации. Если после этого требуется произвести редактирование данных в ячейке, не уничтожая их, ячейку выделяют и изменяют информацию в ней в соответствии с обычными правилами редактирования текстов. Допускается осуществление редактирования путём ввода информации в поле формул. Не следует забывать, что клавиши со стрелками используются для перемещения курсора из одной ячейки в другую. Поэтому для его перемещения внутри ячейки в процессе редактирования информации следует использовать мышь.

В некоторых случаях приходится выделять не одну, а целый набор ячеек. Если этот набор представляет собой столбец или строку, то для выделения достаточно щелкнуть мышью по соответствующему заголовку. Для выделения листа достаточно щелкнуть мышью по кнопке в левом верхнем углу экрана на пересечении осей координат. Если необходимо выделить произвольную прямоугольную область, то достаточно при нажатой кнопке мыши перевести курсор по диагонали из одного угла области в противоположный. В случае необходимости выделить ячейки, расположенные в различных частях экрана (выделение несвязанной области), надо щелкнуть мышью по каждой из них при нажатой клавише «**Ctrl**». Выделенная ячейка (область) имеет жирный черный контур с маленьким квадратом в нижнем правом углу. Этот квадрат называется *маркером заполнения* и играет существенную роль при управлении данными, помещёнными в ячейку.

§3. Копирование и перемещение информации, содержащейся в ячейках

Копирование и перемещение числовых и строковых данных. Функции управляющих клавиш, вид курсора мыши на листе рабочей книги Excel 2007 и изменение информации при перемещении курсора зависят от положения курсора относительно контура выделенной ячейки (см. рис. 3.1). К операциям, выполняемым с использованием левой клавиши мыши, относится генерация про-

стных и прогрессирующих последовательностей данных. Генерация *простой последовательности* — копирование информации, содержащейся в исходной ячейке или в выделенном блоке, во все последовательно выделяемые ячейки.

<i>1</i>	<i>1</i>	<i>1</i>	<i>1</i>	<i>1</i>
----------	----------	----------	----------	----------


Прогрессирующая последовательность, генерируемая при одновременном нажатии левой клавиши мыши и клавиши «*Ctrl*», имеет следующий вид:

<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
----------	----------	----------	----------	----------

причём исходный блок не должен включать более одной ячейки. Операции копирования и перемещения ячеек могут быть выполнены также с использованием контекстного меню, которое появляется в случае нажатия правой клавиши мыши. Закономерно могут изменяться при перемещении курсора не только числа, но и, например, даты, дни недели, названия месяцев и т.д.

Нечисловые последовательности, которые генерирует программа, образуют так называемые списки, хранящиеся в памяти машины (например, названия месяцев).

В справочной системе способ генерации простых и прогрессирующих последовательностей в ячейках таблицы с использованием левой клавиши мыши назван *автозаполнением* с использованием смежных ячеек.

Действие	Вид курсора	Результат
1. Указать любое место в ячейке		Выделить ячейку или область
2. Указать на границу или область выделенной ячейки	↓	Перенести ячейку или область из одного места в другое
3. Указать на границу выделенной ячейки или области при нажатой клавише « <i>Ctrl</i> »	⇓	Скопировать ячейку или область с помощью перетаскивания
4. Указать на маркер заполнения	+	Автоматически генерировать последовательности данных

5. Указать на маркер заполнения выделенной ячейки при нажатой клавише «Ctrl»	++	Автоматически генерировать прогрессирующие последовательности данных
---	----	---

Рис. 3.1

Аналогично осуществляются операции выбора, копирования и перемещения листов рабочей книги. Как и при работе с отдельными ячейками, перемещение листа осуществляется с использованием левой клавиши мыши, а копирование — при нажатой клавише «*Ctrl*». Если щелкнуть по двум ярлычкам листов при нажатой клавише «*Shift*», то выделенными окажутся все листы, расположенные между двумя отмеченными. При выполнении каждой из перечисленных процедур изменяется внешний вид курсора.

§4. Выполнение арифметических операций

Excel 2007 имеет мощный аппарат для проведения математических и прочих расчётов. Простейшие расчёты — арифметические действия над числами, помещёнными в соответствующие ячейки электронной таблицы. Для выполнения таких действий в ячейку, где должен находиться результат расчёта, записывают знак «*=*», свидетельствующий о характере выражения как математической формулы, и адреса ячеек, содержащих исходные данные, соединённые знаками арифметических действий: +, -, *, /, ^ (возведение в степень). Знаки арифметических действий называются *арифметическими операторами*. Наряду с арифметическими операторами программа даёт пользователю возможность применять логические операторы: «*=*» равенство, «*<*» меньше, «*>*» больше, «*<=*» меньше или равно, «*>=*» больше или равно, «*<>*» не равно, текстовый оператор «*&*» объединение последовательностей символов (амперсанд).

Действия программы после записи формул зависят от установок. Обычно вычисления производятся автоматически, и их результат вставляется в соответствующую ячейку. При желании

пользователь может заменить результаты формулами. Для этого надо выбрать **Формулы** → **Зависимости формул** → **Показать**

формулы  (рис. 3.2).

Перемещение и копирование формул. Абсолютная и относительная адресации ячеек. Одним из параметров, который может автоматически изменяться, является адрес ячейки, т.е. номер столбца и номер строки, на пересечении которых расположена ячейка. Изменяемые адреса ячеек называются **относительными**. При обычной записи адреса считаются относительными.

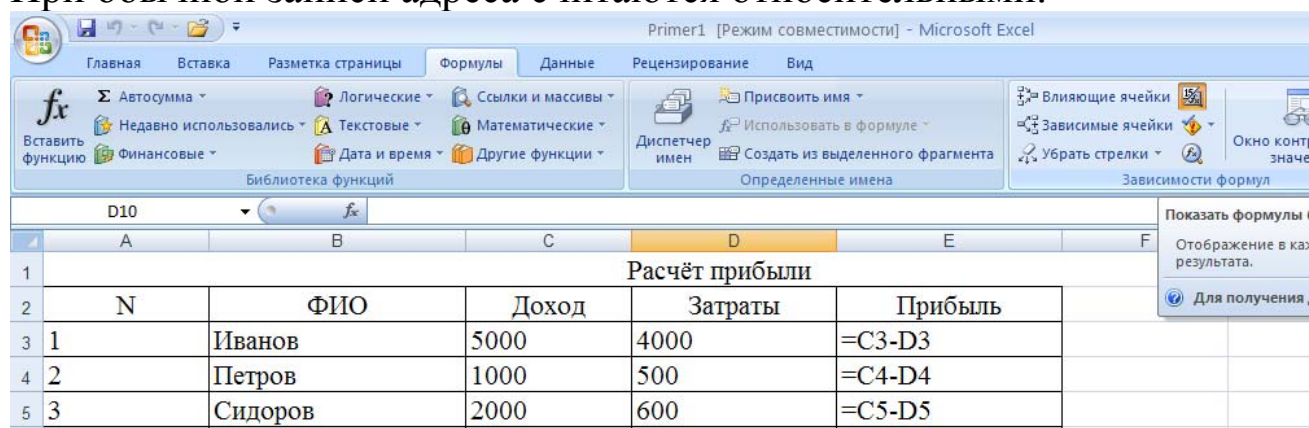


Рис. 3.2

	A	B	C	D	E
1	Расчёт прибыли				
2	N	ФИО	Доход	Затраты	Прибыль
3	1	Иванов	5000	4000	1000
4	2	Петров	1000	500	500
5	3	Сидоров	2000	600	1400
6	4	Макаров	3000	2000	1000
7	5	Фёдоров	2500	2100	400
8					

Рис. 3.3

Пример задачи с использованием относительных адресов ячеек и копированием формулы показан на рис. 3.3 и 3.4. Здесь формула =C3-D3 вносится в ячейку E3 и затем копируется в диапазон E4:E5.

	A	B	C	D	E
1	Расчёт прибыли				
2	N	ФИО	Доход	Затраты	Прибыль
3	1	Иванов	5000	4000	=C3-D3
4	2	Петров	1000	500	=C4-D4
5	3	Сидоров	2000	600	=C5-D5
6	4	Макаров	3000	2000	=C6-D6
7	5	Фёдоров	2500	2100	=C7-D7
8					

Рис. 3.4

Аналогичные манипуляции можно производить и со строками, столбцами и блоками.

Однако бывают случаи, когда адрес ячейки, например C1, при всех манипуляциях должен оставаться неизменным. Такой абсолютный адрес должен иметь знак \$ перед названием столбца и перед номером строки. Например, =A1*\$C\$1.

Пример использования абсолютных адресов ячеек показан на рис. 3.5 и 3.6.

Здесь выражение =C3*\$D\$19 вводится в ячейку F3 и затем копируется в диапазон F3:G16.

В ячейку D19 вводится значение курса доллара, которое используется при расчете розничной и оптовой цены в рублях. Чтобы при копировании формулы значение курса доллара не менялось, в формуле используется абсолютный адрес ячейки \$D\$19.

	A	B	C	D	E	F	G
1	Артикул	Название	Цена розн.	Цена опт.	Гарантия	Цена розн.	Цена опт.
2		Монитора	\$	\$		руб.	руб.
3	3127	Монитор Samsung 74V	240	230	3 года(Ф)	6120	5865
4	3285	Монитор Samsung SyncMaster 172X SDS Silver	380	355	3 года(Ф)	9690	9053
5	3141	Монитор Philips 170X5FB Blak	330	320	3 года(Ф)	8415	8160
6	3782	Монитор LG Flatron L1732S-SF Silver	240	230	3 года(Ф)	6120	5865
7	1749	Монитор LG Flatron L1770HQ-BF Silver	340	330	3 года(Ф)	8670	8415
8	2187	Монитор LG Flatron L1752T-SF Silver	280	265	3 года	7140	6758
9	301	Монитор Samsung SyncMaster 720N VKS	270	265	3 года(Ф)	6885	6758
10	3318	Монитор Samsung SyncMaster 720N AEB	295	285	3 года(Ф)	7523	7268
11	2782	Монитор ViewSonic VX724	320	305	2 года	8160	7778
12	4029	Монитор ViewSonic VA-702	260	250	2 года	6630	6375
13	2828	Монитор Samsung SyncMaster 720N AES	290	280	2 года	7395	7140
14	3928	Монитор Samsung SyncMaster 770P XHQ	395	380	2 года	10073	9690
15	3427	Монитор Nec 1770NX	340	330	3 года(Ф)	8670	8415
16	1721	Монитор ASUS PM17TU	320	305	1 год	8160	7778
17							
18							
19			Курс \$ =	25,5			

Рис. 3.5

	A	B	C	D	E	F	G
1	Артикул	Название	Цена розн.	Цена опт.	Гарантия	Цена розн.	Цена опт.
2		Монитора	\$	\$		руб.	руб.
3	3127	Монитор Samsung 74V	240	230	3 года(Ф)	=C3*\$D\$19	=D3*\$D\$19
4	3285	Монитор Samsung SyncMaster 172X SDS S	380	355	3 года(Ф)	=C4*\$D\$19	=D4*\$D\$19
5	3141	Монитор Philips 170X5FB Blak	330	320	3 года(Ф)	=C5*\$D\$19	=D5*\$D\$19
6	3782	Монитор LG Flatron L1732S-SF Silver	240	230	3 года(Ф)	=C6*\$D\$19	=D6*\$D\$19
7	1749	Монитор LG Flatron L1770HQ-BF Silver	340	330	3 года(Ф)	=C7*\$D\$19	=D7*\$D\$19
8	2187	Монитор LG Flatron L1752T-SF Silver	280	265	3 года	=C8*\$D\$19	=D8*\$D\$19
9	301	Монитор Samsung SyncMaster 720N VKS	270	265	3 года(Ф)	=C9*\$D\$19	=D9*\$D\$19
10	3318	Монитор Samsung SyncMaster 720N AEB	295	285	3 года(Ф)	=C10*\$D\$19	=D10*\$D\$19
11	2782	Монитор ViewSonic VX724	320	305	2 года	=C11*\$D\$19	=D11*\$D\$19
12	4029	Монитор ViewSonic VA-702	260	250	2 года	=C12*\$D\$19	=D12*\$D\$19
13	2828	Монитор Samsung SyncMaster 720N AES	290	280	2 года	=C13*\$D\$19	=D13*\$D\$19
14	3928	Монитор Samsung SyncMaster 770P XHQ	395	380	2 года	=C14*\$D\$19	=D14*\$D\$19
15	3427	Монитор Nec 1770NX	340	330	3 года(Ф)	=C15*\$D\$19	=D15*\$D\$19
16	1721	Монитор ASUS PM17TU	320	305	1 год	=C16*\$D\$19	=D16*\$D\$19
17							
18							
19			Курс \$ =	25.5			

Рис. 3.6

Задание

Самостоятельно скопируйте таблицу на Лист2 и Лист3. Переименуйте листы: Лист1 – на «Январь»; Лист2 – на «Февраль»; Лист3 — на «Март»; Лист4 — на «Квартал». Переименовать листы можно с помощью контекстно–зависимого меню командой **Переименовать**, наведя указатель мыши на вкладку Лист и нажав правую кнопку. В феврале и марте измените, курс доллара, цены мониторов в рублях изменятся автоматически. Назовите таблицы: «Цены мониторов за январь», «Цены мониторов за февраль», «Цены мониторов за март». На листе «Квартал» создайте новую таблицу, в которой укажите розничную цену мониторов в рублях за январь, февраль и март (цены возьмите с соответствующих листов, например =Январь! С3). В последнем столбце рассчитайте среднюю цену каждого монитора за квартал.

В случае необходимости одна из координат ячейки может быть абсолютной, а другая — относительной. Например, запись = $A2*B1$ свидетельствует об абсолютном характере номера столбца первой ячейки и номера строки второй ячейки. Остальные координаты обеих ячеек относительны. Такая форма записи адреса часто используется при создании таблиц для расчёта значений функций двух аргументов с использованием автозаполнения ячеек.

Пример нахождения значений таблицы умножения $z=x*y$ с использованием абсолютных адресов ячеек показан на рис. 3–7. Здесь формула = $A3*B2$ записывается в ячейку B3 и затем копируется в диапазон B3:J11. Таким образом, абсолютными делают адреса аргументов, которые в данном примере находятся в столбце A и в строке 2.

	A	B	C	D	E	F	G	H	I	J
1	Таблица умножения									
2		1	2	3	4	5	6	7	8	9
3	1	1	2	3	4	5	6	7	8	9
4	2	2	4	6	8	10	12	14	16	18
5	3	3	6	9	12	15	18	21	24	27
6	4	4	8	12	16	20	24	28	32	36
7	5	5	10	15	20	25	30	35	40	45
8	6	6	12	18	24	30	36	42	48	54
9	7	7	14	21	28	35	42	49	56	63
10	8	8	16	24	32	40	48	56	64	72
11	9	9	18	27	36	45	54	63	72	81

Рис. 3.7

Задание

Создайте таблицу значений функции $z = \sin^2(x) + \cos^2(y)$. Аргументы задайте в интервале от 0° до 90° с интервалом 10° .

Использование осмысленных имён как одна из форм абсолютной адресации

Excel дает возможность присваивать выделенным ячейкам или областям **осмысленные имена** и заменять ими координаты. Результат, может быть, достигнут с использованием вкладки **Формулы** → **Определенные имена** → **ввести имя** → **Ок**. Если осмысленные имена присвоены многим элементам таблицы, целесообразно вывести все названия и координаты соответствующих им ячеек на какой-либо свободный лист рабочей книги. Вывод перечня имён и вставка нужного имени может осуществляться при помощи команды **Формулы** → **Определенные имена** → **Присвоить имя** → **Применить имена**. После этого на экране появится диалоговое окно с перечнем имён. По имени можно обращаться к ячейкам, расположенным на любом из листов рабочей книги.

Можно также присваивать осмысленные имена листам рабочей книги вместо порядковых номеров. Для этого надо дважды щёлкнуть мышью по ярлычку листа в нижней части экрана, после чего в диалоговое окно вписать новое название листа и нажать «**Enter**». Другой способ переименования листа — щёлкнуть по ярлычку правой кнопкой мыши. Появится контекстное меню, содержащее перечень операций работы с листом (вставка, удале-

ние, переименование, выделение). В нём необходимо выбрать соответствующий пункт и ответить на последующие вопросы, которые будет автоматически задавать программа. Вторая процедура несколько сложнее, однако более универсальна, чем первая, ибо применима для выполнения каждой операции в контекстном меню. При выполнении расчётов осмысленные имена могут быть использованы наравне с координатами. Например, если ячейкам заранее присвоены имена, то возможна такая запись **=Цена*Количество**.

§5. Создание прогрессий

Пользователь имеет возможность производить автозаполнение смежных ячеек значениями, образующими **арифметическую** или **геометрическую** прогрессию. Для создания **прогрессии** следует:

1. Ввести значение первого члена прогрессии в первую ячейку области.
2. Выбрать на вкладке **Главная** команду **Прогрессия** (рис. 3.8).

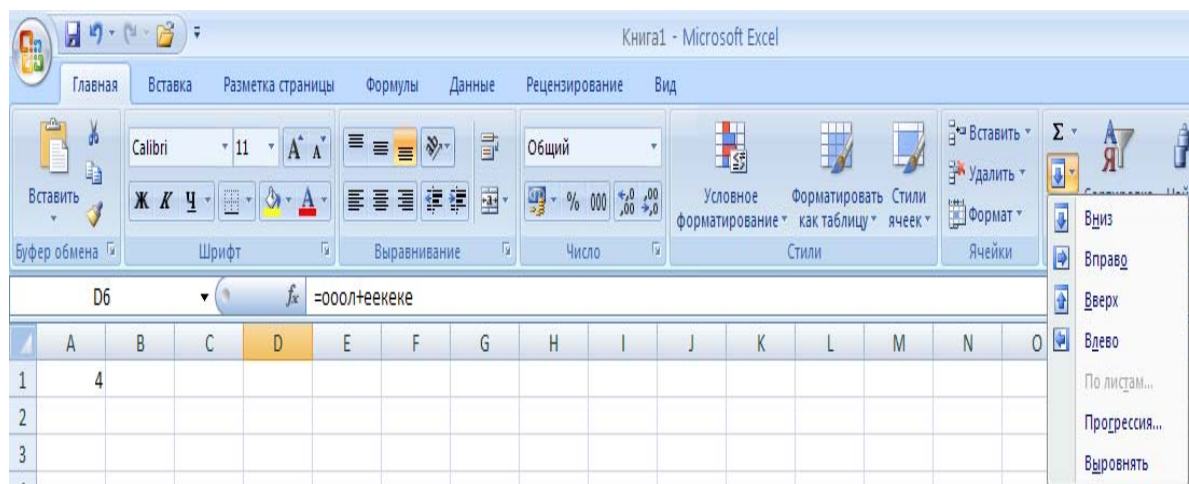


Рис. 3.8

3. В появившемся диалоговом окне (рис. 3.9) выбрать тип прогрессии (арифметическая, геометрическая, даты, автозаполнение). Указать, где располагать прогрессию в строках или в столбцах. Указать величину шага (знаменателя) прогрессии, установить предельное значение элемента прогрессии. Для последовательности дат выбрать единицу измерения (день, рабочий день,

месяц, год). В случае использования дат следует установить в ячейках таблицы соответствующий формат (см. ниже).

Формирование прогрессии прекратится либо после полного заполнения выделенной области, либо после достижения предельного значения.

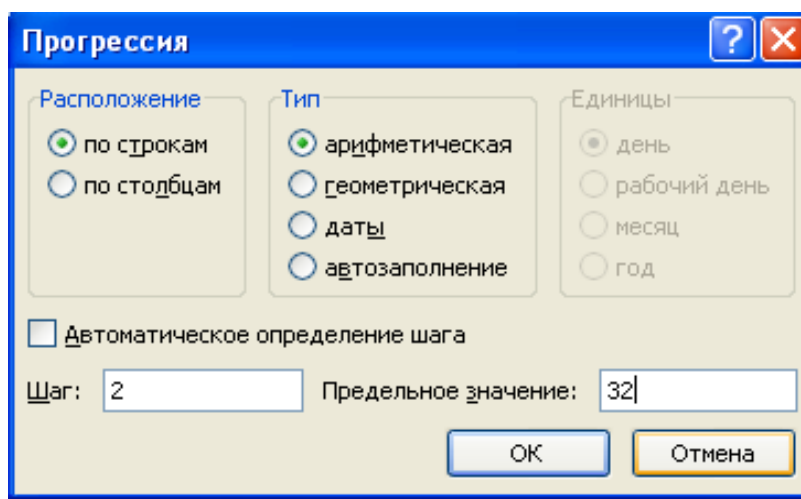


Рис. 3.9

Прогрессия может быть создана с использованием *правой клавиши мыши*. Для этого следует:

1. Ввести значение первого члена прогрессии в ячейку.
2. Установить курсор мыши на маркер заполнения этой ячейки.
3. Нажать правую клавишу мыши и протащить её через весь заполняемый диапазон ячеек.
4. Отпустить правую клавишу и в появившемся контекстном меню выбрать тип заполнения диапазона (например, *Прогрессия*).

§6. Форматы ячеек

Форматирование используется для придания ячейкам или тексту того или иного внешнего вида и для обработки требуемого типа информации. Наряду с методами форматирования, аналогичным методам, применяемым при работе с текстовым редактором Word 2007, программа Excel 2007 предоставляет пользователю ряд дополнительных возможностей. Так, наряду с форматированием текстовой или цифровой информации по центру или по краю ячейки, имеется возможность форматирования по центру

выделенной области. В выделенной области строки или столбца в контекстном меню выбирается **Формат ячеек**—> **Выравнивание**—>**Объединение ячеек**, и надпись располагается в середине области. Такое форматирование целесообразно применять для создания заголовков таблиц.

По умолчанию устанавливается **Общий** формат информации, т.е. информация, представляется в том виде, в каком её вводят с клавиатуры. Имеется возможность представлять числа в так называемом **денежном** формате, когда после числа ставится буква «р.» — рубли, и в **процентном** формате — число, набранное в числовом формате, автоматически умножается на 100 и после него ставится знак %. Вид десятичного разделителя определяется настройкой операционной системы. В русифицированных программах это, как правило, запятая. **Числовой** формат позволяет задать количество десятичных знаков. **Дробный** формат — представляет числа обыкновенными дробями. Причём из неправильной дроби будет автоматически выделяться целая часть. Отдельно следует рассмотреть представление дат. При преобразовании целого числа в дату числу ставится в соответствие количество дней, прошедших после 1.01.1900 года, и путём несложных вычислений определяется текущая дата. Обратное преобразование даёт возможность определить количество дней, прошедших с начала века до указанной даты. Такой сложный способ определения даты упрощает решение задачи вычисления продолжительности интервала времени между двумя событиями.

§7. Расчёты с использованием функций

Для выполнения сложных расчётов программа включает около сотни различных функций, некоторые из которых будут описаны ниже. Нужную функцию можно вызвать, набрав её имя и список аргументов в информационной строке, или выбрать из списка, который выводится на экран при помощи кнопки **Вставить функцию**.

Далее открывается диалоговое окно **Мастер функций** (рис.3.10), в котором перечислены категории функций (**Математические**, **Статистические**, **Финансовые** и т.д.), после вы-

бора, которых можно указать название функции, относящейся к выбранной категории.

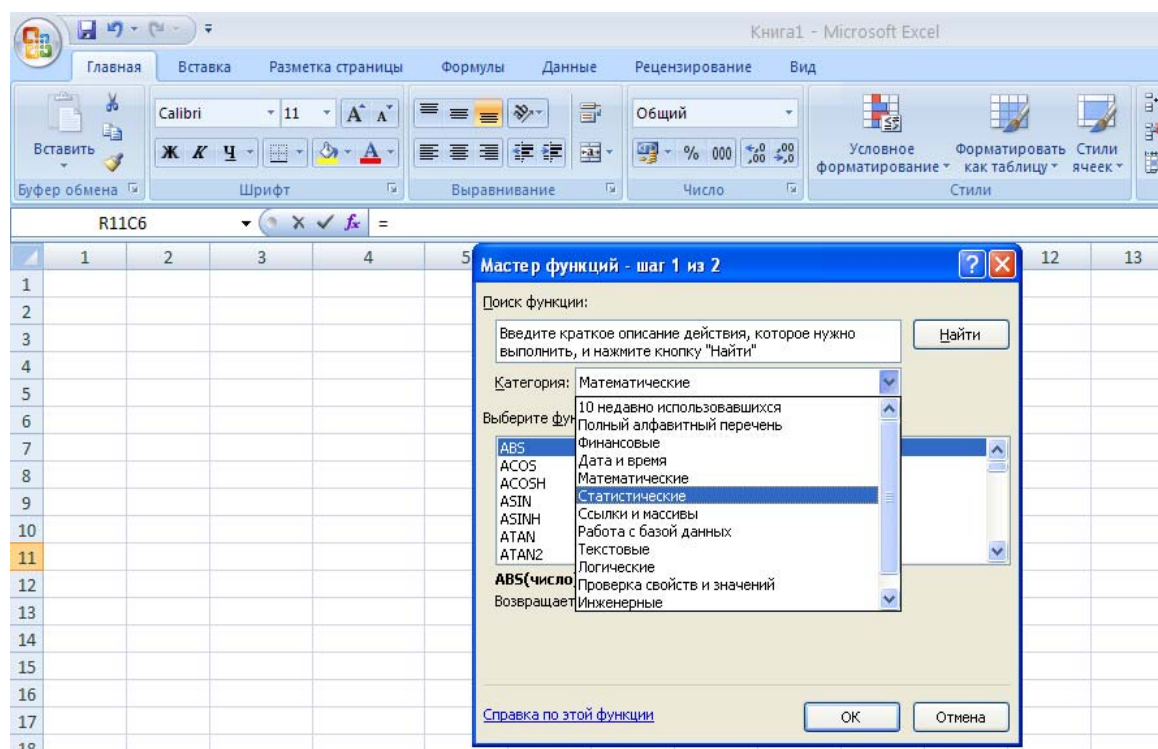


Рис. 3.10

Математические функции

Одна из наиболее часто используемых математических функций - суммирование. Вызов функции суммирования с использованием мастера функций сопровождается появлением окна, которое даёт возможность правильно ввести функцию и записать её аргументы (рис. 3.11). Аналогичные окна используются и при вызове других функций Excel. После ввода всех аргументов нажимают кнопку «*Ok*». Допускается вводить слагаемые, либо как числа, либо как адреса ячеек, в которых эти числа содержатся. На панели вкладки *Главная* имеется кнопка *автосуммирования*, помеченная знаком Σ , упрощающая процедуру вызова указанной функции. При автосуммировании аргументами функции *СУММ()* являются ячейки, расположенные либо сверху, либо слева от выделенной ячейки и содержащие числовые данные.

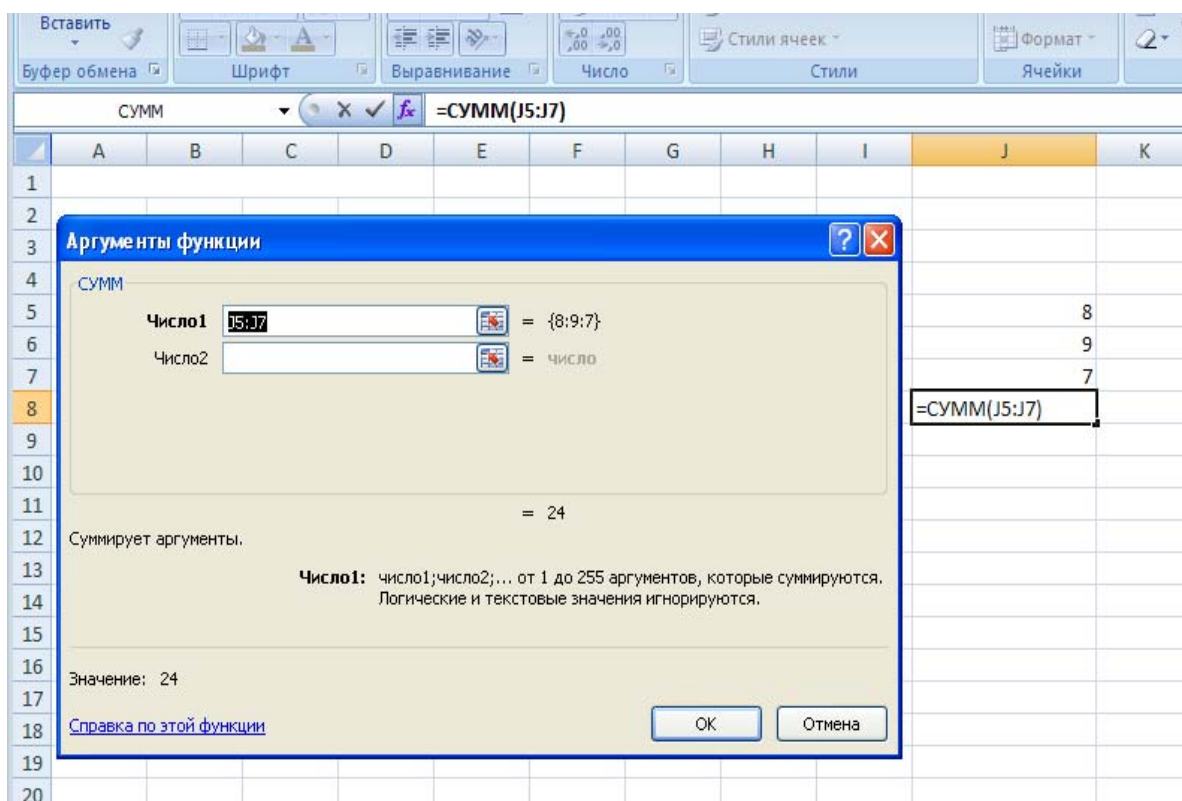


Рис. 3.11

Среди прочих математических функций чаще других используются:

СУММКВ(...) — сумма квадратов, аргументами, которой являются числа или адреса ячеек таблицы. Функция возвращает значение суммы квадратов аргументов.

СУММПРОИЗВ(...) — сумма произведений. Её аргументами являются массивы чисел или адресов ячеек таблицы. Функция СУММПРОИЗВ(...) возвращает сумму произведений элементов массивов — аргументов, имеющих одинаковые номера ($a_1b_1c_1+a_2b_2c_2+\dots$).

Расчёт величины определителя квадратной матрицы производится с использованием функции **МОПРЕД**. Аргументом является массив ячеек, в котором находятся элементы матрицы. Например, **МОПРЕД(A1:C3)**.

Среди математических функций следует отметить также округление, перевод градусной меры углов в радианную и радианной в градусную, тригонометрические функции и др., выполняемые в соответствии с обычными математическими формулами.

Пример с использованием функций представлен на рис. 3.12 и 3.13, где вычисляются расстояния от начала координат до точек замкнутого теодолитного хода, заданных координатами X и Y.

	A	B	C	D
1	Вычисление расстояний			
2	№	X	Y	Расстояние до т. 0,0
3	1	1,00	1,00	1,41
4	2	2,00	-3,00	3,61
5	3	5,00	5,00	7,07
6	4	30,00	20,00	36,06
7	5	25,00	2,00	25,08
8	6	10,00	3,00	10,44
9	7	15,00	10,00	18,03
10			Сумма расст.	101,69
11				
12			Сред. расст.	14,53

Рис. 3.12

	A	B	C	D
1	Вычисление расстояний			
2	№	X	Y	Расстояние до т. 0,0
3	1	1	1	=КОРЕНЬ(B3*B3+C3*C3)
4	2	2	-3	=КОРЕНЬ(B4*B4+C4*C4)
5	3	5	5	=КОРЕНЬ(B5*B5+C5*C5)
6	4	30	20	=КОРЕНЬ(B6*B6+C6*C6)
7	5	25	2	=КОРЕНЬ(B7*B7+C7*C7)
8	6	10	3	=КОРЕНЬ(B8*B8+C8*C8)
9	7	15	10	=КОРЕНЬ(B9*B9+C9*C9)
10			Сумма расст.	=СУММ(D3:D9)
11				
12			Сред. расст.	=СРЗНАЧ(D3:D9)

Рис. 3.13

Здесь формула вычисления расстояния вводится в ячейку D3, а затем копируется в ячейки D4:D9.

Задание

В следующем столбце Е (рис. 3.13) рассчитайте самостоятельно расстояние от i -ой точки (i – ваш номер в журнале) до каждой точки хода как корень из суммы квадратов разностей координат.

Вычислите расстояние между каждой парой смежных точек хода, длину теодолитного хода и площадь полигона внутри хода по формулам

$$L = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

$$2P = \sum_{k=1}^n x_k (y_{k+1} - y_{k-1})$$

Контроль расчета площади выполните по формуле:

$$2P = \sum_{k=1}^n y_k (x_{k-1} - x_{k+1}).$$

Постройте график полигона по координатам его вершин.

Функции даты и времени

Другая группа часто используемых функций — ***функции даты и времени***. Самая простая из этой группы функций — функция ***Сегодня()***, возвращающая дату, определяемую системными часами компьютера. Эта функция не имеет аргументов. Формат даты можно выбрать при помощи команды ***Формат*** → ***Ячейки***. Если в ячейке вместо даты появится бессмысленная комбинация символов, следует поменять формат или увеличить ширину ячейки. Другая аналогичная функция — ***ТДАТА()***. В отличие от ***Сегодня()***, ***ТДАТА()*** наряду с текущей датой возвращает и время. Имеются функции, возвращающие год, месяц, день недели и т. д. Для определения интервала между двумя датами достаточно из конечной даты вычесть начальную. Например, ***08.04.97–06.04.97*** → 2. Если даты поменять местами, ответ получится отрицательный. Допускается использование адресов ячеек, содержащих исходную информацию.

Логические функции

Наиболее распространённой логической функцией является функция **ЕСЛИ**. По своим возможностям она соответствует условному оператору в обычных алгоритмических языках. Форма записи этой функции следующая **ЕСЛИ(логическое_выражение; значение_если_истина; значение_если_ложь)**.

Используются также функции:

логическое сложение **ИЛИ(логическое_выражение1; логическое_выражение2; ..; логическое_выражениеN)**;

логическое умножение **И(логическое_выражение1; логическое_выражение2; ...; логическое_выражениеN)**;

отрицание **НЕ(логическое_выражение)**.

Ниже приведен пример с использованием логической функции **ЕСЛИ**, в котором отбираются лица, достигшие 18 лет и не достигшие 18 лет (рис. 3.14, 3.15).

Задание

Столбцы «От 20 до 30» и «От 30 до 40» рассчитайте самостоятельно. Выведите фамилии в случае попадания в диапазон и выведите пробел в случае не попадания в него. Используйте логическую функцию **И** для определения попадания в диапазон. Например, **И(E4>=20;E4<30)**.

	A	B	C	D	E	F	G	H
1								
2		Возраст						
3	№	ФИО	Дата рождения	Возраст	Достигшие 18 лет	Не достигшие 18 лет	От 20 до 30	От 30 до 40
4	1	Иванов	02.06.1990	17,3		Иванов		
5	2	Петров	13.12.1977	29,8	Петров			
6	3	Сидоров	02.02.1978	29,7	Сидоров			
7	4	Иваненко	13.12.1980	26,8	Иваненко			
8	5	Демченко	14.01.1975	32,7	Демченко			
9	6	Сидорова	11.08.1974	33,2	Сидорова			
10	7	Демиденко	02.09.1980	27,1	Демиденко			
11	8	Лохушкина	01.01.1981	26,8	Лохушкина			
12								
13		Текущая дата	01.10.2007					
14								

Рис. 3.14

	A	B	C	D	E	F	G	H
1								
2		Возраст						
3	№	ФИО	Дата рождения	Возраст	Достигшие 18 лет	Не достигшие 18 лет	От 20 до 30	От 30 до 40
4	1	Иванов	33026	=(\$C\$13-C4)/365	=ЕСЛИ(D4>=18;B4;"")	=ЕСЛИ(D4<18;B4;"")		
5	2	Петров	28472	=(\$C\$13-C5)/365	=ЕСЛИ(D5>=18;B5;"")	=ЕСЛИ(D5<18;B5;"")		
6	3	Сидоров	28523	=(\$C\$13-C6)/365	=ЕСЛИ(D6>=18;B6;"")	=ЕСЛИ(D6<18;B6;"")		
7	4	Иваненко	29568	=(\$C\$13-C7)/365	=ЕСЛИ(D7>=18;B7;"")	=ЕСЛИ(D7<18;B7;"")		
8	5	Демченко	27408	=(\$C\$13-C8)/365	=ЕСЛИ(D8>=18;B8;"")	=ЕСЛИ(D8<18;B8;"")		
9	6	Сидорова	27252	=(\$C\$13-C9)/365	=ЕСЛИ(D9>=18;B9;"")	=ЕСЛИ(D9<18;B9;"")		
10	7	Демиденко	29466	=(\$C\$13-C10)/365	=ЕСЛИ(D10>=18;B10;"")	=ЕСЛИ(D10<18;B10;"")		
11	8	Лохушкина	29587	=(\$C\$13-C11)/365	=ЕСЛИ(D11>=18;B11;"")	=ЕСЛИ(D11<18;B11;"")		
12								
13		Текущая дата	=СЕГОДНЯ()					
14								

Рис. 3.15

Допускается использование вложенных логических функций.

Пусть заданы рост и вес человека. В качестве нормального примем вес, равный 0,9 от роста в сантиметрах минус 100. При весе, отличающемся от нормы не более чем на 2 кг, будем выдавать сообщение «Норма», при весе, отличающемся от нормы более чем на 20 кг, – «Обратиться к врачу», в остальных случаях при весе меньшем нормы — “Надо поправиться”, при весе большем нормы — «Надо похудеть».

Соответствующий пример приведен на рис. 3.16 и 3.17.

Здесь формулы набираются в ячейках E3 и F3, а затем копируются в диапазон E4:F7.

	A	B	C	D	E	F
1	Контроль веса					
2	№	ФИО	Рост	Вес	Норма	Рекомендации
3	1	Иванов	175	90	67,5	обратиться к врачу
4	2	Петров	180	76	72	надо похудеть
5	3	Сидоров	170	61	63	норма
6	4	Иваненко	178	102	70,2	обратиться к врачу
7	5	Демченко	170	50	63	надо поправиться

Рис. 3.16

	A	B	C	D	E	F
1		Контроль веса				
2	№	ФИО	Рост	Вес	Норма веса	Рекомендации
3	1	Иванов	175	90	=0,9*(C3-100)	=ЕСЛИ(И(D3<=(E3+2);D3>=(E3-2)),"норма";ЕСЛИ(ABS(D3-E3)>20;"обратиться к врачу";ЕСЛИ(D3>E3;"надо похудеть";"надо поправиться"))
4	2	Петров	180	76	=0,9*(C4-100)	=ЕСЛИ(И(D4<=(E4+2);D4>=(E4-2));"
5	3	Сидоров	170	61	=0,9*(C5-100)	=ЕСЛИ(И(D5<=(E5+2);D5>=(E5-2));"
6	4	Иваненко	178	102	=0,9*(C6-100)	=ЕСЛИ(И(D6<=(E6+2);D6>=(E6-2));"
7	5	Демченко	170	50	=0,9*(C7-100)	=ЕСЛИ(И(D7<=(E7+2);D7>=(E7-2));"
8	6					

Рис. 3.17

Задание

1. Используя вложенные логические функции, добавить в таблицу на рис. 3.14 столбец, в котором люди, не достигшие 18 лет, должны характеризоваться как «дети», люди от 18 до 20 лет – как «юноши», от 20 до 30 лет - как «молодые люди» и от 30 лет и выше - как «взрослые». Добавить в табл. на рис. 3.14 ряд записей так, чтобы в каждую категорию попал хотя бы один человек.

2. Добавить в табл. 3.16 столбец «пол», содержащий указания М - мужчины и Ж – женщины. Расчет нормального веса выполнить с учетом пола: если М - $0,9(\text{Рост} - 100)$, иначе $0,8(\text{Рост} - 100)$. Рекомендации тоже выполнить с учетом пола: Если М, в соответствии с таблицей 3.17, иначе «норма» это норма±4, «обращение к врачу» это $\text{ABS}(\text{вес-норма}) > 25$.

В качестве еще одного примера рассмотрим расчёт суммы налога на стоимость имущества, которое переходит в порядке наследования или дарения.

Ставки налога на имущество, переходящее в порядке наследования, в зависимости от его стоимости и степени родства наследника приведены на рис. 3.18.

Структура таблицы расчёта налога приведена на рис. 3.19 и 3.20.

	Наследование	Дарение
0) Стоимость имущества менее 850 МРОТ		
Вне зависимости от степени родства	Ставка налога равна 0%	<i>От 80 до 850 — МРОТ</i> Детям, родителям — 3 % стоимости им-ва, превыш. 80 МРОТ Другим физическим лицам — 10 % ст-сти им-ва, превыш. 80 МРОТ
1) Стоимость имущества от 850 до 1700 МРОТ		
Наследники 1-й очереди(дети, родители)	5% стоимости им-ва, превышающей 850 МРОТ	23,1МРОТ + 7% стоимости имущества, превышающей 850 МРОТ
Наследники 2-й очереди(дед, бабушка, братья, сёстры)	10% стоимости имущества, превышающей 850 МРОТ	77 МРОТ + 20 % стоимости имущества, превышающей 850 МРОТ
Другие наследники	15%стоимости им-ва, превышающей 850 МРОТ	
2) Стоимость имущества от 1700 до 2550 МРОТ		
Наследники 1-й очереди	42,5 МРОТ +10% стоимости им-ва, превышающей 1700 МРОТ	82,6 МРОТ + 11% стоимости имущества, превышающей 1700 МРОТ
Наследники 2-й очереди	85 МРОТ +20% стоимости имущества, превышающей 1700 МРОТ	247 МРОТ + 30% стоимости имущества, превышающей 850 МРОТ
Другие наследники	170 МРОТ +30% стоимости имущества, превышающей 1700 МРОТ	
3) Стоимость имущества превышает 2550 МРОТ		
Наследники 1-й очереди	127,5 МРОТ +15% стоимости имущества, превышающей 2550 МРОТ	176,1 МРОТ + 15% стоимости имущества, превышающей 2550 МРОТ

Наследники 2-й очереди	255 МРОТ +30% стоимости имущества, превышающей 2550 МРОТ	502 МРОТ + 40 % стоимости имущества, превышающей 2550 МРОТ
Другие наследники	425 МРОТ +40% стоимости имущества, превышающего 2550 МРОТ	

Рис. 3.18

	A	B	C	D	E	F	G	H	I
1	Налог на наследство								
2	Стоимость имущества		1 очередь	2 очередь		Другие наследники			
3	МРОТ	Руб.	%	налог	%	налог	%	налог	
4	1	100							
5	850	85000	5	0	10	0	15	0	
6	1700	170000	10	4250	20	8500	30	12750	
7	2250	225000	15	9750	30	19500	40	29250	
8									
9	наследство	150000		3250		6500		9750	
10	очередь	1							
11	налог=	3250							
12									
13									
14									

Рис. 3.19

Здесь столбцы А, С, Е и G заполняются в соответствии с данными таблицы на рис. 3.18. В ячейку В4 вводится действующая величина МРОТ (в сентябре 2007 г. 2300 руб.). Для перевода стоимости имущества из МРОТ в рубли в ячейку В5 вводится формула =A5*B\$4 и копируется в ячейки В6 и В7. Налог на первое граничное значение (850 МРОТ) равен 0 (D5).

Для вычисления величины налога на второе граничное значение стоимости имущества (1700 МРОТ) для наследников первой очереди в ячейку D6 вводится формула =D5+(\$B6-\$B5)*C5/100. Для вычисления величины налога на следующее граничное значение стоимости имущества (2250 МРОТ) для наследников первой очереди эта формула копируется из ячейки D6 в ячейку D7. Для вычисления соответствующих граничных величин налога для наследников второй очереди и других наследни-

ков содержимое ячеек D6:D7 копируется последовательно в F6:F7 и H6:H7.

D9	=	=ЕСЛИ(\$B9<=\$B5;0;ЕСЛИ(\$B9<=\$B6;D5+(\$B9-\$B5)*C5/100;ЕСЛИ(\$B9<=\$B7;D6+(\$B9-\$B6)*C6/100;D7+						
	A	B	C	D	E	F	G	H
1	Налог на наследство							
2	Стоимость имущества		1 очередь		2 очередь		Другие наследники	
3	ММОТ	Руб.	%	налог	%	налог	%	налог
4	1	100						
5	850	=A5*B\$4	5	0	10	0	15	0
6	1700	=A6*B\$4	10	=D5+(\$B6-\$B5)*C5/100	20	=F5+(\$B6-\$B5)*E5/100	30	=H5+(\$B6-\$B5)*G5/100
7	2250	=A7*B\$4	15	=D6+(\$B7-\$B6)*C6/100	30	=F6+(\$B7-\$B6)*E6/100	40	=H6+(\$B7-\$B6)*G6/100
8								
9	наследство	150000		=ЕСЛИ(\$B9<=\$B5;0;ЕСЛИ(\$B9<=\$B6;D5+(\$B9-\$B5)*C5/100;ЕСЛИ(\$B9<=\$B7;D6+(\$B9-\$B6)*C6/100;D7+(\$B9-\$B7)*C7/100)))		=ЕСЛИ(\$B9<=\$B5;0;ЕСЛИ(\$B9<=\$B6;F5+(\$B9-\$B5)*E5/100;ЕСЛИ(\$B9<=\$B7;F6+(\$B9-\$B6)*E6/100;F7+(\$B9-\$B7)*E7/100)))		=ЕСЛИ(\$B9<=\$B5;0;ЕСЛИ(\$B9<=\$B6;H5+(\$B9-\$B5)*G5/100;ЕСЛИ(\$B9<=\$B7;H6+(\$B9-\$B6)*G6/100;H7+(\$B9-\$B7)*G7/100)))
10	очередь	1						
11	налог=	=ЕСЛИ(B10=1;D9;ЕСЛИ(B10=2;F9;H9))						
12								
13								

Рис. 3.20

Для вычисления налога на наследство для наследников первой очереди с любой конкретной стоимости имущества, помещаемой в ячейку B9, в D9 вводится соответствующая формула (см. таблицу на рис. 3–20).

Для вычисления налога для наследников второй очереди и других наследников формула из ячейки D9 копируется последовательно в ячейки F9 и H9. Формула для получения налога на наследство в зависимости от номера очереди наследников вводится в ячейку B11.

Для контроля правильности расчетов полезно ввести в B9 стоимость имущества, соответствующую второй (1700 МРОТ) либо третьей (2250 МРОТ) границам. При этом величина налога, вычисленная по общей формуле, должна равняться величине налога, вычисленного для соответствующего граничного значения. Например, для стоимости имущества, равной 2250 МРОТ, должны совпадать налоги, определенные в ячейках D9 и D7, F9 и F7, H9 и H7.

Рассмотрим ещё один пример использования вложенных логических функций — расчёт заработной платы. Структура таблицы представлена на рис. 3–21.

Столбец «Сумма1» в таблице вычисляется по формуле:

Сумма1=Оклад/25.6*Рабочие дни.

Сумма2=Сумма1+Другие начисления.

Удержание в пенсионный фонд составляет 1% от «Суммы 2».

Для расчёта льгот на подоходный налог используются данные из таблицы учёта льгот, количество детей и МРОТ.

Общий вид формулы вычисления льгот:

$L = \text{если}(\text{Сумма2} - \text{ПФ} + \text{СД} \leq 15000; 2 * \text{ММОТ} + 2 * \text{Д} * \text{МРОТ}; \text{если}(\text{Сумма2} - \text{ПФ} + \text{СД} \leq 50000; \text{ММОТ} + \text{Д} * \text{МРОТ}; 0))$,

где

ПФ – налог в пенсионный фонд,

СД – совокупный доход за предыдущий период.

Формула вычисления льгот в табл. на рис. 3.22:

=ЕСЛИ((F3-G3+J2) <= 15000;2*F\$19+2*F\$20*F\$19; ЕСЛИ (F3-G3+J2<=50000;F\$19*1+F\$19*F\$20*1;0))

Облагаемая сумма = Сумма2 - Пенсионный фонд - Льготы.

Совокупный доход = Совок. Доход за предыдущий период + Обл.Сумма.

	A	B	C	D	E	F	G	H	I	J	K	L
	Месяцы	Оклад	Рабочие дни	Сумма 1	Другие начисления	Сумма2	Пенсионный фонд	Льготы	Облагаемая сумма	Сов. доход	Подходный налог	К выдаче
2												
3	январь	2000	23	1797	5000	6797	68	334	6395	6395	767	6729
4	февраль	2000	20	1563	10000	11563	116	167	11280	17675	1354	10093
5	март	3000	22	2578	10000	12578	126	167	12285	29960	1474	10978
6	апрель	3000	21	2461	10000	12461	125	167	12169	42130	1460	10876
7	май	3000	23	2695	20000	22695	227	0	22468	64598	3864	18604
8	июнь	4000	21	3281	50000	53281	533	0	52748	117346	10550	42199
9	июль	4000	22	3438	25000	28438	284	0	28153	145499	5631	22523
10	август	4000	20	3125	50000	53125	531	0	52594	198093	15328	37266
11	сентябрь	4000	24	3750	10000	13750	138	0	13613	211706	4084	9529
12	октябрь	4000	19	2969	10000	12969	130	0	12839	224545	3852	8987
13	ноябрь	4000	22	3438	10000	13438	134	0	13303	237848	3991	9312
14	декабрь	4000	20	3125	10000	13125	131	0	12994	250842	3898	9096
15												
16						254219	2542	835	250842		56252	
17									250842		56252	
18												
19				ммот =		83,49						
20				кол-во детей =		1		3864			56253	
21												
22	Шкала подоходного налога						Таблица учёта льгот					
23												
24	Считать от суммы	Сумма налога	Процент налога				Считать суммы	Коеф-т льгот				
25							0	2				
26												
27	0	0	12				15000	1				
28	50000	6000	20				50000	0				
29	150000	26000	30									

Рис. 3.21

В методических целях подоходный налог рассчитывается на основе шкалы подоходного налога (рис. 3.22), совокупного дохода и облагаемой суммы. (После 2002 г. реальный подоходный налог составляет 13% от облагаемой суммы).

	A	B	C	D	E	F	G	H	I	J	K	L
2	Месяцы	Оклад	Рабочие дни	Сумма1	Другие начисления	Сумма2	Пенсионный фонд	Льготы	Облагаемая сумма	Сов. доход	Подходный налог	К выдаче
3	январь	2000	23	= (B3/25,6)*C3	5000	=D3+E3	=F3*0,01	=ЕСЛИ(F3-G3<G\$27;H	=F3-G3-H3	=I3	=ЕСЛИ(J3+I3	=F3-(G3+C17)
4	февраль	2000	20	= (B4/25,6)*C4	10000	=D4+E4	=F4*0,01	=ЕСЛИ(F4+J3-G3<G\$	=F4-G4-H4	=J3+H4	=ЕСЛИ(J3+H4	=F4-(G4+K4)
5	март	3000	22	= (B5/25,6)*C5	10000	=D5+E5	=F5*0,01	=ЕСЛИ(F5+J4-G4<G\$	=F5-G5-H5	=J4+H5	=ЕСЛИ(J4+H5	=F5-(G5+K5)
6	апрель	3000	21	= (B6/25,6)*C6	10000	=D6+E6	=F6*0,01	=ЕСЛИ(F6+J5-G5<G\$	=F6-G6-H6	=J5+H6	=ЕСЛИ(J5+H6	=F6-(G6+K6)
7	май	3000	23	= (B7/25,6)*C7	20000	=D7+E7	=F7*0,01	=ЕСЛИ(F7+J6-G6<G\$	=F7-G7-H7	=J6+H7	=ЕСЛИ(J6+H7	=F7-(G7+K7)
8	июнь	4000	21	= (B8/25,6)*C8	50000	=D8+E8	=F8*0,01	=ЕСЛИ(F8+J7-G7<G\$	=F8-G8-H8	=J7+H8	=ЕСЛИ(J7+H8	=F8-(G8+K8)
9	июль	4000	22	= (B9/25,6)*C9	25000	=D9+E9	=F9*0,01	=ЕСЛИ(F9+J8-G8<G\$	=F9-G9-H9	=J8+H9	=ЕСЛИ(J8+H9	=F9-(G9+K9)
10	август	4000	20	= (B10/25,6)*C10	50000	=D10+E10	=F10*0,01	=ЕСЛИ(F10+J9-G9<G	=F10-G10-H10	=J9+H10	=ЕСЛИ(J9+H10	=F10-(G10+K
11	сентябрь	4000	24	= (B11/25,6)*C11	10000	=D11+E11	=F11*0,01	=ЕСЛИ(F11+J10-G10<	=F11-G11-H11	=J10+H11	=ЕСЛИ(J10+H11	=F11-(G11+K
12	октябрь	4000	19	= (B12/25,6)*C12	10000	=D12+E12	=F12*0,01	=ЕСЛИ(F12+J11-G11<	=F12-G12-H12	=J11+H12	=ЕСЛИ(J11+H12	=F12-(G12+K
13	ноябрь	4000	22	= (B13/25,6)*C13	10000	=D13+E13	=F13*0,01	=ЕСЛИ(F13+J12-G12<	=F13-G13-H13	=J12+H13	=ЕСЛИ(J12+H13	=F13-(G13+K
14	декабрь	4000	20	= (B14/25,6)*C14	10000	=D14+E14	=F14*0,01	=ЕСЛИ(F14+J13-G13<	=F14-G14-H14	=J13+H14	=ЕСЛИ(J13+H14	=F14-(G14+K
15												
16						=СУММ(F3:F	=СУММ(G3:G	=СУММ(H3:H6)	=СУММ(I3:I14		=ЕСЛИ(J16+H1	
17									=F16-G16-H16		=СУММ(K3:K	
18												
19				ммот =		83,49						
20				кол-во детей =		1		=14598*0,2+7870*0,12			= (250842-1500	
21												
22	Шкала подоходного налога						Таблица учёта льгот					
23												
24	Считать суммы	Сумма	Процент				Считать от суммы	Коеф-т льгот				
25	0	0	12				15000	1				
26	50000	=(A28-20					50000	0				
27	150000	=(A29-30										

Рис. 3.22

В данном примере ставка ПН и ОС меняется при $СД+ОС > 50000$ рублей и при $СД+ОС > 150000$ рублей. При этом возможны следующие шесть вариантов исчисления ПН.

Если $СД$ (за предыдущий период) + $ОС \leq 50000$ руб., ПН составляет 12% от $ОС$.

Если $СД$ больше 150000 руб., ПН составляет 30% от $ОС$.

Если $СД > 50000$ и $СД+ОС \leq 150000$, ПН составляет 20% от $ОС$.

Если $СД \leq 50000$ и $СД+ОС \leq 150000$, то ПН с части $СД+ОС$, превышающей 50000, берётся по ставке 20%, а с остальной части $ОС$ – по ставке 12%.

Если $СД > 50000$ и $СД+ОС > 150000$, то ПН с части $СД+ОС$, превышающей 150000, берётся по ставке 30%, а с остальной части $ОС$ – по ставке 20%.

Наконец, в последнем из возможных случаев (его можно уже формально не записывать), если $СД < 50000$ и $СД+ОС > 150000$, то ПН с части $ОС$, где $СД+ОС$ больше 150000, берётся по ставке

30%, ПН с части от 50 до 150000 берётся по ставке 20%, а с остальной части ОС по ставке 12%.

Формула расчёта подоходного налога в табл. на рис. 3–22 :
 =ЕСЛИ(J2+I3<=50000;I3*0,12;ЕСЛИ(J2>=150000;I3*0,3;ЕСЛИ(И(J2>50000;(J2+I3)<=150000);I3*0,2;ЕСЛИ(И(J2<=50000;(J2+I3)<=150000);(J2+I3-50000)*0,2+(I3-(J2+I3-50000))*0,12;ЕСЛИ(И(J2>50000;(J2+I3)>150000);(J2+I3-150000)*0,3+(I3-(J2+I3-150000))*0,2;(J2+I3-150000)*0,3+100000*0,2+(I3-(J2+I3-150000)-100000)*0,12))))).

Формулы расчета показаны в табл. на рис. 3.22.

Контролем правильности расчетов в табл. на рис. 3.21 служит равенство суммы подоходных налогов по месяцам (К17) и подоходного налога на сумму облагаемых сумм (К16).

§8. Формулы массива

Формулы массива используются, когда необходимо произвести над данными какую-либо операцию, результатом которой является массив ячеек.

А В С D E

Модель	Цена 2002	Цена2003	Отношение	%
Pentium4	500	450	=(B2:B5-C2:C5)/B2:B5	=D2
AMD	450	400	=(B2:B5-C2:C5)/B2:B5	=D3
Celeron	350	300	=(B2:B5-C2:C5)/B2:B5	=D4
AMD Athlon	500	440	=(B2:B5-C2:C5)/B2:B5	=D5

Рис. 3.23

В таблице на рис. 3.23 проиллюстрирована динамика цен на ПК различных моделей. Во 2–м и 3–м столбцах приведены цены 2002–го и 2003–го годов, а в последних двух столбцах — изменение цен в абсолютных величинах и в процентах. Во всех ячейках столбца D производятся одинаковые арифметические операции. При этом можно не вписывать формулу в каждую ячейку, а выделить **D2:D5** и записать формулу один раз в активную ячейку.

Такая формула называется *формулой массива*. После набора формулы массива надо нажать одновременно 3 клавиши «*Ctrl*»–«*Shift*»–«*Enter*». Тогда будут выполнены преобразования данных всех ячеек массива. То же в столбце Е. В информационной строке формулы массива заключены в *фигурные* скобки. *Изменение части результирующего массива недопустимо*.

В простейшем случае, проиллюстрированном выше, формула массива решает практически ту же задачу, что и автозаполнение ячеек (копирование). Однако, например, в матричных операциях формулы массива незаменимы.

Умножение матриц

Для выполнения этой операции необходимо

1. Ввести в соответствующие поля массивы ячеек, содержащих матрицы–сомножители.
2. Выделить односвязанную область, в которой должен располагаться результат.
3. Вызвать (или вписать в информационную строку) функцию **МУМНОЖ**.
4. Одновременно нажать клавиши «*Ctrl*»–«*Shift*»–«*Enter*».

Если просто нажать кнопку **Готово** в окне диалога вызова функции, то на экран будет выведен только первый элемент матрицы–произведения.

Аналогичные действия следует выполнить для определения элементов матрицы, *обратной* заданной, с использованием функции **МОБР**.

В качестве примера рассмотрим решение системы линейных уравнений методом обратной матрицы:

$$\begin{aligned} \mathbf{Ax} &= \mathbf{b} \\ \mathbf{x} &= \mathbf{A}^{-1}\mathbf{b}. \end{aligned}$$

Пусть элементы матрицы **A** (коэффициенты системы уравнений) записаны в диапазоне **A1:D4**, а элементы вектора **B** (свободные члены уравнений) — в диапазоне **F1:F4** (рис. 3.24).

Для получения значений неизвестных x_1, \dots, x_4 выделим диапазон **H1:H4**. Наберем в активной ячейке **H1** формулу

$$= \text{МУМНОЖ}(\text{МОБР}(\text{A1:D4}); \text{F1:F4})$$

и введем ее одновременным нажатием «*Ctrl*»–«*Shift*»–«*Enter*». В ячейках **H1:H4** получим величины x_1, \dots, x_4 .

	A	B	C	D	E	F	G	H	I
1	1	2	8	1		1		-6,2	
2	1	2	5	1		2		3,3	
3	4	7	1	1		1		-0,3	
4	5	1	1	9		2		3,4	
5									

Рис. 3.24

Задание

1. Решите систему линейных уравнений 4-го порядка, где коэффициенты при неизвестных по главной диагонали равны номеру студента по журналу.

Проверьте правильность решения, вычислив исходные свободные члены по формуле

$$\mathbf{b}=\mathbf{A}\mathbf{x}.$$

Убедитесь, что произведение $\mathbf{A}\cdot\mathbf{A}^{-1}$ даёт единичную матрицу \mathbf{E} .

2. Получите решение той же системы уравнений по формулам Крамера:

$$x_j = \frac{\Delta_j}{\Delta},$$

где Δ - определитель системы уравнений, составленный из коэффициентов при неизвестных, а Δ_j - j -тый дополнительный минор, т.е. определитель системы, где j -тый столбец заменен свободными членами.

Порядок решения:

- Ввести матрицу коэффициентов (A1:D4) и свободные члены (F1:F4) системы уравнений.
- Вычислить определитель матрицы в G7 по формуле =МОПРЕД(A1:D4).
- Скопировать матрицу коэффициентов в (A7:D10).

- Заменить первый столбец (A7:A10) свободными членами системы (F1:F4) и вычислить первый дополнительный минор в ячейке G8.
- Вычислить $x_1 = \frac{\Delta_1}{\Delta}$, в ячейке G9.
- Скопировать ячейку G9 и вставить (специальная вставка, значение) в ячейку J1.
- Восстановить первый столбец матрицы коэффициентов (A7:A10) по исходным значениям (A1:A4).
- Заменить следующий столбец коэффициентов (B7:B10) свободными членами системы (F1:F4). При этом в G8 и G9 появятся значения, соответствующие D2 и X2.
- Скопировать x_2 и вставить его в J2.
- Аналогично получить в G9 и вставить в J2-J4 остальные неизвестные.

§9. Индикация ошибок при использовании формул

В случае неправильного использования функций Excel сообщает пользователю о допущенной ошибке. Перед сообщением об ошибке всегда присутствует знак # (диез). Смысл наиболее часто встречающихся сообщений приведен на рис. 3.25.

Сообщение	Ошибка
Последовательность знаков диез	Ширина ячейки недостаточна для размещения результата
#ССЫЛКА	В формуле задана ссылка на несуществующие ячейки
#ДЕЛ/0	Деление на нуль
#ЧИСЛО!	Нарушение правил работы с оператором, например, отрицательная величина под корнем
#ИМЯ?	Неправильно задано имя функции

#ПУСТО!	Неверно указаны диапазоны. Они не имеют общих ячеек
#Н/Д!	Ячейка — аргумент функции, не содержит данных
#ЗНАЧ!	Задан аргумент недопустимого типа

Рис. 3.25

§10. Решение уравнений с одним неизвестным с использованием функций Excel 2007

В экономической работе и других областях деятельности часто приходится сталкиваться с задачами по вычислению какого-либо параметра математического выражения. Рассмотрим несколько примеров.

Уровень инфляции 10% в год. Чему равен ежемесячный рост цен? Для определения увеличения цен за месяц необходимо решить уравнение $(1+X)^{12}=1,10$.

В ячейку **A1** рабочего листа Excel вносим какое-либо число, которое, по нашему мнению, может быть решением, например **1**. В др. ячейку, например **B1** записываем формулу зависимости годовой величины инфляции от ежемесячного роста цен $=(1+A1)^{12}$. Ячейка **B1** должна остаться выделенной. Затем даём команду *Данные*—*Работа с данными*—*Анализ «что-если»*—*Подбор параметра*. На экране появится диалоговое окно, в котором потребуется заполнить 3 строки:

- 1 – указать адрес ячейки, в которую записана формула,
- 2 – указать требуемое значение этой формулы,
- 3 – указать адрес ячейки, где находится изменяемый параметр.

В нашем примере в первой строке появится адрес выделенной ячейки, т.е. нужное значение в ней появится автоматически. Во 2-ю строку надо записать число **1,10** (целая часть от дробной отделяется запятой, а не точкой), а в 3-ю — **A1** (рис. 3.26).

Далее следует щёлкнуть мышью по кнопке «**Ok**», и через мгновение будет получен ответ **0,008003**, т.е. ежемесячный рост цен ~**0,8%** (рис. 3.27). Очевидно, между величинами, помещён-

ными в 1-ю и 3-ю строки окна диалога должна существовать функциональная зависимость. Если в качестве изменяемой указать ячейку, содержание которой не влияет на результат, то программа даст ответ **Решение не найдено**.

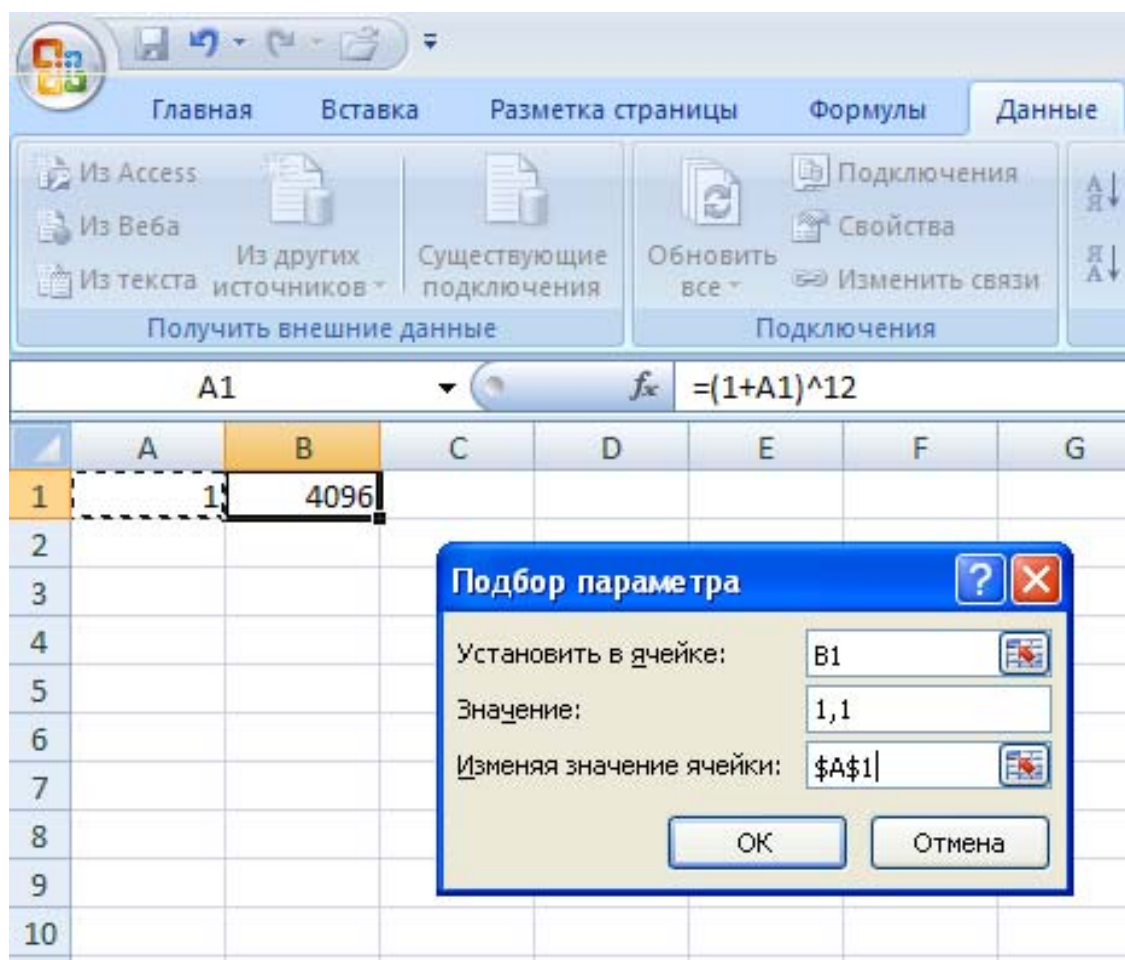


Рис. 3.26

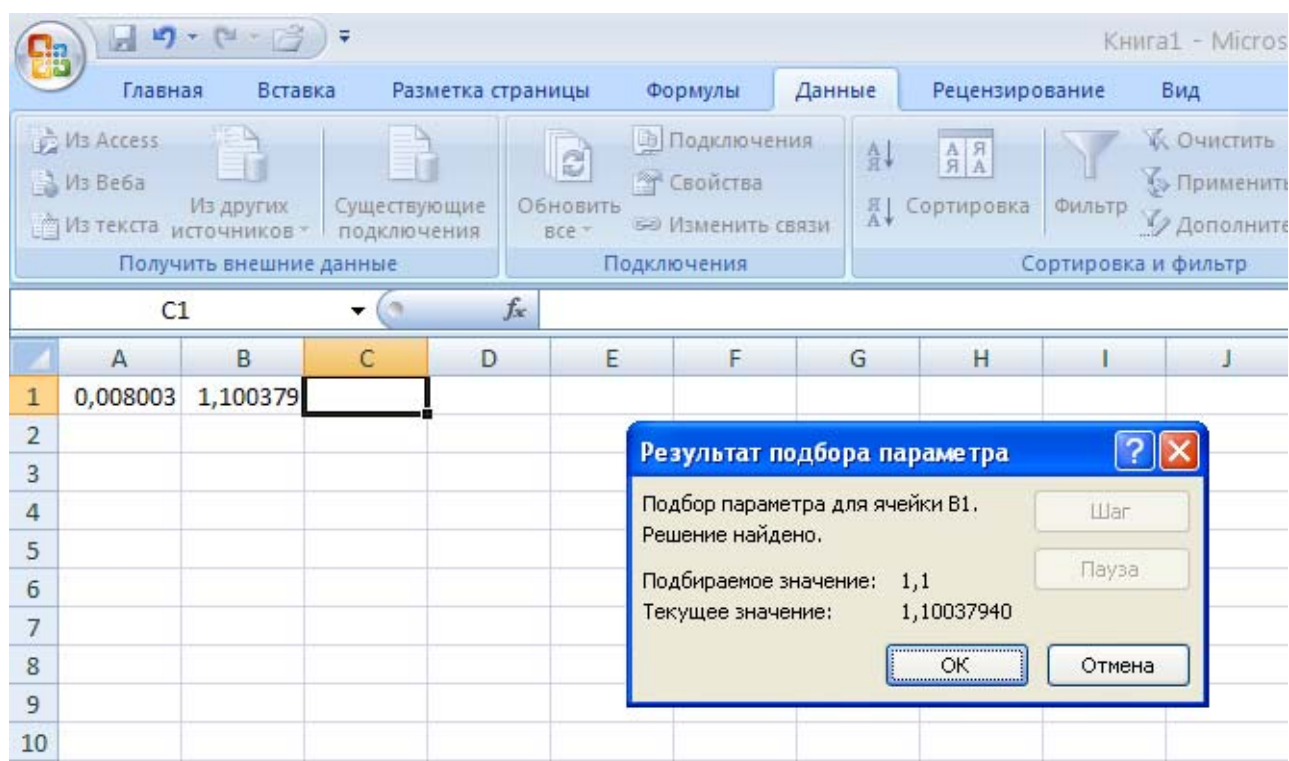


Рис. 3.27


Допустимо совместное использование функций рабочего листа и функции **Подбора параметра**.

§11. Решение оптимизационных задач линейного и нелинейного программирования с помощью процедуры «Поиск решения»

Электронный процессор Excel обладает мощной процедурой решения оптимизационных задач **Поиск решения**.

Загрузку надстройки **Поиск решения** можно выполнить следующим образом.



1. Щелкнуть по значку , затем по кнопке **Параметры Excel**.
2. Выбрать команду **Надстройки**.
3. В окне **Управление** выбрать **Надстройки Excel**.
4. Нажать кнопку **Перейти**.
5. В окне **Доступные надстройки** включить **Поиск решения**.

6. На вкладке **Данные** становится доступной функция **Поиск решения**.

Для эффективного использования процедуры **Поиск решения** полезно соблюдать определенные правила подготовки информации.

Рассмотрим вначале линейные, квадратичные и нелинейные задачи общего вида, затем транспортные задачи ЛП.

Пусть нужно найти max целевой функции

$$z = x_1 - 2x_2$$

при условиях

$$\begin{aligned} -2x_1 + 3x_2 &\leq 12, \\ 3x_1 - x_2 &\leq 18, \\ x_1 + x_2 &\geq 6, \\ x_1, x_2 &\geq 0. \end{aligned}$$

На листе Excel эти данные удобно расположить так, как показано на рис. 3.28.

	A	B	C	D	E	F
1	Наимен.перемен.Xj	X1	X2	Лев. части	Условия	Прав. части
2	Значен.перемен.Xj	1	1			
3	Козф. цел. ф-ии Cj	1	-2	=СУММПРОИЗВ(B\$2:C\$2;B3:C3)		
4	Козф. 1 условия	-2	3	=СУММПРОИЗВ(B\$2:C\$2;B4:C4)	<=	12
5	Козф. 2 условия	3	-1	=СУММПРОИЗВ(B\$2:C\$2;B5:C5)	<=	18
6	Козф. 3 условия	1	1	=СУММПРОИЗВ(B\$2:C\$2;B6:C6)	>=	6

Рис. 3.28

Исходные значения переменных x_j можно задать произвольно, например равными единице.

$$Z = \sum_{j=1}^n c_j x_j$$

Для формирования целевой функции и всех ле-

вых частей ограничений $\sum_{j=1}^n a_{ij} x_j$ ($i=1, \dots, m$) используется функция **СУММПРОИЗВ** ($f_x \rightarrow$ математические \rightarrow **СУММПРОИЗВ**), которая вызывается в строку коэффициентов целевой функции c_j столбца левых частей (D3), а затем, после закрепления адреса

строки «значения переменных x_j », копируется в остальные строки этого столбца.

Для удобства ввода условий в процедуру **Поиск решения** их следует сгруппировать. Например, вначале записать все условия типа \leq , затем \geq , затем $=$.

При такой подготовке в процедуру придется ввести не больше трех **Групповых условий**, что для больших задач существенно сокращает время ввода.

После подготовки исходных данных вызывается процедура **Поиск решения (Данные \rightarrow Анализ \rightarrow Поиск решения)**.

В её диалоговом окне (рис. 3.29) устанавливается адрес ячейки, где находится целевая функция (D3), нужное значение экстремума (max), диапазон адресов значений переменных (B2:C2), диапазон адресов левых и правых частей условий типа \leq (**Добавить** \rightarrow D4:D5 \leq F4:F5 \rightarrow **Добавить**), диапазон адресов левых и правых частей условий типа \geq (**Добавить** – D6 \geq F6 – «ОК»).

В следующем диалоговом окне **Параметры** устанавливаются флажки в позициях **Линейная модель** и **Неотрицательные значения переменных**. Вернувшись к первому диалоговому окну, нажимаем **Выполнить**.

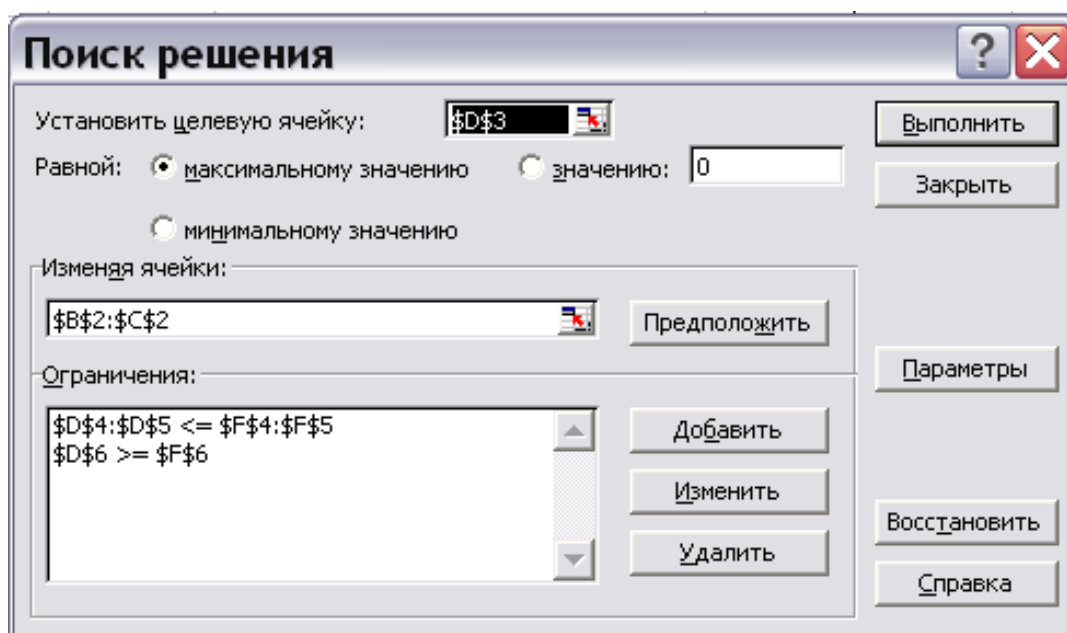


Рис. 3.29

В строке «Значения переменных x_j » таблицы (рис. 3.30) получаем оптимальные значения неизвестных ($x_1=6$, $x_2=0$), а в ячей-

ке целевой функции (D3) - соответствующее её максимальное значение ($Z=6$).

	A	B	C	D	E	F
1	Наимен. перемен. Xj	X1	X2	Лев. Части	Условия	Прав. Части
2	Значен. перем. Xj	6	0			
3	Коэф. целев. ф-ции C	1	-2	6		
4	Коэф. 1 условия	-2	3	-12	<=	12
5	Коэф. 2 условия	3	-1	18	<=	18
6	Коэф. 3 условия	1	1	6	>=	6

Рис. 3.30

Если решение получено, отмечается позиция **Сохранить решение** и интересующие пользователя типы отчетов (результаты, устойчивость, пределы).

Отчеты выводятся на отдельные листы Excel, которые вызывают из командной строки.

Если выводится сообщение, что решение не найдено, то либо задача поставлена некорректно (условия несовместимы или экстремум в бесконечности), либо поставленные в диалоговом окне **Параметры** ограничения по числу итераций, времени решения и точности слишком жесткие и их нужно ослабить (увеличить число итераций и время решения, уменьшить требования к точности).

Для решения задачи с нелинейной целевой функцией нужно в ячейку целевой функции ввести нужное выражение и в диалоговом окне **Параметры** снять флажок в позиции **Линейная модель**.

Например, чтобы решить рассмотренную выше задачу с целевой функцией

$$R = x_1^2 - 2x_2^2 \rightarrow \max$$

набираем эту функцию в ячейке (F3) и указываем ее в диалоговом окне **Поиск решения**.

В результате получим $x_1=6,35$, $x_2=1,06$, $R=38,12$ (рис. 3.31).

	A	B	C	D	E	F
1	Наимен. перемен. Xj	X1	X2	Лев. части	Условия	Прав. части
2	Значен. перемен. Xj	6,35	1,06			
3	Козф. цел. ф-ии Cj	1	-2	4,24		38,12
4	Козф. 1 условия	-2	3	-9,53	<=	12
5	Козф. 2 условия	3	-1	18,00	<=	18
6	Козф. 3 условия	1	1	7,41	>=	6

Рис. 3.31

Для получения целочисленных решений достаточно в ограничения добавить диапазон изменяемых ячеек с условием целочисленные (**цел**).

Добавив условия целочисленности, в нашей задаче получим $x_1=6$, $x_2=0$, $R=36$ (в диалоговом окне **Поиск решения - Добавить** → **B2:C2=целое**).

Если просто округлить предыдущее решение до целых чисел $x_1=6$, $x_2=1$, то целевая функция $R=34$, что хуже целочисленного **max**.

Процедура **Поиск решения** успешно справляется не только с задачами квадратичного программирования, но и с задачами собственно нелинейного программирования.

Пусть нужно найти **max** целевой функции

$$R = x_1^2 - 2x_2^2$$

при условиях

$$-2x_1 + 3x_2^2 \leq 12,$$

$$3x_1^2 - x_2 \leq 18,$$

$$x_1^2 + x_2^2 \geq 6,$$

$$x_1, x_2 \geq 0.$$

Исходные данные можно записать, например, в соответствии с рис. 3.32.

	А	В	С	Д	Е	Ф
1	Наимен.перемен.Xj	X1	X2	Лев. части	Условия	Прав. части
2	Значен.перем.Xj	1	1			
3	Козф.цел.ф-ии Cj	1	-2	=B3*B2^3+C3*C2^2	ц.ф.	
4	Козф. 1 условия	-2	3	=B4*B2+C4*C2^2	<=	12
5	Козф. 2 условия	3	-1	=B5*B2^2+C5*C2	<=	18
6	Козф. 3 условия	1	1	=B6*B2^2+C6*C2^2	>=	6

Рис. 3.32

В результате решения получим
 $x_1=2.470$, $x_2=0.309$, $R=14.89$

При решении транспортной задачи ЛП тоже важно удобно расположить исходные данные в таблице Excel.

Пусть нужно найти min затрат при перемещении грузов из трех пунктов отправления с запасами $a_1=8$, $a_2=5$, $a_3=7$ в четыре пункта назначения с потребностями $b_1=4$, $b_2=4$, $b_3=2$, $b_4=10$.

Затраты на перемещение единицы груза по каждому маршруту соответствуют матрице:

3	4	5	6
4	4	2	1
2	1	6	8

Можно рекомендовать расположить исходные данные, как показано на рис. 3.33

	A	B	C	D	E	F	G	H
1		3	4	5	6			
2	Матрица	4	4	2	1	=СУММПРОИЗВ		
3	цен C _{ij}	2	1	6	8	(B1:E3;B5:E7)		
4						лев части	условия	прав части
5	Матрица	1	1	1	1	=СУММ(B5:E5)	=	8
6	перемен	1	1	1	1	=СУММ(B6:E6)	=	5
7	ных X _{ij}	1	1	1	1	=СУММ(B7:E7)	=	7
8								
9	левые	=СУММ	=СУММ	=СУММ	=СУММ			
	части	(B5:B8)	(C5:C8)	(D5:D8)	(E5:E8)			
10	условия	=	=	=	=			
	правые							
11	части B _j	4	4	2	10			
12								

Рис. 3.33

Исходные значения всех неизвестных x_{ij} принимают произвольно, например, равными единицам.

Целевая функция формируется функцией **СУММПРОИЗВ**, в диалоговое окно которой первый массив c_{ij} и второй x_{ij} вводятся движениями курсора по главным диагоналям соответствующих матриц. Значения левых частей условий по запасам и по потребностям формируются функцией сумм (рис. 3.33).

Отметим, что условием существования решения является равенство суммы запасов и суммы потребностей:

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j .$$

Если $\sum_{i=1}^m a_i > \sum_{j=1}^n b_j$, для обеспечения баланса добавляется фик-

тивный пункт назначения с потребностями $b_{n+1} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j$. Чтобы не менять величину целевой функции, затраты на перемещение грузов к фиктивному пункту принимаются равными нулю:

$$C_{i,n+1} = 0 \quad (i=1, \dots, m).$$

Аналогично, если $\sum_{i=1}^m a_i < \sum_{j=1}^n b_j$, добавляется фиктивный пункт отправления с запасами $a_{m+1} = \sum_{i=1}^m b_j - \sum_{j=1}^n a_i$ и затратами на перемещение

$$C_{m+1,j} = 0 \quad (j=1, \dots, n).$$

В диалоговом окне **Поиск решения** устанавливают адрес целевой функции, вид экстремума, вводят адреса массива неизвестных (курсор по диагонали матрицы x_{ij}).

В окно условий вводят (**Добавить**) диапазон адресов левых частей условий по запасам, знак равенства и соответствующий диапазон правых частей условий по запасам. Затем выполняют то же с условиями по потребностям (рис. 3.34).

В окне **Параметры** отмечают **Линейная модель** и **Неотрицательные значения**. Компоненты решения получают в матрице x_{ij} , а соответствующее значение целевой функции $Z = \sum \sum c_{i,j} x_{i,j}$ в ячейке F2 (рис. 3.35).

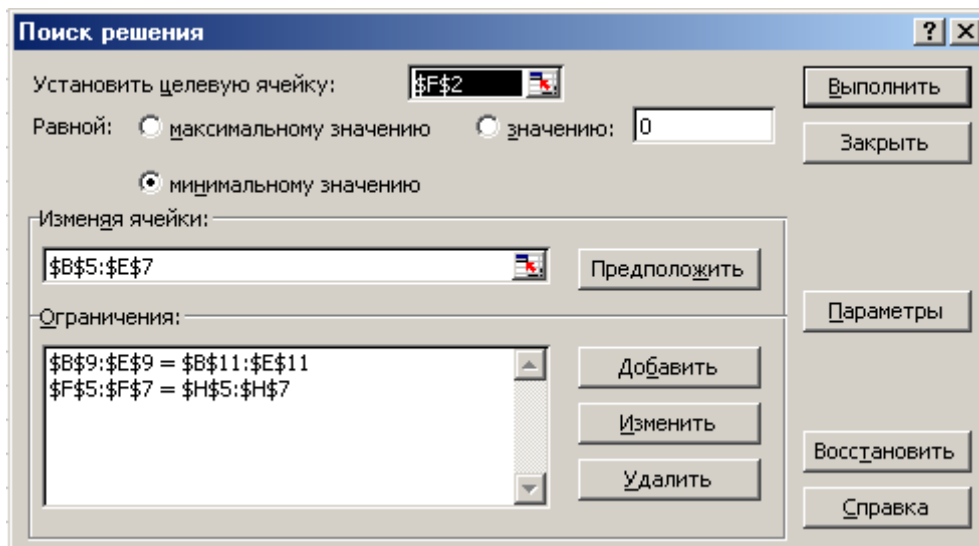


Рис. 3.34

	A	B	C	D	E	F	G	H
1		3	4	5	6			
2	Матрица	4	4	2	1	58		
3	цен C_{ij}	2	1	6	8			
4						лев части	условия	прав части
5	Матрица	1	0	2	5	8	=	8
6	переменн	0	0	0	5	5	=	5
7	ых X_{ij}	3	4	0	0	7	=	7
8								
9	левые							
10	части	4	4	2	10			
11	условия	=	=	=	=			
12	правые							
13	части B_j	4	4	2	10			

Рис. 3.35

При необходимости создаются отчеты по результатам, устойчивости, пределам.

Задание

Решить систему линейных уравнений из задания к §8 с помощью процедуры **Поиск решения**. В качестве целевой функции возьмите выражение $L = \sum_j N x_j$. N – индивидуальный номер.

§12. Графическое представление данных с помощью диаграмм

Диаграммы, как правило, используются для наглядного представления соотношений между какими-либо величинами или динамики их изменения. Excel предоставляет в распоряжение пользователя целый ряд средств работы с диаграммами. Во многих случаях целесообразно перед созданием диаграммы выделить ячейки, содержащие необходимые данные.

Покажем на графике затраты студентов (рис. 3.36).

Для вызова конструктора диаграмм надо выбрать **Вставка** → **Диаграммы**. Затем необходимо указать данные для графика **Кон-**

структур—»*Данные*—»*Выбрать данные* (рис.3.37). Появится диалоговое окно **Выбор источника данных**. Нажав на кнопку в правой части поля **Диапазон данных для диаграммы**, выделим столбцы «ФИО» и «Затраты» в таблице (без заголовков). Несмежные столбцы выделяются при нажатой клавише «Ctrl».

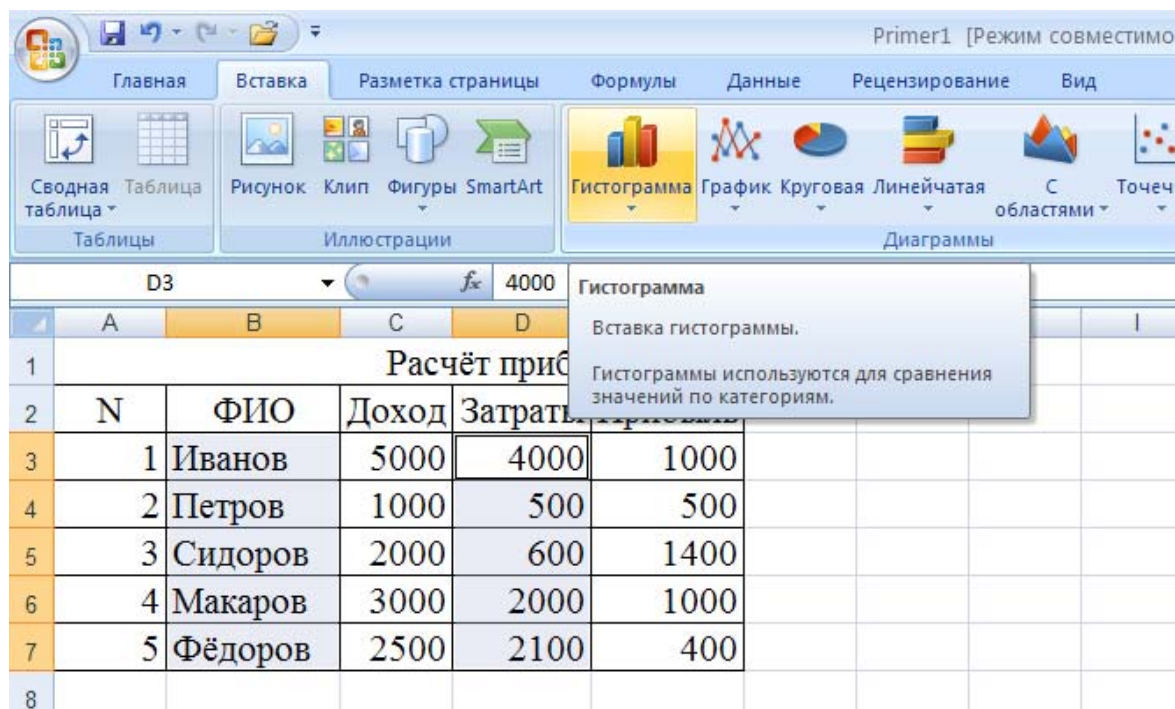


Рис. 3.36

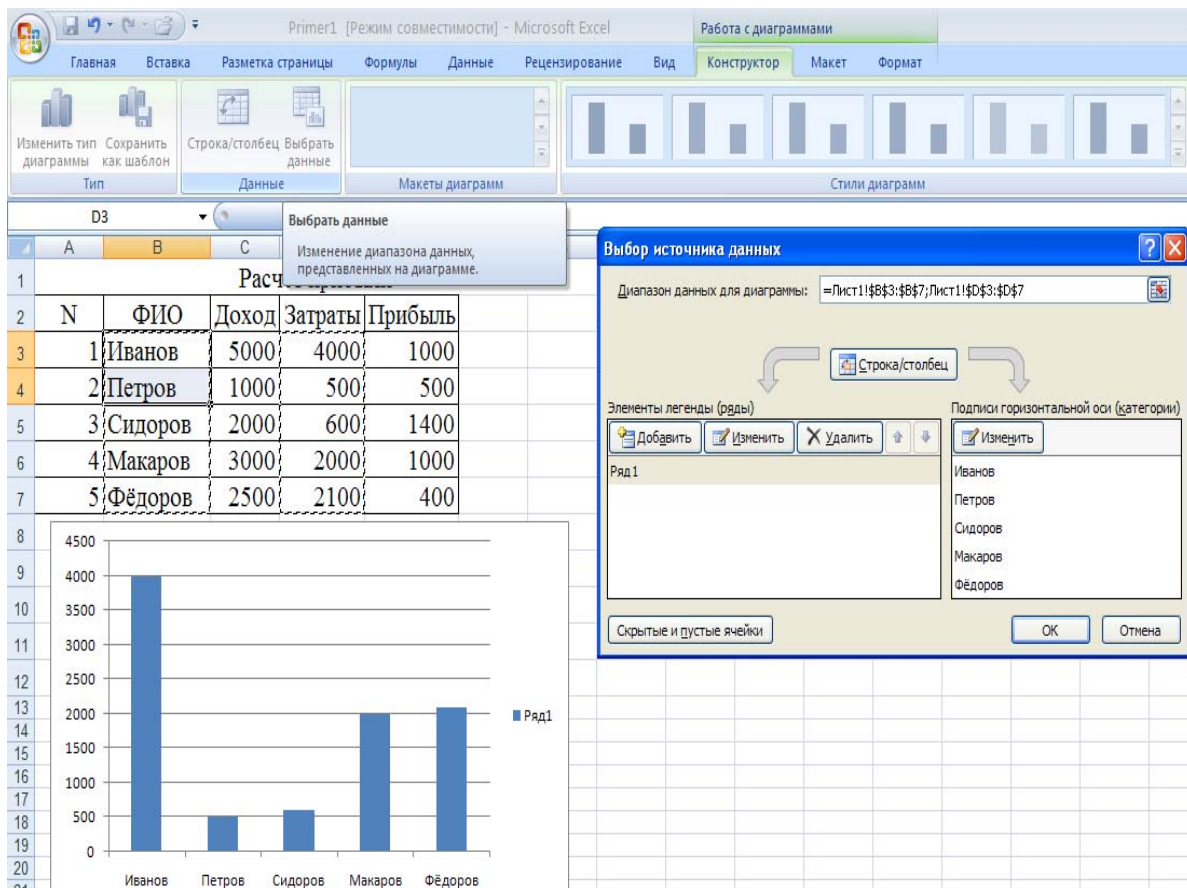


Рис. 3.37

Чтобы задать заголовок графику, надо нажать на кнопку **Изменить** в поле **Элементы легенды** и ввести название графика (рис. 3.38).

Следует также отметить, что вертикальная ось — ось значений, а горизонтальная — ось категорий, т.е. на ней все названия, даже если они имеют форму цифровых значений, воспринимаются как текст и располагаются на равных расстояниях. Это следует иметь в виду при построении графиков, что возможно.

Если дважды щёлкнуть мышью по созданной диаграмме, вокруг неё появится контур, что свидетельствует о возможности редактирования диаграммы. В частности, потянув левой клавишей мыши какой-либо из квадратов на этом контуре, можно изменять размер области диаграммы. Пользователь имеет возможность дополнить диаграмму новыми данными, например, показать на графике и «Доход» (рис. 3.39).

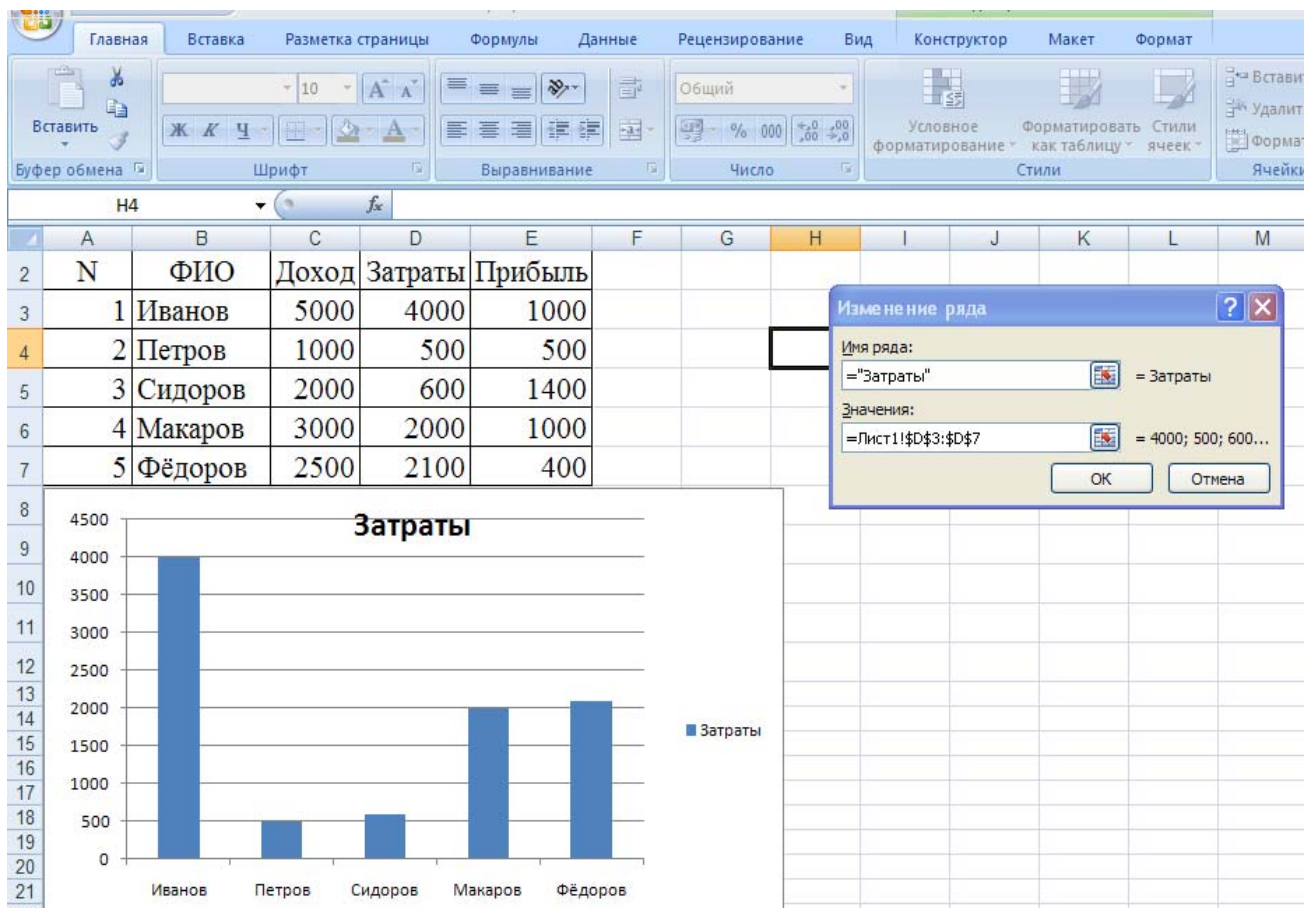


Рис. 3.38

Щёлкнув мышью по какой-либо части диаграммы, в области построения, пользователь получает возможность редактировать отдельные её элементы.

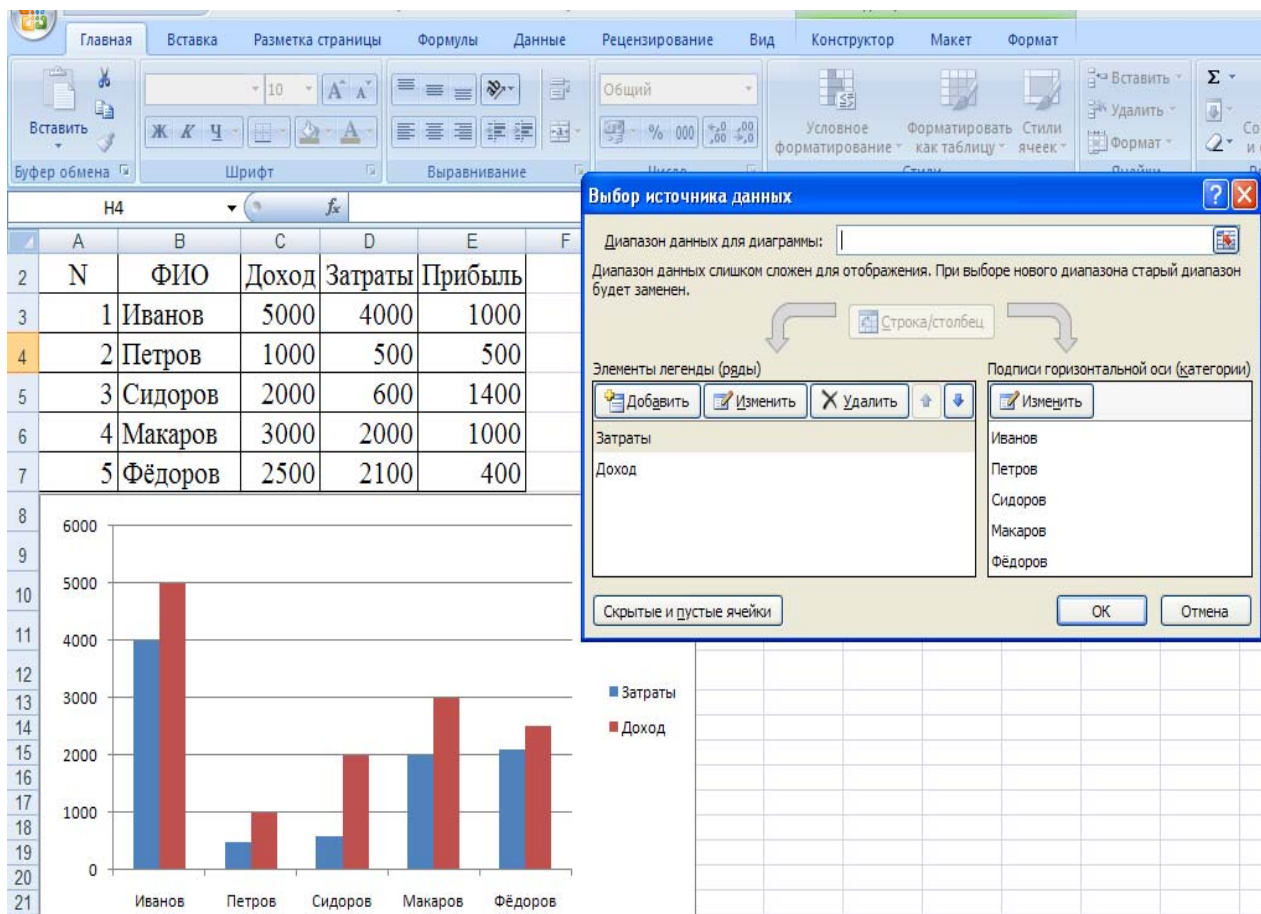


Рис. 3.39

§13. Использование Excel для управления базами данных

Наряду с математическими расчётами Excel даёт возможность производить обработку данных, свойственную системам управления базами данных.

Прежде чем производить такую обработку, необходимо создать **список**. Термин «список» в Excel используется для обозначения базы данных и представляет собой набор записей. Запись состоит из логически связанных между собой данных различных типов. Например, каждая запись в адресной книге может содержать следующие данные: фамилия, имя, отчество, адрес, телефон, место работы, должность. Под каждое из этих данных в записи отводится отдельное поле. Поля определяют структуру списка

(базы данных). Списки создаются в обычных таблицах, но при этом должны соблюдаться следующие правила.

1. Имена полей (столбцов) должны быть указаны в верхней строке списка (строке заголовка).
2. Каждая запись размещается на отдельной строке.
3. Недопустимы пустые строки между записями, а также между заголовком и первой записью (пустая строка интерпретируется программой как конец списка).

Для удобства работы со списком, особенно при большом количестве записей, Excel предоставляет пользователю ещё целый ряд средств, среди которых важнейшими являются **сортировка** и **фильтрация** записей. Для того чтобы активизировать эти функции, необходимо щелкнуть в какую-либо ячейку таблицы.

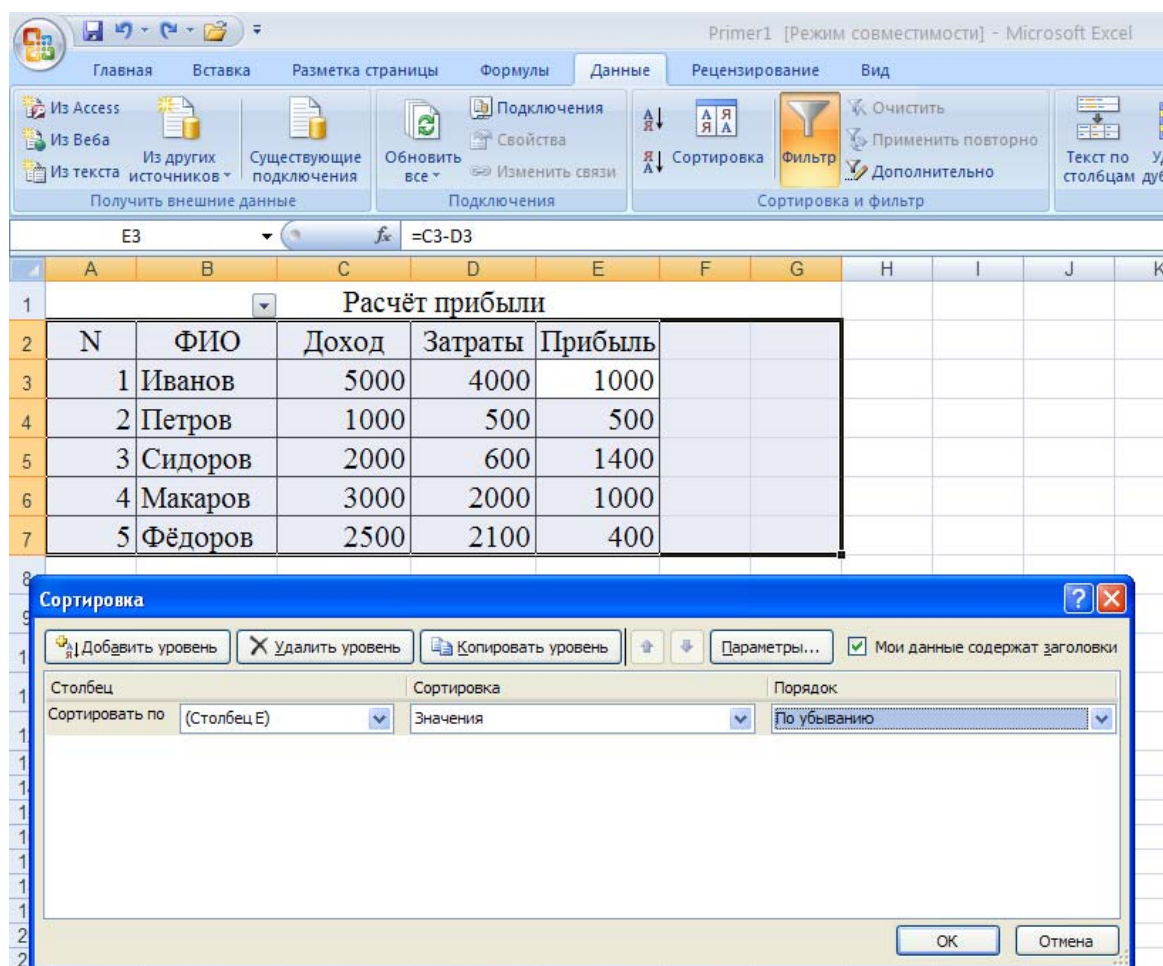


Рис. 3.40

Операцию сортировки можно произвести с использованием команды **Данные** → **Сортировка и фильтр** → **Сортировка**. В

появившемся диалоговом окне следует выбрать параметры сортировки – столбец, порядок сортировки. Так как в таблице есть заголовки, нужно поставить флажок в **Мои данные содержат заголовки** (рис. 3.40). Результат сортировки представлен на рис.3.41.

	A	B	C	D	E	F	G
1	Расчёт прибыли						
2	N	ФИО	Доход	Затраты	Прибыль		
3	3	Сидоров	2000	600	1400		
4	1	Иванов	5000	4000	1000		
5	4	Макаров	3000	2000	1000		
6	2	Петров	1000	500	500		
7	5	Фёдоров	2500	2100	400		

Рис. 3.41

	A	B	C	D	E	F	G	H	I	J	K
1	Расчёт прибыли										
2	N	ФИО	Доход	Затраты	Прибыль						
3	3	Сидоров	2000	600	1400						
4	1	Иванов	5000	4000	1000						
5	4	Макаров	3000	2000	1000						
6	2	Петров	1000	500	500						
7	5	Фёдоров	2500	2100	400						

Сортировка

Мои данные содержат заголовки

Столбец	Сортировка	Порядок
Сортировать по (Столбец Е)	Значения	По убыванию
Затем по (Столбец С)	Значения	По возрастанию

Рис. 3.42

Возможна сортировка с использованием нескольких уровней. Например, необходимо выполнить сортировку по убыванию прибыли и возрастанию доходов. Для этого с помощью кнопки **Добавить уровень** добавляют нужный столбец и нужный порядок сортировки и т. д. (рис. 3.42).

Результат такой сортировки показан на рис. 3.43.

	A	B	C	D	E	F
1	Расчёт прибыли					
2	N	ФИО	Доход	Затраты	Прибыль	
3	3	Сидоров	2000	600	1400	
4	4	Макаров	3000	2000	1000	
5	1	Иванов	5000	4000	1000	
6	2	Петров	1000	500	500	
7	5	Фёдоров	2500	2100	400	

Рис. 3.43

Ещё один способ, облегчающий решение задачи выбора нужной записи, — использование фильтров. Выделяют столбец, по данным которого предполагается производить фильтрацию, например столбец «**ФИО**», и два раза щелкают по **Фильтр (Данные) → Сортировка и фильтр → Фильтр** (рис. 3.44).

	A	B	C	D	E	F	G	H	I
1	Расчёт прибыли								
2	N	ФИО	Доход	Затраты	Прибыль				
3	1	Иванов	5000	4000	1000				
4	2	Петров	1000	500	500				
5	3	Сидоров	2000	600	1400				
6	4	Макаров	3000	2000	1000				
7	5	Фёдоров	2500	2100	400				

Рис. 3.44

Пусть пользователя интересуют фамилии *Макаров* и *Сидоров*. Нажав на кнопку списка, следует поставить флажки у нужных фамилий (рис. 3.45) и «*ОК*». После этого на экране останутся записи, содержащие информацию о Макарове и Сидорове (рис. 3.46). Все отфильтрованные записи отображаются на экране, и пользователь имеет возможность работать с ними, как с обычной таблицей Excel.

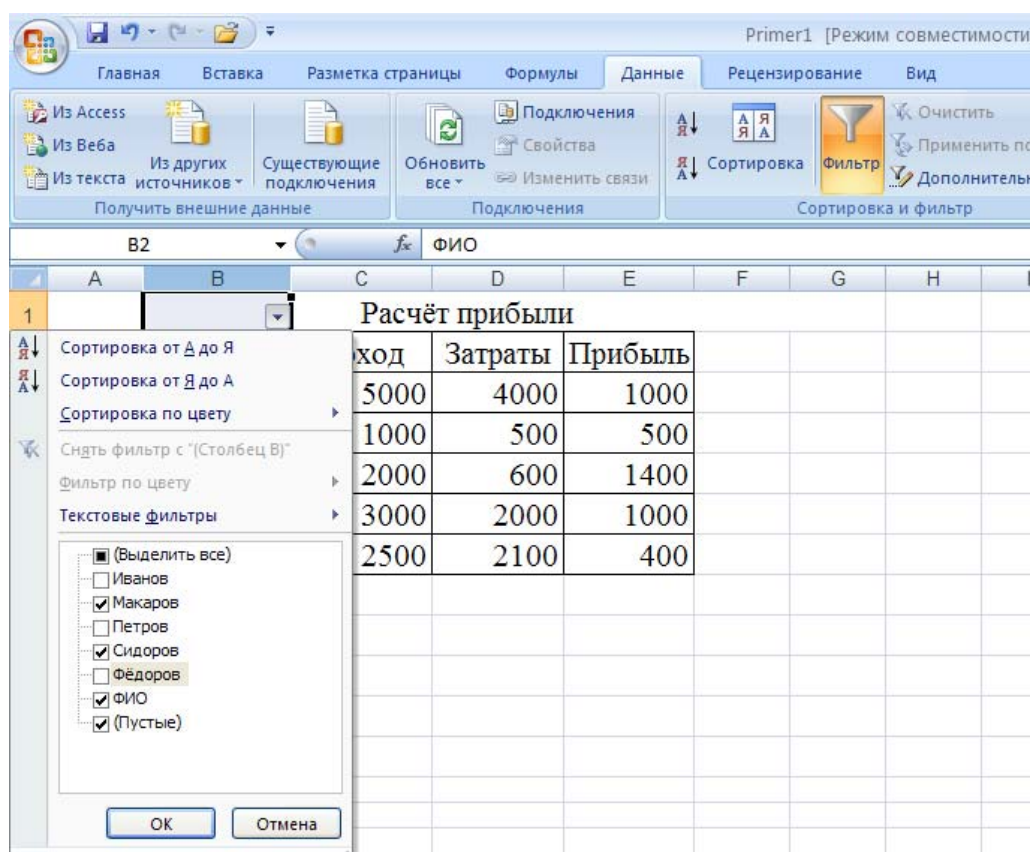


Рис. 3.45

Наряду с простым фильтром имеется ещё одна разновидность фильтрации – расширенный фильтр. Она даёт возможность произвести фильтрацию на части таблицы, а полученный результат перенести в заданное место, например на новый лист.

Расчёт прибыли					
N	ФИО	Доход	Затраты	Прибыль	
3	Сидоров	2000	600	1400	
4	Макаров	3000	2000	1000	

Рис. 3.46

Задание

Самостоятельно в таблицу «Мониторы» (рис. 3.5) добавьте еще одно устройство, например принтер, и столбец «код устройства», например 100 - код мониторов, 200 – код принтеров. Выполните отбор по кодам устройств и по ценам.

§14. Анализ данных в Excel с помощью сводных таблиц

Предположим, что база данных содержит ежедневно обновляемую информацию по продаже архитектурных проектов в трех отделениях фирмы (рис. 3.47).

Необходимость систематизации и обобщения информации базы данных может возникнуть при оценке работы отделений фирмы: Южного, Северного и Западного.

Перечислим ряд вопросов, которые могут интересовать руководителя:

1. У какого отделения больше сумма продаж?
2. Какой тип проекта, какое отделение продает больше?
3. Как распределена сумма продаж по разным типам проектов в отделениях?

Сводная таблица поможет ответить на эти вопросы.

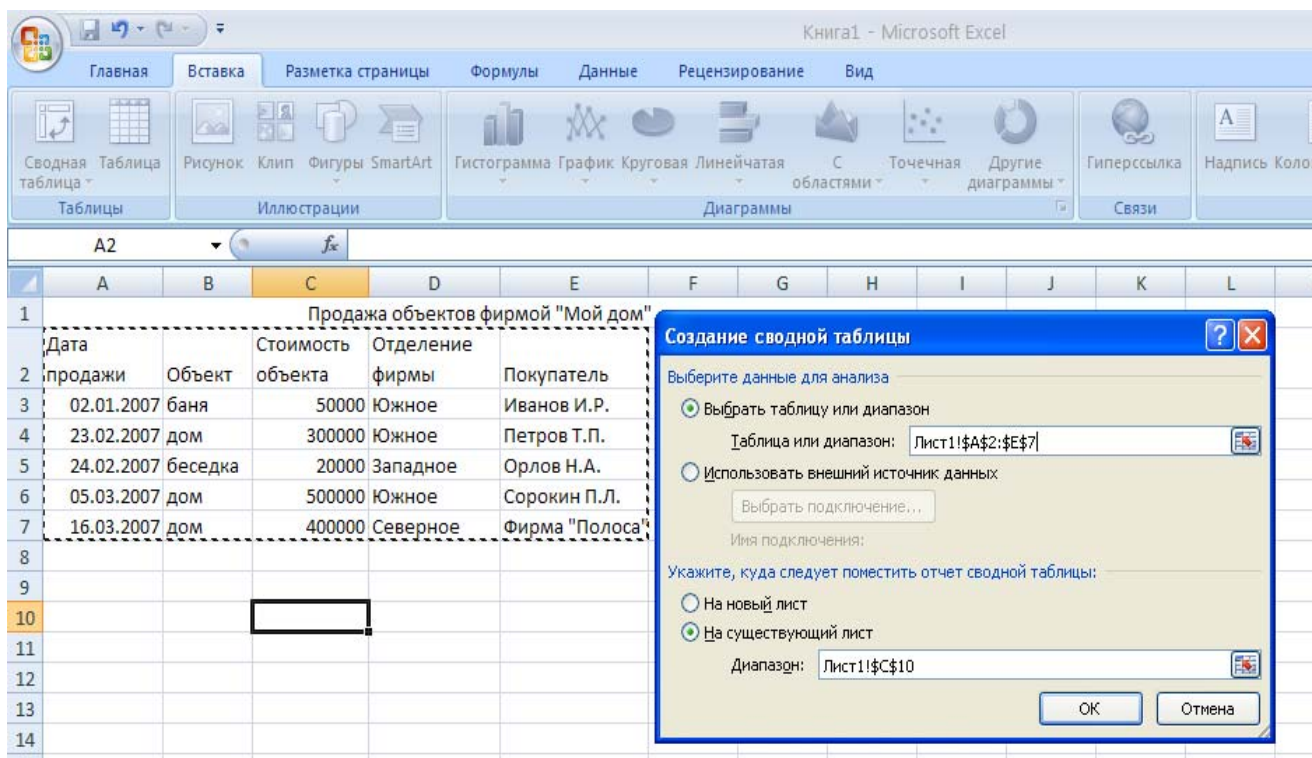


Рис. 3.47

Создать сводную таблицу можно следующим образом:
Вставка → **Таблицы** → **Сводная таблица** (рис. 3.48).

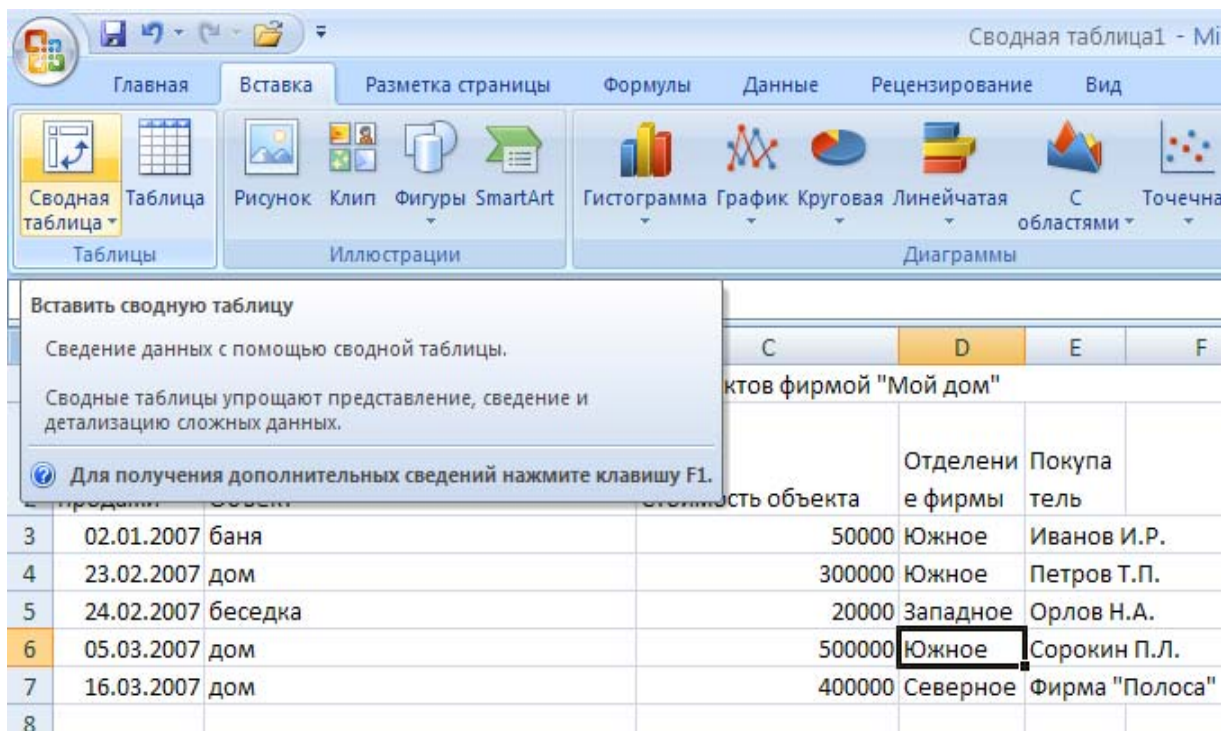


Рис. 3.48

Затем в диалоговом окне нужно указать диапазон данных, которые будут использоваться в сводной таблице (рис. 3.49).

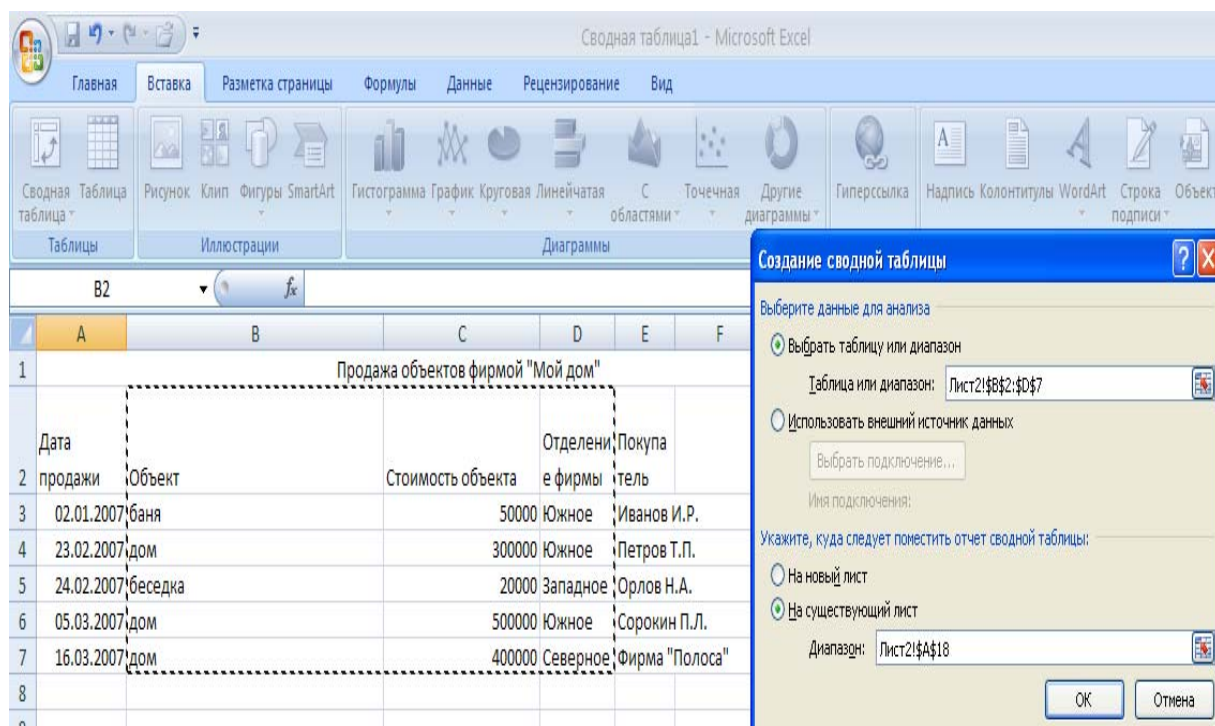


Рис. 3.49

В следующем диалоговом окне следует выбрать, отметив галочками, поля для добавления в отчет и переместить их в нужные области нижней части окна (рис. 3.50). Сводная таблица готова.

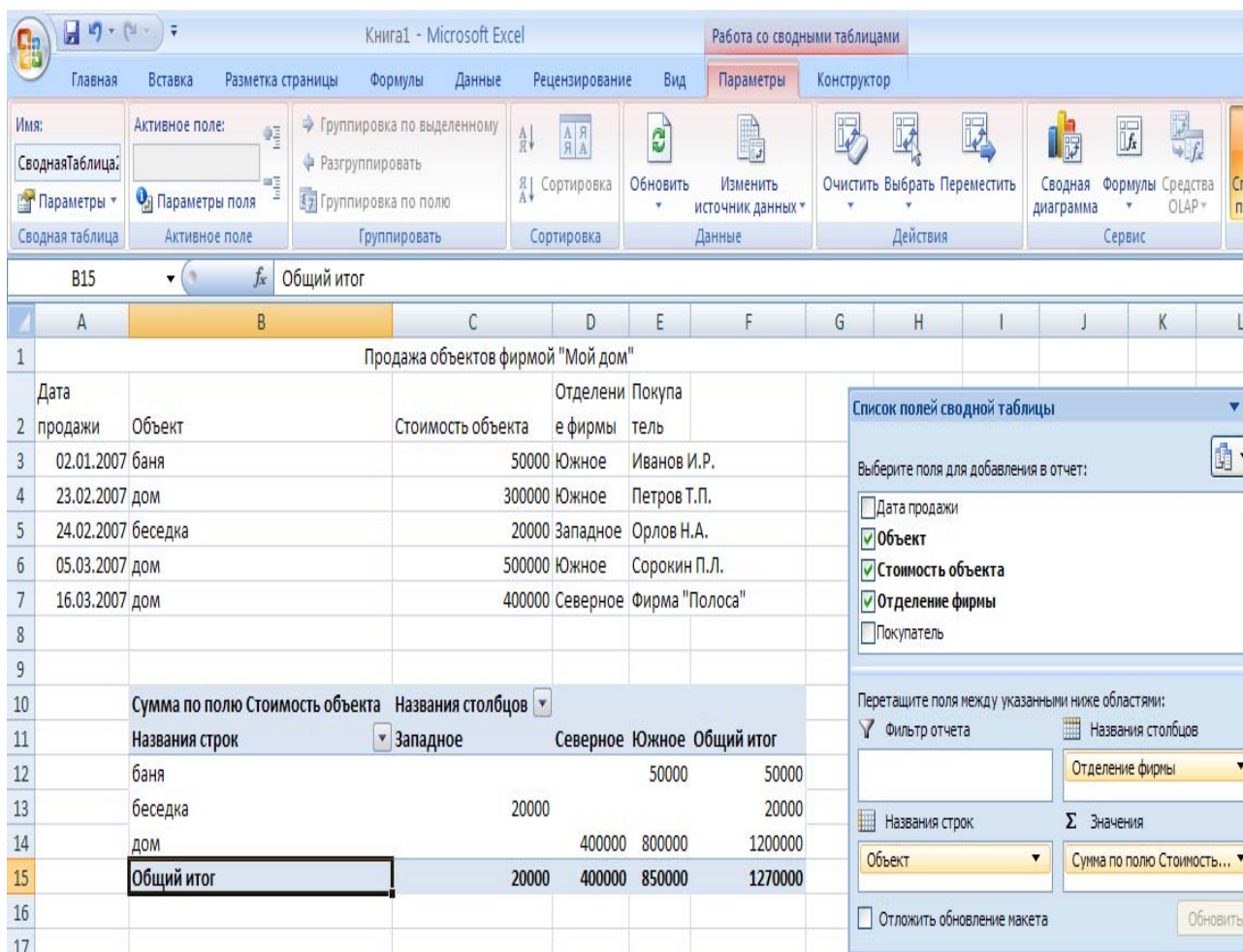


Рис. 3.50

§15. Создание и использование макросов в Excel

Язык макрокоманд Visual Basic Application (VBA) – основной инструмент для настройки и автоматизации рутинных операций в Excel.

Макрос – это последовательность команд, которая используется для автоматизации некоторых действий Excel, благодаря чему повышается эффективность работы и уменьшается число ошибок.

После создания макроса достаточно запустить его, и целая последовательность однотипных операций будет выполнена автоматически. Простейший способ создания макроса – запись последовательности выполняемых вами действий и автоматическое преобразование ее в макрос VBA. В сущности, макрос является программой, при запуске которой Excel вновь выполнит записан-

ную последовательность действий. Рассмотрим пример создания макроса, который будет строить график на Листе2 по данным таблицы на Листе1.

Создаем макрос: **Вид**→**Макросы**→**Запись макроса**. Затем выполняем действия для построения графика: переходим на Лист2, щелкаем по кнопке **Мастера диаграмм**, переходим на Лист1 и в таблице выделяем данные для отображения на графике – столбцы «Название монитора» и «Цена монитора» (рис. 3.5). Получаем график.

Для запуска макроса необходимо выполнить **Вид**→**Макросы**→ выбрать макрос и нажать **Выполнить**. На Листе2 будет построен график.

Для создания макросов также используют язык программирования VBA. Рассмотрим пример макроса для вычисления площади треугольника.

Сначала создадим макрос, выполнив следующие действия:

Вид→**Макросы**→**Запись макроса**→в диалоговом окне задать имя макроса и сохранить (рис. 3.51).

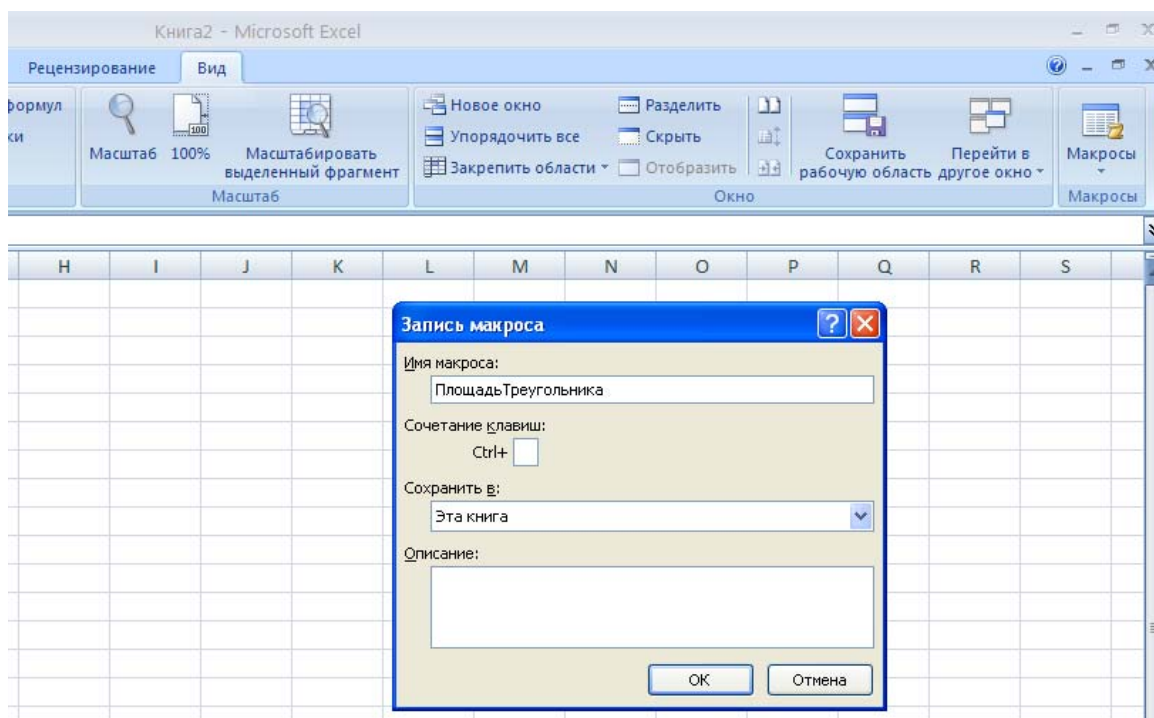


Рис. 3.51

Остановить запись
(**Вид**→**Макросы**→**Макросы**→**Остановить запись**). Затем вы-

полнить **Вид**→**Макросы**→**Макросы**→Выбрать макрос в диалоговом окне и нажать на **Изменить**, попадем в редактор Visual Basic.

Сначала построим форму, выполнив следующие действия: **Insert**→**UserForm**. Щелкнув один раз внутри формы, откроем окно с компонентами для формы. Компонентой **Commandbutton1** создаем кнопку, на которой напишем «Вычислить площадь». Для этого выделим компоненту **Commandbutton1** на форме и в окне свойства (**Properties**) напротив **Caption** введем «Вычислить площадь».

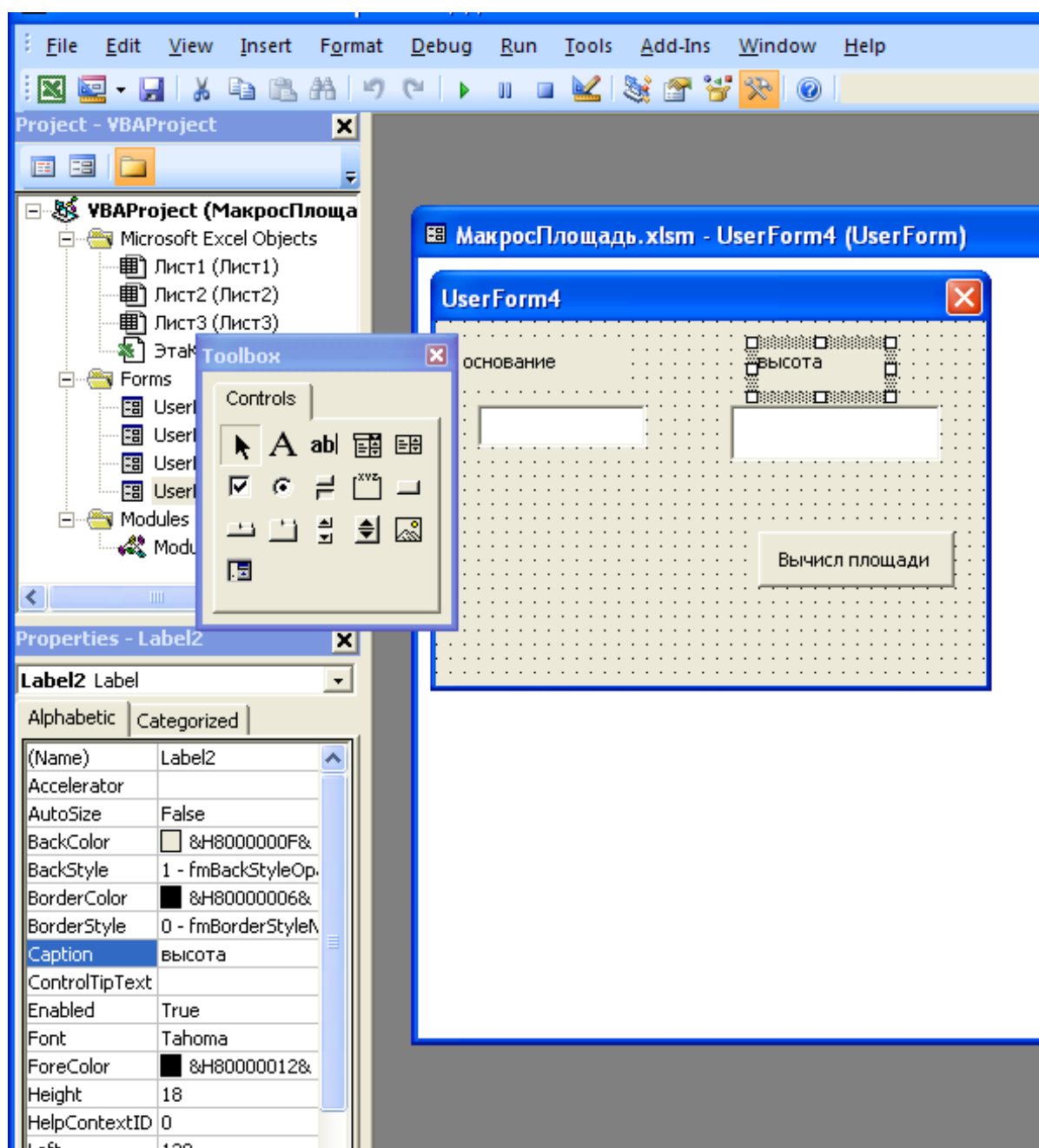


Рис. 3.52

Компонентой **TextBox** (кнопка с буквами «ab») создадим поля для ввода значений основания и высоты. Компонентой **Label** (кнопка с буквой «A») подпишем поля (рис. 3.52).

Нажав дважды на кнопку **Вычислить площадь**, введем текст модуля (рис. 3.53):

```
Private Sub
  Dim a,h,s,c As single
  a=TextBox1
  h=TextBox2
  s=(a*h)/2
  c=Str(S)
  MsgBox c
End Sub
```

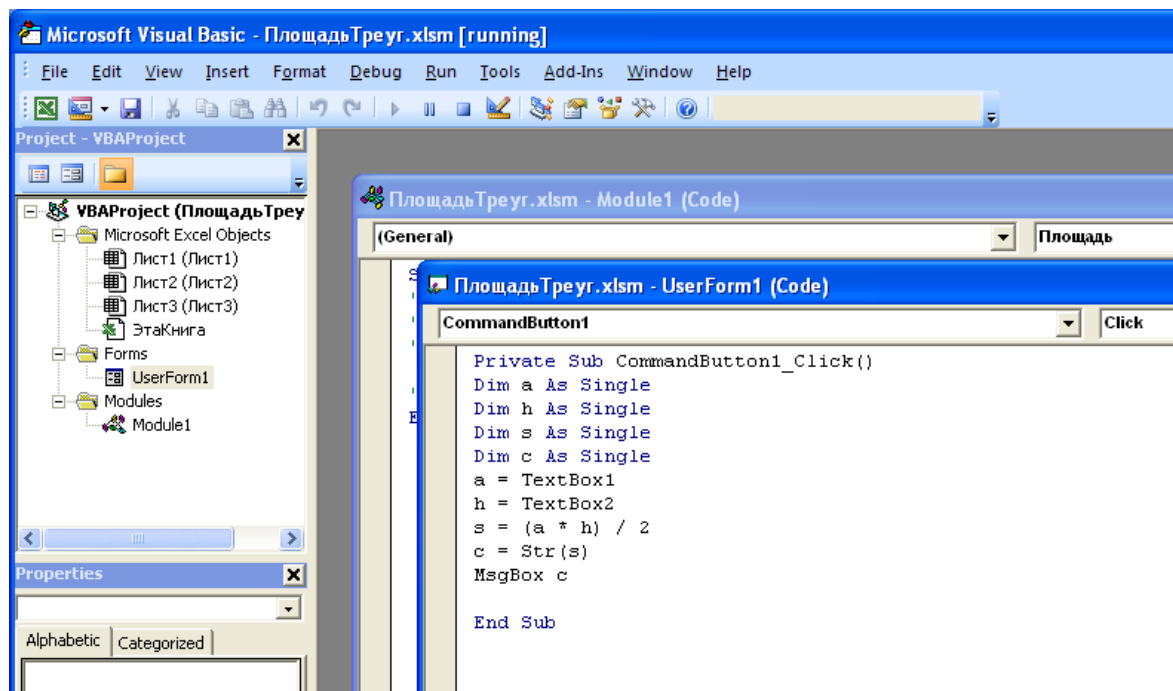


Рис. 3.53

Запустим макрос на выполнение командами **Run→Run Sub/UserForm** и получим результат (рис. 3.54).

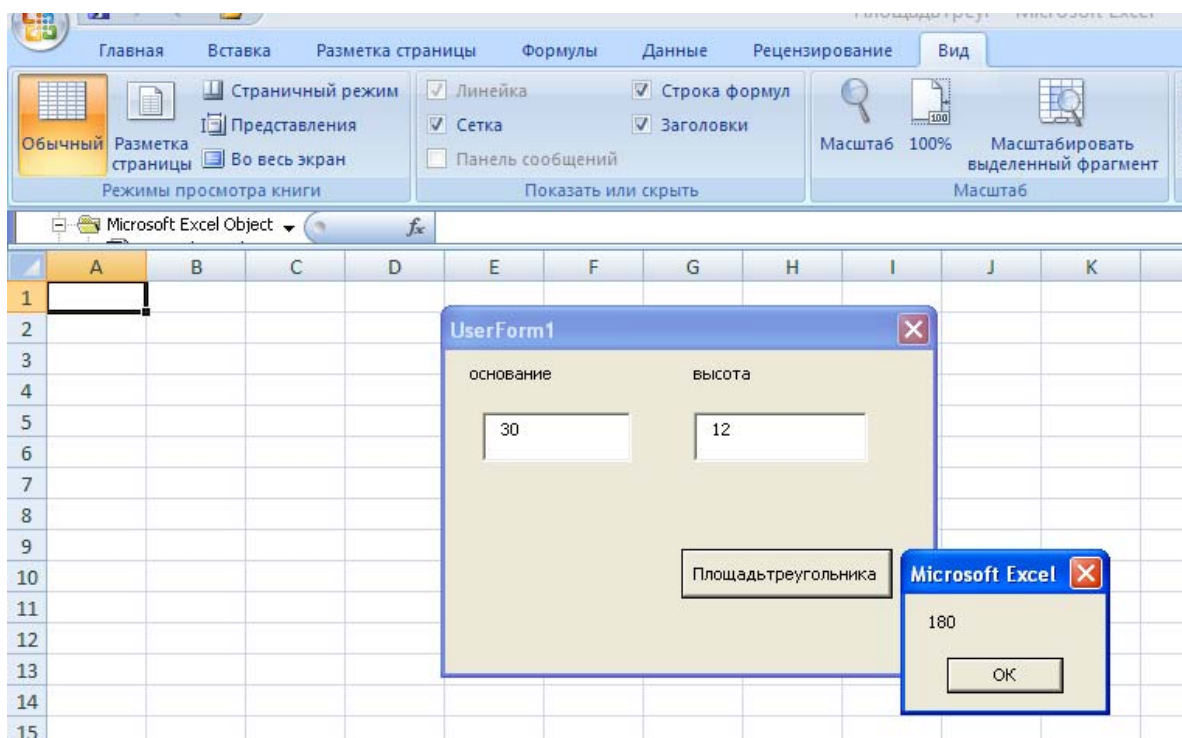


Рис. 3.54

Задание

Подготовьте макрос для решения по формулам Крамера системы линейных уравнений с матрицей коэффициентов размерностью $n \times n$, где n – номер по журналу+2 (см. §8).

§16. Одновременное использование Excel и Word

Самое простое средство взаимосвязи между программами из пакета Microsoft Office - это *буфер обмена*. Буфер обмена — часть памяти, используемая для временного хранения информации. Скопировать информацию в буфер и вставить её в документ можно с использованием соответствующей кнопки на стандартной панели инструментов. Недостатком такого способа связи является его статический характер, т.е. изменения, например, в электронной таблице не сопровождаются автоматически соответствующими изменениями в текстовом документе.

Для создания динамической связи между двумя документами необходимо выполнить следующие действия **Главная—** **›Вставить—** **›Специальная вставка** либо в Excel, либо в Word.

Обычно итоговый документ создают в Word, вставляя в него промежуточные результаты, полученные с использованием электронных таблиц. Поэтому ниже будут рассматриваться примеры, связанные с переносом информации из Excel в Word.

Существуют два типа динамической связи между документами: *внедрение* и *динамический обмен данными*. В обоих случаях необходимо запустить приложения, в которых были созданы связанные документы, открыть оба документа, выделить требуемую часть электронной таблицы и скопировать её в буфер обмена. Затем следует перейти в Word и дать команду **Главная**—**Вставить**—**Специальная вставка**. На экране появится диалоговое меню, где слева будут значки **Вставить**, **Связать**, в центре - окно с вопросом о типе вставляемого объекта (текст, лист таблицы, рисунок и т.д.), а справа — кнопки «**ОК**» и «**Отмена**». Если выбрать режим вставки, а потом нажать «**ОК**», то в текстовом документе в точке, соответствующей положению курсора, появится выделенный фрагмент электронной таблицы. Внешний вид этой таблицы не отличается от вида обычной таблицы в Word, но если по внедренной таблице дважды щелкнуть мышкой, то она превратится в таблицу Excel и появится возможность обрабатывать содержащуюся в ней информацию при помощи Excel. Таким образом, одновременно используются возможности Word и Excel.

Однако, если имеется отчёт большого размера и в его текст требуется оперативно вносить обновленные результаты, появляющиеся в процессе расчётов с использованием электронных таблиц, целесообразно применять не внедрение таблиц, а устанавливать связи между документами. В этом случае каждый раз при открывании текстового документа происходит обновление связей, т.е. вносятся изменения, соответствующие изменениям, внесённым к этому моменту в электронную таблицу. Связанную информацию можно редактировать только в исходном приложении. Для того чтобы открыть это приложение, необходимо дважды щёлкнуть мышью по связанной информации в документе Word.

Использование обоих методов динамического обмена данными открывает перед пользователем широкие возможности. Также

существует возможность осуществлять динамический обмен данными между всеми программами пакета Microsoft Office.

Задачи для самостоятельного решения

Тема: Математические функции

1. В первый столбец таблицы ввести названия месяцев (январь .. декабрь), во второй столбец показания электросчётчика в начале каждого месяца, в третий столбец — показания счетчика в конце каждого месяца, в четвертый столбец — расход электроэнергии в каждом месяце, а в пятый месячную оплату с учётом тарифа. Величину тарифа указать в 1-й ячейке 6-го столбца. В 1-й, 3-й, 4-й, 5-й столбцы вводить информацию с использованием автозаполнения ячеек. В 1-ю строку таблицы ввести заголовок.

2. Создать таблицу, содержащую данные о ценах на какие-либо товары в прошлом году и в текущем году. Определить величину изменения цены выбранных товаров в процентах. При расчетах использовать автозаполнение ячеек. В 1-ю строку таблицы ввести заголовок.

3. В квартире площадью $N+60 \text{ м}^2$ проживают 5 человек. Создать таблицу для расчета стоимости оплаты квартиры с учетом основных ее составляющих (наём, отопление, горячее и холодное водоснабжение, электроэнергия и т.д.), учитывая, что плата за отопление пропорциональна площади квартиры, плата за водоснабжение — количеству жильцов, плата за электроэнергию — начальному и конечному показаниям счётчика. Результат определить при помощи функции автосуммирования. N — индивидуальный номер студента. В 1-ю строку таблицы ввести заголовок.

4. Приблизённо рассчитать значения первой и второй производных функции одного аргумента, как отношения приращений функций к приращениям аргументов. В процессе расчёта использовать автозаполнение ячеек. ($\{z=x^n, 0 \leq x \leq 1\}$; $\{z=x^{1/n}, 0 \leq x \leq 1\}$; $\{z=\sin(nx), 0 \leq x \leq 2\pi\}$; $\{z=\cos(nx), 0 \leq x \leq 2\pi\}$; $\{z=\text{tg}(nx), -\pi/2n < x < \pi/2n\}$; $\{z=\text{ctg}(nx), 0 < x < \pi\}$; $\{z=\ln(nx), 0 < x \leq 1\}$; $\{z=\exp(nx), 0 \leq x \leq 1\}$). Здесь n — индивидуальный номер. Оце-

нить точность приближённого расчёта и проиллюстрировать результаты графически.

5. Создать таблицу для расчёта значений функции 2–х переменных. Построение таблицы произвести с использованием автозаполнения ($z=x^n+y^m$; $z=x^n*y^m$; $z=\text{tg}(x)*\text{ctg}(y)$).

6. Рассчитать величины первых пяти элементов арифметической и геометрической прогрессий, у которых первый элемент равен последней, а шаг – предпоследней цифре номера зачётной книжки.

Тема: Логические функции

1. Лимоны продаются по цене либо Ц1/кг, либо Ц2/шт. Составить таблицу для расчета стоимости покупки, пригодную для обоих случаев. Значения Ц1 и Ц2 ввести соответственно в ячейки А2 и В2, а название единицы измерения – в ячейку С2. В ячейку А3 ввести расчётную формулу. В 1-ю строку таблицы ввести заголовок.

2. Составить таблицу, включающую список должников и срок отдачи долга для каждого из них. Каждый день необходимо: 1) автоматически определять, чей срок возврата долга прошёл; 2) если срок возврата долга прошёл более 10 дней назад, выдать соответствующее сообщение. В 1-ю строку таблицы ввести заголовок.

3. Составить таблицу, которая на основе данных о результатах трех сданных экзаменов определяет качество учёбы (отличник, хороший студент, троечник, неуспевающий).

4. Создать таблицу, позволяющую рассчитывать сумму налога на стоимость имущества, переходящего в порядке дарения. В 1-ю строку таблицы ввести заголовок.

5. Создать таблицу, позволяющую рассчитывать сумму налога на зарплату. В 1-ю строку таблицы ввести заголовок.

6. Создать таблицу для расчета заработной платы по типу табл. 3.17, 3.18. При расчете льгот (Л) учесть, что если совокупный доход за предыдущий период (СД) плюс текущая облагаемая сумма (ОС) не превышает 20 000 руб., то

$$L = \text{МРОТ} + 0.75 \text{МРОТ} * Д,$$

где Д – количество детей.

При $\text{СД} + \text{ОС} > 20\ 000$ льготы не предоставляются.

Подходный налог (ПН) на основную зарплату принять равным 13%. ПН на другие начисления (ДН) при $СД+ОС>50000$ и $ДН>20000$ принять равным 35%, а в остальных случаях 13%.

Выполнить контроль: сумма подходных налогов (ПН) за каждый месяц рассматриваемого периода (года), должна равняться ПН на сумму облагаемых сумм (ОС) за тот же период.

Тема: Формулы массива

1. Решить системы линейных уравнений 5-го порядка с использованием формул Крамера и обратной матрицы. Коэффициенты при неизвестных по главной диагонали принять равными номеру в списке группы по журналу.

Тема: Процедура «Подбор параметра» и построение диаграмм

1. Клиент кладёт в банк A рублей под P процентов в год на срок n лет. В конце каждого года клиент снимает со счёта B рублей.

а) Найти сумму вклада S в конце срока с использованием обычной математической функции

$$S=A(1+P)^n-B((1+P)^{n-1}+(1+P)^{n-2}+\dots+(1+P)^1+1)$$

и с использованием функции **Будущее значение**

БЗ($P, n, B, -A, 0$).

б) Через сколько лет сумма вклада удвоится?

в) Какую сумму нужно положить в банк, чтобы сумма вклада достигла 10000 рублей через 3 года?

г) Построить график зависимости суммы вклада S от срока в годах.

Тема: Процедура «Поиск решения»

1. Измерены превышения h_i по сторонам S_i замкнутого нивелирного полигона (см. таблицу), N - индивидуальный номер.

Отличие суммы превышений от нуля является следствием ошибок измерения.

№	h_i (мм)	S_i (м)
1	$1000+5N$	100
2	$-1000+5N$	150

3	2000-5N	200
4	2000+5N	250
5	-4000-5N	300
Σ	0+5N	1000

Нужно найти такие целочисленные поправки V_i к измеренным превышениям, которые обеспечат равенство нулю суммы исправленных превышений и минимизируют сумму взвешенных квадратов самих поправок.

Целевая функция:

$$R = \sum_{i=1}^5 \frac{1}{S_i} V_i^2 \rightarrow \min$$

Условие:

$$\sum_{i=1}^5 (h + V_i) = 0$$

V_i - целочисленные.

Пример подготовки данных и решения для $N=13$ показан на рисунке.

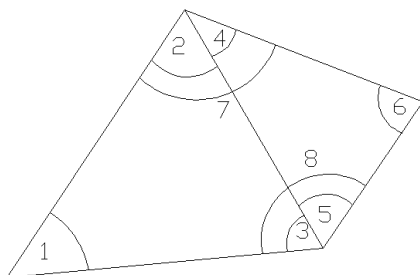
Excel spreadsheet showing a table of data and a "Поиск решения" (Solver) dialog box.

	A	B	C	D	E	F	G	H	I	J	K
1	Уравнивание замкнутого нивелирного пс N=13										
2											
3	Расстояния S _m	100	150	200	250	300					
4	Измер. знач. h _i мм	1065	-935	1935	2065	-4065		Лев. части	Цел. ф-ия.	Прав. част	Нач. знач.
5	Переменные	V1	V2	V3	V4	V5			CVV		N
6	Значения V _i мм	-6	-10	-13	-16	-20			4,2290001		13
7	Веса C _i = 1/S _i	0,010	0,007	0,005	0,004	0,003					
8	Условие	1059	-945	1922	2049	-4085		-9,9E-07		0	
9											
10	Исправл. знач.	1059	-945	1922	2049	-4085		-9,9E-07			

Поиск решения dialog box:

- Установить целевую ячейку: $\$I\6
- Равной: максимальному значению значению: 0 минимальному значению
- Изменяя ячейки: $\$B\$6:\$F\6
- Ограничения:
 - $\$B\$6:\$F\$6 = \text{целое}$
 - $\$N\$8 = \$J\8

2. В четырехугольнике с диагональю измерены 8 внутренних углов β_i (см. рис.). Средняя квадратическая ошибка измерений $m_\beta = \pm 3''$. Значения измеренных углов в градусах, минутах и секундах приведены в таблице. N – индивидуальный номер.



№ углов	Измеренные знач. углов β_i				Исправленные знач. углов β_i				
	G	M	S	В сек.	Поправки в сек.	Углы в градусах	G	M	S
1	60	00	00+N						
2	60	00	00-N						

3	60	00	00+N						
4	30	00	00+N						
5	80	00	00-N						
6	70	00	00+N						
7	90	00	00+N						
8	140	00	00-N						

Математическая модель задачи.

Найти поправки в углы V_i сек., которые обеспечат выполнение геометрических условий в треугольниках и четырехугольнике при минимальной сумме квадратов самих поправок.

Целевая функция:

$$R = \sum_{i=1}^8 V_i^2 \rightarrow \min$$

Условия:

$$(\beta_1 + V_1) + (\beta_2 + V_2) + (\beta_3 + V_3) = 180^\circ$$

$$(\beta_4 + V_4) + (\beta_5 + V_5) + (\beta_6 + V_6) = 180^\circ$$

$$(\beta_1 + V_1) + (\beta_7 + V_7) + (\beta_6 + V_6) + (\beta_8 + V_8) = 360^\circ$$

Требуется:

а) Найти оптимальные поправки V_i .

б) Найти оптимальные целочисленные поправки V_i .

в) Найти оптимальные целочисленные поправки V_i , не превосходящие по модулю утроенной ср. кв. ошибки измеренных углов.

Пример подготовки данных и решения для $N=17$ показан на рис.

Установить целевую ячейку:

Равной: максимальному значению значению: минимальному значению

Изменяя ячейки:

Ограничения:

Микрософт Excel [Режим совместимости] - Microsoft Office Excel

Вид

Структура

Поиск решения

Анализ

Работа с данными

	G	H	I	J	K	L	M						
17													
	внутренние углы												
5	N	углов	G	M	S	Изм.угл.в сек	попр.в сек.	Углы в град.	G	M	S	Контроль	
6		1	60	0	17	216017	-10	60,00194444	60	0	7		
7		2	59	59	43	215983	-4	59,99416667	59	59	39	180	
8		3	60	0	17	216017	-3	60,00388889	60	0	14		
9		4	30	0	17	108017	-3	30,00388889	30	0	14		
10		5	79	59	43	287983	-4	79,99416667	79	59	39	180	
11		6	70	0	17	252017	-10	70,00194444	70	0	7		
12		7	90	0	17	324017	-7	90,00277778	90	0	10		
13		8	139	59	43	503983	-7	139,99333333	139	59	36	360	
14													
15													
16						Поиск решения							
17	змер.углы bi в се	216017,0	215983,0	216017,0	108017,0	287983,0	252017,0	324017,0	503983,0			Лев. части	Прав. части
18	Наимен.перем. V	V1	V2	V3	V4	V5	V6	V7	V8	умма попр.		Цел. функция	
19	Знач.перем. Vi с	-9,00	-4,00	-4,00	-4,00	-4,00	-9,00	-9,00	-7,00	-50,00		356	
20	Козф.цел.ф-ии C	1	1	1	1	1	1	1	1				
21	1 условие	216008,0	215979,0	216013,0								648000	= 648000
22	2 условие				108013,0	287979,0	252008,0					648000	= 648000
23	3 условие	216008,0					252008,0	324008,0	503976,0			1296000	= 1296000
24					В условиях поправки в секундах								
25													

Примечание. Перенос данных из столбца в строку или из строки в столбец выполняется в последовательности:

Копирование (откуда) → (куда) Специальная вставка → Значения → Транспонирование.

3. Между твердыми точками с координатами X_A, Y_A и X_B, Y_B измерены расстояния S и дирекционный угол α .

Нужно найти такие поправки V_s и V_α к измеренным значениям, которые обеспечат выполнение геометрических условий и минимизируют сумму квадратов самих поправок:

$$R = V_s^2 + V_\alpha^2 \rightarrow \min$$

$$(S + V_s) \cos(\alpha + V_\alpha) = X_B - X_A$$

$$(S + V_s) \sin(\alpha + V_\alpha) = Y_B - Y_A$$

Принять X_A, Y_A и X_B, Y_B — произвольно.

$$S = \sqrt{(X_B - X_A)^2 + (Y_B - Y_A)^2} - 2N \text{ см}$$

$$\alpha = \arctg\left(\frac{Y_B - Y_A}{X_B - X_A}\right) + 0,001 * N$$

N — индивидуальный номер.

Пример подготовки данных для варианта $N=13$:

	A	B	C	D	E	F	G	H	I
1									
2									
3		Исходные данные					Измеренные	Твердые	
4	Xa	0		N=	13	S=	565,425425	565,68542	
5	Ya	0							
6	Xb	400				α rad=	0,78409816	0,7853982	
7	Yb	400							
8									
9			Поиск решения						
10									
11	Изм. дир. угол α rad		0,784098						
12	Изм. сторона S	565,4254				Цел. ф-ия			
13	Переменные Vi	Vs	Va		Лев. части		Прав. части		
14	Значения	0,26	0,0013			0,067602			
15	Весы	1	1						
16	Условие 1	400			400		400		
17	Условие 2	400			400		400		
18									

Поиск решения

Установить целевую ячейку:

Равной: максимальному значению значению:

минимальному значению

Изменяя ячейки:

Ограничения:

Тема: Организация динамического обмена информацией между электронной таблицей и текстовым документом

1. Создать в текстовом редакторе Word бланк оплаты квартиры и коммунальных услуг. Создать в табличном процессоре Excel бланк расчёта суммы оплаты. Установить динамический обмен информацией между обоими документами.

Глава 4. Система управления базой данных Access

Условные обозначения и сокращения

При описании процессов работы с программой и при описании выполнения упражнений мы будем использовать следующие условные обозначения и сокращения.

- ЛКМ — левая клавиша мыши (или однократное нажатие на нее).
- 2ЛКМ — двукратное нажатие на левую клавишу мыши.
- ПКМ — нажатие на правую клавишу мыши.
- АЦК — алфавитно-цифровая клавиатура.
- Кн. — кнопка: элемент интерфейса, содержащий название или пиктограмму и имеющий эффект нажатия или включения.
- Кл. — клавиша клавиатуры или мыши.
- ДО — диалоговое окно.
- Ins — клавиша «Insert».
- Del — клавиша «Delete».
- Sh — клавиша «Shift».

Для выделения упражнений используются специальные маркеры например, символ «●».

В описании схем выполнения последовательности команд или операций используется символ «/».

Выполнение выбранной команды или операции инициируется нажатием на ЛКМ или «Enter».

§ 1. Введение

Данная глава служит для поддержки курса «Информатика» при изучении темы, посвященной созданию баз данных (БД) и средств их эксплуатации в среде системы управления базами данных (СУБД) MS Access. Мы будем ориентироваться на систему Access 2007.

Базы данных относятся к так называемым информационным системам (ИС). В основе создания информационных систем лежат такие понятия, как информационная модель (ИМ) и информационный объект (ИО).

Информационная модель – это средство формирования представления о данных, их составе и использовании в определенных условиях.

Основу ИМ составляют **информационные объекты** (далее – **объекты**) и взаимосвязи между ними.

Объектом может быть любой предмет реального мира (организация, человек, земельный участок, культура и т. д.), о котором необходимо хранить нужные сведения в базе данных.

Будем понимать под базой данных именованную совокупность данных, отражающую состояние информационных объектов, их отношений и характеристик.

Наиболее распространенной формой представления данных (описания ИО) является двухмерная таблица. База данных, построенная на основе таких таблиц, называется **реляционной базой данных (РБД)**.

Каждый информационный объект БД представляется в БД с помощью набора качественных и количественных характеристик, которые являются единицами хранения в таблицах БД.

Важным составным элементом базы данных является **поле** (столбец таблицы).

Определим поле как характеристику (или показатель) информационного объекта, которая имеет уникальное имя (название), тип, совокупность свойств и определенное множество значений.

Например, информационным объектом в организации может быть сотрудник. Его характеристики определяются совокупностью полей: табельный номер, фамилия, дата рождения, стаж, оклад и др.

Совокупность значений полей, которые описывают конкретный информационный объект, образует **запись** (строку таблицы).

Совокупность однородных записей, объединенных по смысловому или функциональному принципу, составляет **таблицу БД**.

Характерные особенности табличной организации представления данных состоят в следующем.

- Каждая таблица БД имеет строки и столбцы. Строка таблицы – это запись. Столбец таблицы — это — поле. Таким образом,

строка таблицы определяет некоторые характеристики информационного объекта, а столбец – множество текущих значений определенной характеристики разных объектов.

- Каждый столбец имеет уникальное имя (имя поля) и однороден по структуре (тип информации, разрядность, точность).
- Таблицы, составляющие базу данных, могут быть связаны различными отношениями, устанавливающими взаимосвязь (логическую, функциональную или смысловую) между родственными или одинаковыми (по свойствам) полями разных таблиц.

Таблицы БД не используются для расчетов (и не содержат формул), они служат только для хранения (исходных) данных, подмножества которых могут быть использованы разными пользователями для решения своих задач разного класса (статистических, информационных, математических и др.).

Введем еще одно очень важное понятие, выделяющее некоторые поля таблицы в особую категорию.

Выражение (арифметическое, логическое, текстовое), построенное с использованием полей и служащее для сортировки, фильтрации, поиска и выборки информации при запросах, а также для организации связей между таблицами БД, назовем **ключевым (или ключом)**. В качестве ключа может использоваться и отдельное поле. Особое значение имеет ключ, который называется **первичным ключом**. Этот ключ всегда присутствует в любой таблице и служит для однозначной идентификации каждой записи. Если разработчик забывает определить такой ключ, то система это делает сама, добавляя специальное поле «Код», куда записывает порядковый номер записи таблицы.

Для создания БД и средств их эксплуатации используются специальные системы, которые называются **системами управления базами данных (СУБД)** и представляют собой совокупность языковых и программных средств, обеспечивающих разработчика и пользователя БД необходимым набором соответствующих инструментов.

Возможности и характеристики СУБД MS Access

Система управления базами данных Microsoft Access предназначена для создания и эксплуатации баз данных реляционного типа. Это достаточно полная по функциональным характеристикам реляционная СУБД с развитыми средствами создания БД, удобными элементами их

эксплуатации (отчетами, экранными формами, запросами и т.д.) на уровне визуального проектирования.

MS Access позволяет определить в одной записи поля разных типов.

- Текстовые поля используются для хранения алфавитно-цифровых данных длиной до 255 символов.
- Числовые поля предназначены для хранения числовых данных целого и вещественного типов.
- Поля типа «Дата» и «Время» служат для хранения календарных дат и значений времени.
- Счетчики — поля с автоматической генерацией целых чисел натурального ряда, они используются в основном при вводе новых записей или в качестве первичного ключа. Таблица может содержать только одно поле этого типа.
- Поле примечаний (Мето) — специальный тип поля для хранения больших объемов текстовой информации.
- Существуют и другие типы полей.

Каждое поле можно наделить определенными свойствами, которые делятся на две категории: «Общие» и свойства «Подстановки». Состав свойств зависит от типа поля. Более подробно мы рассмотрим эти свойства позже.

MS Access позволяет переносить информацию в текстовый редактор Word или преобразовывать ее в форматы других СУБД

Для создания сложных приложений, позволяющих автоматизировать процесс эксплуатации БД, в среде СУБД MS Access имеется набор макрокоманд и встроенная система программирования на языке Visual Basic (VBA).

Создавать БД, таблицы и элементы управления можно:

- с помощью специальных программ — «Мастеров», которые самостоятельно определяют алгоритм вашей работы;
- используя шаблоны «стандартных» БД и других элементов;
- используя образцы (при создании таблиц, запросов, форм и отчетов);
- самостоятельно (в режиме конструктора).

Управление внешним видом окон и изображением в них элементов, отображение и настройка панелей инструментов, установка об-

щих параметров и режимов работы системы выполняется аналогично другим приложениям Windows через меню «Вид», «Сервис», «Окно».

§ 2. Технология построения реляционной базы данных

При создании БД можно выделить следующие стадии.

1. Проектирование базы данных с созданием информационной модели (например, в виде схемы взаимосвязанных таблиц).
2. Создание БД и средств ее эксплуатации в среде СУБД.
3. Тестирование и документирование процесса работы с БД и ее элементами.

В результате проектирования должна быть создана специальная информационная реляционная модель данных, отвечающая определенным требованиям.

- Каждое поле таблицы БД должно иметь уникальное имя, определенный тип и набор определенных свойств.
- Все записи в таблице должны быть уникальными. Этого всегда можно добиться, если в таблице определить первичный ключ, который может состоять из отдельного поля или совокупности полей, объединенных каким либо выражением (арифметическим, текстовым, логическим). Значение первичного ключа не должно иметь повторений в таблице.
- Все поля таблицы (кроме первичного ключа) должны быть независимы. Если Вы изменили значение одного поля, это не должно сказываться на значении других полей. Исключение могут составлять вычисляемые поля в таблицах запросов, экранных формах и отчетах.
- При проектировании многотабличной БД поля, по которым осуществляется связь, должны иметь одинаковые имена и свойства (но не обязаны входить в первичный ключ).

Для удовлетворения перечисленных требований используют специальную технологию **нормализации ИМ**. Первоначально ИМ формируется на основе обобщения внешних представлений разных пользователей будущей БД.

В процессе нормализации элементы данных группируются в таблицы, представляющие собой информационные объекты и их взаимосвязи (отношения).

Этот процесс рассмотрим на примере ИС для автоматизации ведения учета и оценки качественных и количественных показателей садово-огородных культур на участке личного пользования.

Замечание.

Здесь приводится упрощенный вариант описания процесса нормализации. Более полное описание можно найти в специальной литературе по проектированию баз данных.

Проектирование БД «Садово–огородные культуры»

Допустим, пользователю БД необходимо иметь (локальную) ИС, обеспечивающую решение различных информационных задач, связанных с выращиванием культур на его участке.

Такая база данных должна предоставить:

- информацию о фактическом составе и характеристиках садово-огородных культур;
- количественную информацию об урожайности (продуктивности) культур;
- информацию для своевременного сбора урожая (сроки созревания);
- информацию для оценки качественного состояния почвы (например, кислотность);

Кроме того, культуры должны быть разбиты на группы (классы) по видовому признаку и способам выращивания (деревья, кустарники и т. д.). Необходимо предусмотреть возможность хранить поля с некоторыми характеристиками групп и по мере эксплуатации расширять их состав.

В данном варианте базы данных мы не приводим эти поля (используем только поле «Единицы измерения»).

БД и средства ее эксплуатации должны обеспечить решение следующих задач.

1. Создать условия надежного ввода, хранения и корректировки информации о культурах, которые размещаются или могут быть размещены на участке.

2. Обеспечить при отображении информационных массивов, содержащих характеристики культур с делением культур на группы.
3. По требованию пользователя выводить данные о культурах, которые имеют определенные значения заданных характеристик.
4. Обеспечить вывод на печать полного списка культур, размещенных на участке, с подсчетом их количества или площади, занимаемой отдельными группами культур.

Очевидно, что данный список задач может быть продолжен.

В соответствии с общей схемой проектирования БД мы должны тщательно проанализировать задачи и запросы пользователя и на основе этого анализа определить из каких информационных массивов будет состоять база данных, по каждому массиву определить структуру, состав и тип полей, установить взаимосвязь массивов и т. д.

Допустим, что после обсуждения с пользователем состава и характеристик БД мы определили два массива: массив «Группы», содержащий общие характеристики групп и массив «Культуры», содержащий характеристики культур. После исключения одинаковых полей и объединения всех полей из обоих массивов мы получили исходную реляционную информационную модель, которую можно представить в виде таблицы 4.1 «Культуры участка». Из методических соображений мы не стали включать набор общих характеристик групп (например, общее количество единиц представителей группы на участке, общие технологические параметры возделывания культур группы и др.), которые пользователь сможет добавить в процессе эксплуатации базы данных.

С точки зрения реляционной теории данная ИМ имеет ряд недостатков (дублирование данных, взаимозависимость некоторых полей, наличие вычисляемого поля). Кроме того использование текстовых названий групп и культур для идентификации записей может вызвать определенные неудобства при решении ряда задач обслуживания и эксплуатации БД.

Для устранения недостатков используем **процесс нормализации**.

Нормализация должна определить такой набор таблиц и их взаимосвязи, который обладает лучшими свойствами по сравнению с другими вариантами при выполнении операций добавления, корректировки и удаления данных, а также обеспечивает максимальное быст-

родействие при запросах и требует минимального расхода электронной памяти.

Нормализация состоит из ряда этапов, в результате которых создаются информационные модели разного уровня.

Такие модели называются **нормальными формами**.

Первый этап нормализации заключается в образовании таблицы, содержащей все необходимые поля информационной модели и ключевых полей. При этом необходимо исключить повторяющиеся столбцы и столбцы-массивы (содержащие значения типа списка). Фрагмент таблицы 4.1 представляет ИМ в первой нормальной форме. Однозначно определять записи будут поля «Название культуры» (далее – «Культуры») и «Группа» (т.е. мы имеем составной ключ). Все остальные поля будут функционально зависеть от этих двух.

Второй этап нормализации заключается в выделении полной и частичной зависимости неключевых полей от первичного ключа.

В нашем случае мы имеем частичную зависимость от составного первичного ключа полей: «Сроки созревания», «Почвы», «Количество культур», «Урожай» и «Урожайность». Значения этих полей зависят от поля «Культуры» и не зависят непосредственно от поля «Группы». А поле «Ед. измерения» зависит от поля «Группы».

Для решения проблем зависимости полей выделим из исходной таблицы поля связанные с группировкой культур, и объединим их в отдельную таблицу (информационный объект), которую назовем «Группы».

Определим поля «Название группы» и «Ед. измерения». Добавим поле «Обозначение», которое будем использовать в некоторых запросах и отчетах при выводе информации (на печать или экран).

Для связывания таблиц добавим в таблицу специальное числовое поле «Код_Гр», что позволит (без ущерба удобству эксплуатации) значительно быстрее выполнять запросы, экономить память (особенно во второй таблице) и упростит ввод данных.

Структура таблицы «Группы» и некоторые данные приводятся в таблице 4.2. Ключевым столбцом в этой таблице будет «Код_Гр».

Таблица 4.1

Фрагмент исходной таблицы «Культуры участка»

Группа	Ед. изм.	Название культуры	Срок созр. От	Срок созр. До	Почвы от	Почвы до	К-во в тек. году	Урожай в тек. году (КГ)	Урожайность Кг/ед.изм
1	2	3	4	5	6	7	8	9	10
Деревья	Дер.	Яблоня зимняя	09.25	10.10	5,5	6,0	1,0	20,0	20
Деревья	Дер.	Яблоня Летняя	08.01	08.25	5,5	6,0	2,0	30,0	15
Кусты	Куст	Смородина красная	07.15	07.25	5,6	6,0	2,0	5,0	2.5
Кусты	Куст	Смородина черная	07.20	07.25	5,6	6,0	4,0	8,0	2
Овощные однолет	Кв.м.	Картофель	08.10	09.15	6,0	6,5	100,0	250,0	2.5
Овощные однолет	Кв.м.	Томаты ранние	08.15	08.25	6,3	6,8	40,0	30,0	0.75

Замечание.

Названия столбцов в постановке задачи (в информационных моделях) могут отличаться от имен полей в таблицах базы данных.

Для хранения характеристик культур создадим таблицу «Культуры». В нее включим все поля, зависящие только от «культуры» и добавим необходимое поле для связывания таблиц в единую информационную модель.

Для связи таблиц будем использовать числовое поле «Код_Гр.» Это поле будет использоваться и для группировки культур при выполнении запросов и в отчетах БД.

Таблица 4.2.

Содержание таблицы «Группы»

Код гр.	Название	Обозначение	Ед. измерения
1	Деревья	ДЕР.	дер.
3	Ягодные	ЯГОДН.	кв.м.
2	Кусты	КУСТ.	куст.
6	Цветочные	ЦВЕТЫ	кв.м.
4	Овощные двухлетние	ОВОЩ_Д	кв.м.
5	Овощные однолетн.	ОВОЩ_О	кв.м.

Так как для оценки количественных и качественных показателей возделывания культур необходимо иметь единицы измерения оставим это поле в таблице «Культуры». Это позволит решать некоторые задачи, не обращаясь к таблице «Группы».

Для удобства идентификации культур вместо поля «Название» будем использовать специальное текстовое поле «Код культуры». Значения этого поля будут иметь простую mnemonic форму и контролироваться с помощью специальной маски и другими свойствами, что снизит уровень ошибок при вводе данных и формировании запросов. Использование числового кода в данном случае нерационально, т.к. в отличие от таблицы «Группы», таблица «Культуры» будет постоянно корректироваться, а пользователю запомнить числовые значения при большом количестве культур значительно сложнее чем mnemonicские.

Первичным ключом в этой таблице будет составной ключ, состоящий из значений полей «Код_Гр.» и «Код_культ».

В результате преобразований получим две взаимосвязанные таблицы **во второй нормальной форме** (структура здесь не приводится), где все неключевые поля зависят только от ключевых.

На третьем этапе нормализации мы должны исключить зависимые (неключевые) поля. В нашем случае, значения поля «Урожайность» вычисляются на основе значений полей «Кол-во в тек. году» и «Урожай в тек. году». Этого требует не только теория нормализации, но и особенности СУБД Access, которая не поддерживает вычисляемые поля в таблицах БД.

При необходимости мы можем включить это поле в отчеты, запросы, экранные формы и другие объекты БД.

После исключения зависимых полей мы получим информационную модель **в третьей нормальной форме**.

Окончательный вид таблицы «Культуры» представлен в таблице 4.3.

Первая графа («№ записи») никакого отношения к нашей базе данных (и таблице) не имеет. Пояснения к другим полям таблицы мы приведем позже.

Таблица 4.3

Пример содержимого таблицы «Культуры»

№ записи	Код гр.	Код культ.	Название	Созр. От	Созр. До	Почвы от	Почвы до	Ед. изм.	К-во в Тек. Году	Урожай в тек. году (КГ)
1	2	3	4	5	6	7	8	9	10	11
1	1	ЯБЛ.–ЗИМН	Яблоня зимняя	09.25	10.10	5,5	6,0	Дер.	1,0	20,0
2	1	ЯБЛ.–ЛЕТН	Яблоня Летняя	08.01	08.25	5,5	6,0	Дер.	2,0	30,0
3	1	ЯБЛ.–ОСЕН	Яблоня осенняя	09.10	09.20	5,5	6,0	Дер.	2,0	15,0
4	1	ГРУ.–ОСЕН	Груша осенняя	09.20	09.15	5,4	6,2	Дер.	2,0	10,0
5	1	ВИШН.–ОБЩ	Вишня	07.10	07.20	6,2	7,0	Дер.	3,0	15,0
6	2	СМОР.–КРАС	Смородина красная	07.15	07.25	5,6	6,0	Куст	2,0	5,0
7	2	СМОР.–ЧЕРН	Смородина черная	07.20	07.25	5,6	6,0	Куст	4,0	8,0
8	5	КАРТ.–ОБЩ.	Картофель	08.10	09.15	6,0	6,5	Кв.м.	100,0	250,0
9	5	ТОМ.–РАН.	Томаты ранние	08.15	08.25	6,3	6,8	Кв.м.	40,0	30,0
10	5	ТОМ.–ПОЗД	Томаты позднеспелые	08.25	09.20	6,2	6,9	Кв.м.	30,0	50,0
11	4	МОРК.–ОБЩ	Морковь	07.10	09.20	6,0	7,0	Кв.м.	5,0	50,0
12	4	КАП.–Б.Р.	Капуста белокочанная ранняя	07.20	08.10	6,5	7,2	Кв.м.	20,0	50,0
13	4	КАП.–Б.П.	Капуста белокочанная, поздняя	08.20	09.20	6,6	7,7	Кв.м.	20,0	80,0
14	4	КАП.–ЦВЕТ	Капуста цветная	08.10	08.30	6,7	7,3	Кв.м.	5,0	30,0

§3. Создание базы данных

Общая схема создания БД в СУБД MS Access

При создании элементов БД мы воспользуемся разными способами (мастерами, конструкторами, образцами и смешанными).

Замечание.

Ранее мы использовали термин «объект» для сокращения названия «информационный объект». Далее мы будем использовать аналогичный термин и для специальных объектов СУБД. **Объектами в СУБД** называется все то, что может иметь имя: таблицы, запросы, отчеты, формы, макросы, модули.

Можно предложить следующую схему создания БД и ее элементов в среде СУБД MS Access.

1. Создание таблиц. Опытный разработчик может сразу начинать создание таблиц в режиме конструктора. Начинающему мы рекомендуем сначала создать одну таблицу (например, «Группы») по образцу тех данных, которые будут храниться в таблице, а затем откорректировать свойства полей в режиме конструктора.

2. После создания всех таблиц БД можно установить структурные связи (создать схему данных). Лучше это делать, когда таблицы пусты или содержат хорошо выверенные данные.

3. Затем следует заполнить таблицы данными из контрольного примера и проверить работу связей и свойств (масок, условий на значения, установки по умолчанию, синхронизацию работы связанных таблиц при удалении данных и т. д.). Далее, если необходимо, следует исправить ошибки и откорректировать структуру таблиц и связей.

4. Следующий этап - создание экранных форм для ввода и редактирования записей таблиц БД. На начальном этапе лучше воспользоваться мастером форм. После создания основы — форма корректируется в режиме конструктора и наполняется функциональными элементами.

5. Самый трудный этап — создание запросов. Здесь необходимо иметь формализованные требования на выборку данных из БД и хорошее знание о построении логических выражений.

Запросы позволяют объединять данные из разных таблиц и представлять их в одной таблице в нужном формате. Запросы создаются с помощью мастера, конструктора или специального языка запросов (SQL).

6. Для удобного представления «объемной» информации, содержащей сгруппированные, отсортированные и итоговые данные отдельных таблиц и запросов или объединенные данные из разных источников, необходимо создавать отчеты. Технология создания отчетов определяется их структурой и может быть выполнена аналогично созданию форм.

Создание основного файла БД

Для обеспечения надежной сохранности результатов выполнения упражнений организуйте на своей дискете (в системе Windows) папку (например, «БД_Культуры») и постарайтесь придерживаться следующей схемы.

В начале сеанса переписывайте все созданные файлы БД со своего диска из созданной папки на жесткий диск (например, в папку «Мои документы»).

Работайте (открывайте БД и т.д.) на жестком диске.

В конце сеанса перепишите файл БД на свою дискету в нужную папку.

Создадим базу данных.

- Любым способом запустите систему MS Access. Появится окно системы. В левой части находится колонка категорий шаблонов, где можно выбрать нужный шаблон для БД. Справа – перечень баз данных, с которыми работали пользователи в предыдущих сеансах.

Для открытия уже созданной базы данных надо нажать кн. «Другие».

Для создания новой базы данных – щелкните ЛКМ на значке «Новая база данных». В ДО «Новая база данных» укажите место хранения и имя файла базы данных. **Имя файла должно быть уникальным.** По умолчанию система устанавливает тип .accdb.

Замечание.

Для работы с файлами можно воспользоваться круглой кнопкой «Office» в верхнем левом углу окна системы. В процессе ра-

боты используйте эту кнопку для регулярного сохранения файла после внесения изменений. Для открытия существующего файла используйте команду «Открыть».

Учтите, что в Access нет команды «Сохранить как...» для переименования или перезаписи базы данных на другой диск. Эта команда действует только на отдельные объекты БД (таблицы, формы, запросы и др.).

- После установки имени файла нажмите кн. «Создать».

Внутри главного окна СУБД появятся окна «Область переходов» с именем одной таблицы и рабочее окно таблицы (пустой).

Окно системы состоит из стандартных (для Windows) элементов.

- В верхней части расположен заголовок с набором стандартных кнопок.
- Главное меню (**ГМ**) системы – «лента»— используется для выполнения всех команд и операций с помощью кнопочных панелей инструментов (**ПИ**), расположенных на **вкладках**, которые разделены на функциональные группы.
- Внизу расположена строка состояния, которая содержит общую справочную информацию, текущие подсказки и сообщения об ошибках. В нижнем правом углу находятся кнопки управления режимами просмотра объектов БД.

Центральная часть окна СУБД служит для отображения разных функциональных окон. Наиболее важными из них являются.

- **Окно «Области переходов» (ОПЕР).**

Это аналог окна базы данных в предыдущих версиях. Оно расположено справа. Появляется сразу при создании или открытии БД. По мере создания объектов БД в этом окне отображается их список, разделенный на группы.

Следующие окна открываются (в разных режимах) при выполнении соответствующих команд или при нажатии на соответствующую кнопку в окне базы данных.

- **Окно таблицы.** В этом окне отображается содержимое указанной таблицы БД доступное для ввода и редактирования данных, или структура таблицы (в режиме конструктора).

- **Окно запросов** служит для построения запросов на выборку информации из БД по определенному критерию.
- **Окно формы** используется для обработки записей БД в удобной форме типа «формуляр».
- **Окно отчета.** Если оно открывается в режиме предварительного просмотра, то показывает (в разных масштабах) вид сформированного документа для печати. Изменить форму, программу формирования и структуру отчета можно в режиме конструктора.

Другие окна открываются по мере выполнения определенных команд или операций с данными.

Создание таблицы «Группы».

При создании новой БД, если параметры рабочей среды установлены по умолчанию открывается окно пустой таблицы в режиме «Таблицы» (см. рис. 4.1)

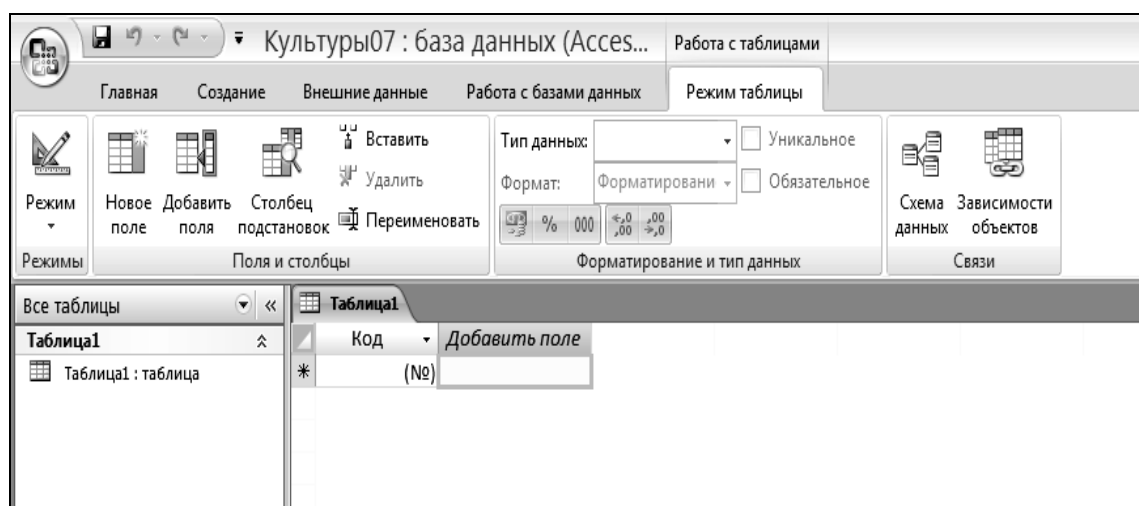


Рис. 4.1. Начальное состояние окна таблицы

Из методических соображений мы будем создавать таблицу «Группы» «по образцу», т.е. введем в каждое поле данные из таблицы 4.2 и система сама установит тип поля (столбца) и некоторые его характеристики.

Сейчас таблица имеет условное имя «Таблица1». Позже мы дадим ей конкретное имя - «Группы».

По умолчанию система сформировала таблицу с одним полем «Код», присвоив ему статус первичного ключа. Справа от него

находится вкладка «Добавить поле». Значения в поле «Код» будут формироваться автоматически.

Введите в столбец «Добавить поле» значение = 1 (код группы «Деревья») и нажмите Enter. Появится новое поле «Поле1» со значением = 1. По этому значению система определит тип поля – «числовой». Значение поля «Код» будет установлено = 1.

Используя контрольный пример из таблицы 4.2 – «Группы», введите две первые записи, строго соблюдая регистр (прописные/строчные буквы) и знаки препинания. В конце ввода значения нажимайте Enter или Tab. Для установки курсора в начало следующей записи используйте стрелки или курсор мыши.

После ввода первых записей фрагмент таблицы может выглядеть следующим образом.

Код	Поле1	Поле2	Поле3	Поле4	Добавить поле
1	1	Деревья	ДЕР.	дер.	
2	3	Ягодные	ЯГОДН.	кв.м.	

Внимание !

Если при вводе в числовые поля разных записей вы допустите ошибку, меняющую тип поля, то система автоматически определит для него тип «Текстовый». Если вы не исправите такую ошибку и сохраните таблицу, то исправить тип можно будет только в режиме конструктора.

После ввода начальных образцов записей необходимо переименовать столбцы в соответствии с макетом из графы «Имя поля» таблицы 4.4.

- Для изменения имени поля щелкните на его заголовке ПКМ и выберите команду «Переименовать столбец». После этого введите новое название из таблицы 4.4. Например, вместо «Поле1» введите «Код_Гр».
- Любым способом переименуйте остальные поля.
- Для того, чтобы поля имели размеры, соответствующие размерам своих значений, проводите курсором мыши по строке заголовков полей и, как только курсор примет форму крестика с двунаправленными стрелками, щелкните 2ЛКМ. Размер столбца изменится.

Сохраните макет таблицы с данными. Для этого введите команду «Сохранить» и в отрывшемся окне «Сохранение» введите

имя таблицы: «Группы». Чтобы столбец «Добавить поле» не отображался на экране, выделите его, откройте контекстное меню (ПКМ) и введите команду «Скрыть столбцы». После проделанных операций основная часть таблицы примет следующий вид.

Код	Код_Гр	Назв	Обозн	Ед_изм
1	1	Деревья	ДЕР.	дер.
2	3	Ягодные	ЯГОДН.	кв.м.

Таблица 4.4

Типы и описание полей таблицы «Группы»

Имя поля	Тип данных	Описание
1	3	4
Код_Гр	Числовое	Код группы культур
Назв	Текстовое	Название группы культур
Обозн	Текстовое	Краткое обозначение группы
Ед_изм	Текстовое	Ед. Измерения затрат

§ 4. Установка и исправление свойств полей

Для того чтобы убедиться в правильности определения характеристик полей и добавить или откорректировать свойства полей, необходимо воспользоваться специальным **режимом конструктора**.

Перейти в режим конструктора (в случае, если таблица уже создана) можно одним из следующих способов:

- если таблица открыта в режиме таблицы - нажать кнопку «Режим», расположенную в левом верхнем углу и выбрать «Конструктор таблиц»;
- если таблица закрыта, то в окне области переходов надо выделить имя таблицы нажать ПКМ и ввести команду «Конструктор».
- Воспользуйтесь любым способом для перехода в режим конструктора.

На экране вместо таблицы появится диалоговое окно конструктора (рис. 4.2). Обратите внимание, что на ленте появится новая вкладка «Конструктор».

Замечание.

Вид окна может немного отличаться от представленного на рисунке. Поле «Код» уже удалено

В верхней части окна конструктора находится бланк определения полей таблицы, который разбит на три графы, где указывается имя поля (столбца), тип данных и краткое описание для вывода в строку состояния краткой справки.

Как только мы определим имя поля и его тип, в нижней части окна активизируются списки свойств. Для нашего примера некоторые свойства полей уже определены. Активная (текущая) строка бланка таблицы помечена символом « ▸ ». В нижнем правом углу выводится краткая информация о типах полей и свойствах.

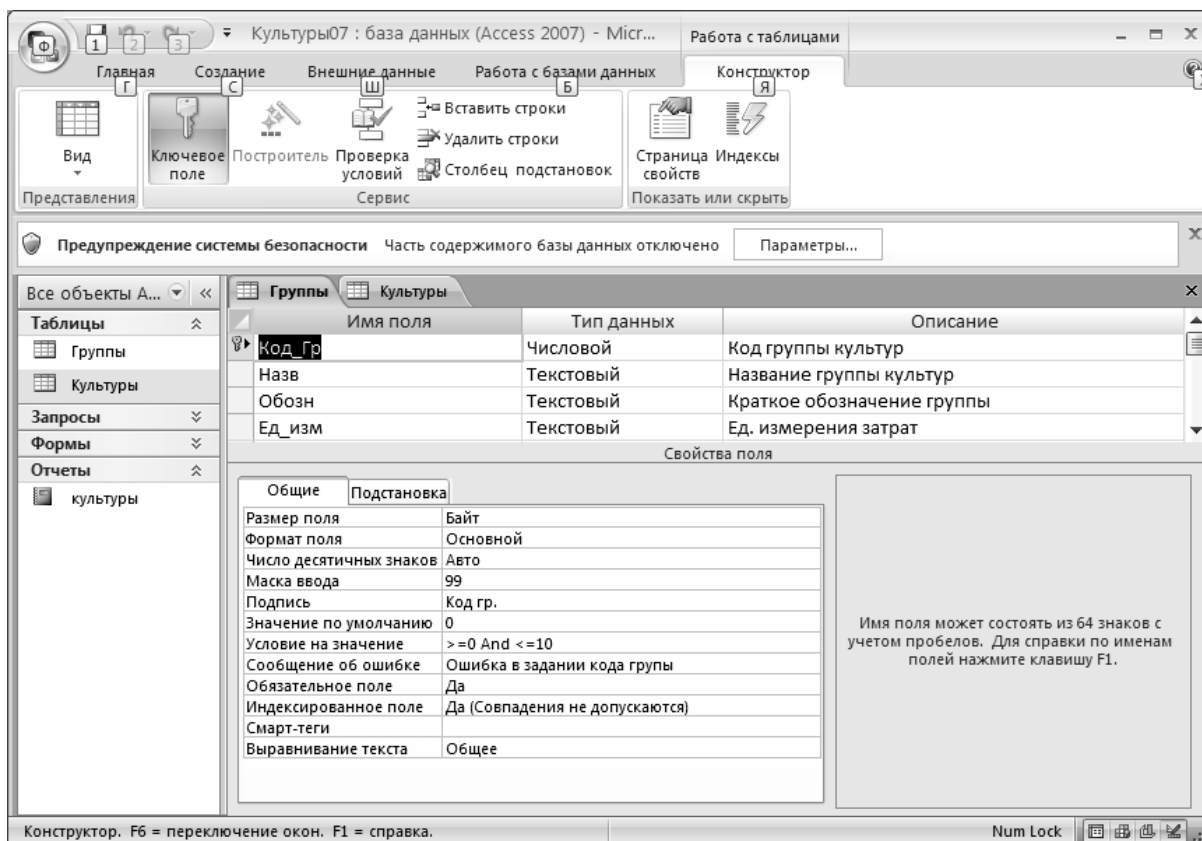


Рис. 4.2. Вид окна конструктора

В первой строке бланка находится поле «Код», которое система создала сама и присвоила этому полю статус «Первичный ключ» (он отображается в соответствующей строке бланка). Это поле надо удалить.

- Выделите строку с этим полем (ЛКМ на кн. слева от имени поля), нажмите кн. Del. и подтвердите в следующем запросе системы свои намерения.

Нашей следующей задачей является подробное определение свойств всех полей таблицы «Группы».

Если вы правильно ввели образцы, то типы полей уже определены. Нам надо дополнить или подкорректировать значения столбцов верхнего бланка в соответствии с данными таблицы 4.4.

Замечание.

Для определения типа данных поля надо щелкнуть ЛКМ на треугольнике в соответствующей клетке таблицы бланка и выбрать из списка нужный тип.

Прежде чем приступать к описанию свойств полей, определим «первичный ключ», т. к. тот, который создала система, мы удалили. В таблице «Группы» первичным ключом будет поле «Код_Гр».

- Установите указатель на это поле и на вкладке «Конструктор» нажмите кн. «Ключевое поле».

Теперь нам надо определить свойства полей, которые задаются на вкладках в нижней части окна.

Перечисленные ниже общие свойства определяют.

- Размер поля — количество символов (знаков) для хранения значений.
- Формат поля — задает формат отображения данных при выводе на экран или печать.
- Число десятичных знаков (для числового поля) - устанавливает количество знаков после десятичной точки.
- Маска ввода — устанавливает шаблон ввода данных. Используется для качественного контроля данных сложной структуры (телефонных номеров, почтовых индексов и т.п.) при вводе.

- Подпись – задает название столбца БД, которое будет отображаться в режиме «Таблица» вместо имени поля.
- Значение по умолчанию – устанавливает то значение поля, которое будет автоматически установлено при вводе новой записи.
- Сообщение об ошибке – определяет текст, который надо вывести при нарушении условий контроля при вводе исходных данных.
- Условие на значение — определяет количественные условия, по которым система будет контролировать ввод исходных данных.
- Свойство «Обязательное поле». Значение «да» требует обязательного ввода значения в поле, иначе система «не выпустит» вас из поля или записи до тех пор, пока вы не введете правильное значение.

Установка остальных свойств здесь не рассматривается.

Категория свойств «Подстановки» позволяет определить особые свойства поля как объекта управления. В этом случае поле снабжается списком, который может формироваться автоматически по значениям полей из другой таблицы, или из сформированного в ручную списка, или из запроса, или из других источников. Мы рассмотрим использование этого свойства при создании некоторых полей второй таблицы — «Культуры».

При установке курсора в строку определенного свойства справа может появиться кнопка с треугольником или многоточием. Треугольник открывает список возможных вариантов значения поля, а многоточие запускает «мастера» построителя выражений или формирования значения свойства.

Переход из верхнего бланка в нижний и обратно выполняется клавишей F6 или мышью.

Перед установкой свойств полей внимательно прочитайте следующие замечания.

О масках ввода.

Маски ввода (для поля) - это набор символов, каждый из которых указывает системе, какой тип данных может быть указан в данной позиции поля.

Маска состоит из трех частей, разделенных точкой с запятой.

Первая часть — собственно маска ввода.

Вторая часть определяет, нужно ли сохранять в значении поля указанные постоянные символы (разделители). Если нужно, то следует указать 0, иначе — 1.

Третья часть определяет символ — заполнитель маски (по умолчанию «_»).

В таблице 4.5 приведены основные коды маски, которые можно использовать в разных типах данных.

Таблица 4.5

Коды, используемые в масках ввода

Вид символов при вводе или описание	Обязательный код	Необязательный код
1	2	3
Только цифры от 0 до 9	0	9
Цифры, пробел и знаки + или —	нет	#
Буквы (любые)	L	?
Только буквы и цифры	A	A
Любые символы (в т. ч. И пробел)	&	C
Место десятичной точки или запятой	.	
Преобразование символов справа к нижнему регистру	<	
Преобразование символов справа к верхнему регистру	>	

В верхней части таблицы приводятся коды маски символов. Ниже указаны коды управления преобразованием символов. Обязательный код требует обязательного ввода соответствующего символа. Необязательный код позволяет переместить курсор ввода в другую позицию без занесения символа в текущую позицию.

Пример.

Допустим, что нам надо вводить сложный код XXX–PPP–NN, где XXX — любые (необязательные символы), PPP — обязательные буквы, NN — обязательные цифры. Все символы должны находиться в верхнем регистре. Разделители следует хранить со значением поля. Тогда маску для ввода этого кода можно задать следующим образом:

«>CCC–LLL–00;0;_».

Замечание.

В этом примере, вторая и третья части маски указаны из методических соображений. Все символы набиваются в латинском регистре.

Организация контроля при вводе данных (свойства: «Значение по умолчанию» и «Условие на значение»)

Установка маски не решает всех проблем, связанных с надежным вводом значений полей. MS Access предоставляет дополнительные возможности обеспечения контроля исходных данных.

Вы можете:

- установить начальное значение поля при вводе новой записи (значение по умолчанию);
- установить признак обязательного ввода значения поля (обязательное поле). В этом случае система не выпустит вас из поля, пока вы не введете туда значение;
- задать логическое выражение для проверки вводимого значения (установить условие на значение) и определить текст сообщения при ошибке.

Простые условия проверки значений формируются из операторов сравнения и значений (операндов). К операторам сравнения относятся знаки: «<» — меньше, «>» — больше, «>=» — больше или равно, «<=» — меньше или равно, «<>» — не равно.

Сложные условия формируются с помощью логических операторов AND (И) и OR (ИЛИ).

Например, если значение поля должно быть равно 0 или изменяться от 100 до 200, но не равно 150, то условие будет выглядеть так (имя поля в этих выражениях не указывается):

$=0 \text{ OR } > 100 \text{ AND } < 200 \text{ AND } <> 150$.

Ниже приводится таблица 4.6 с характеристиками полей (свойствами) для таблицы «Группы». Некоторые свойства полей в этой таблице и в последующих таблицах такого рода приводятся в условных обозначениях.

«Размер и Формат» – может быть указан в виде перечисления трех компонент (Р, Ф, ЧДЗ), где:

Р — размер поля (один из стандартных: Б — байт, Ц — целое, ДЦ — длинное целое, ПЛТ — с плавающей точкой);

Ф — формат поля (для чисел мы используем либо основной формат (ОФ), либо фиксированный (ФК), либо с разделителем разрядов (РР));

ЧДЗ — число десятичных знаков (используется для описания числовых полей). Система по умолчанию назначает два знака.

Для целых чисел (байт, целое и длинное целое) компоненты Ф и ЧДЗ могут не указываться или указывается «Авто», как установила сама система.

Для текстовых данных указывается только размер поля (количество символов).

Таблица 4.6

Свойства полей таб. «Группы»

Имя поля	Размер и формат	Маска ввода	Подпись	Значен. по ум.	Условие на значен.	Обяз. поле
1	2	3	4	5	6	7
Код_Гр	Р = байт, Ф = основной ЧДЗ = Авто	99	Код гр.	0	>= 0 and <= 10	Да
Назв	Р = 20		Назв. Гр.			Нет
Обозн	Р = 6	>СССССС	Обозн.гр.			Да
Ед_изм	Р = 5		Ед. измер.			Нет

- Используя данные этой таблицы, заполните все необходимые позиции бланка свойств полей БД. Тексты «Сообщений об ошибке» для контролируемых полей придумайте самостоятельно. Правильное смысловое содержание сообщений послужит оценкой уровня Вашего отношения к данной работе и придаст ей индивидуальный характер.

Ввод данных в таблицу «Группы»

После внесения изменений в свойства таблицы нам надо продолжить ввод данных.

- Для этого перейдите в «Режим таблицы».
 - Перед выводом таблицы система запросит сохранение макета. Подтвердите кнопкой «Да» сохранение изменений.
- Приведем несколько замечаний по работе с таблицей.

Текущая запись и текущий столбец обозначены слева и сверху цветными метками, а номер записи приводится в нижней части окна таблицы. Там же расположены кнопки навигации, которые позволяют перемещаться по записям таблицы, кнопка добавления новой записи «▶*», полоса горизонтальной прокрутки полей записи и другие сообщения.

В статусной строке окна MS Access приводится дополнительная информация о режимах работы и расшифровка содержания поля, которую мы определили при создании таблицы.

Слева вместо маркера текущей записи может находиться маркер новой записи «*».

Перемещение по таблице может быть выполнено мышью — установкой на нужное поле, мышью с помощью линеек прокрутки, клавиатурой (стрелками, PgUp, PgDn).

Замечания.

1. Если вы допустили ошибки при вводе данных, то лучше (для данной таблицы) сотрите ошибочное значение и введите новое.

2. Для отмены неправильного действия можно воспользоваться кнопками отмены.

- Ведите в таблицу «Группы» остальные записи из таблицы 4.2. Поведение курсора при нажатии на клавишу Enter после ввода значения в поле зависит от настроек рабочей среды и может отличаться от текущего.
- Закройте таблицу сохранив изменения и откройте ее вновь. Система должна отобразить данные уже в отсортированном по коду группы виде (см. таблицу 4.7).

Таблица 4.7

Состояние таблицы «Группы» после ввода записей

Код гр.	Назв. Гр.	Обозн. Гр.	Ед. Измер.
1	Деревья	ДЕР.	дер.
2	Кусты	КУСТ.	куст.
3	Ягодные	ЯГОДН.	кв.м.
4	Овощные двухлет-	ОВОЩ_Д	кв.м.
5	Овощные однолетн.	ОВОЩ_О	кв.м.
6	Цветочные	ЦВЕТЫ	кв.м.

Окончание работы с БД

- Если вы заканчиваете работу с системой, то закройте таблицу и все окна объектов БД. Введите команду «Заккрыть» (с подтверждением всех изменений) и закройте окно системы. Если вы работали на жестком диске, скопируйте файл базы данных на свою дискету.
- Если вы продолжаете работать с системой, то выполните перечисленные действия в качестве упражнения, а затем опять откройте Access и свою БД.

Открытие базы данных (загрузка в окно системы)

- Запустите MS Access. В окне «Microsoft Access» щелкните на кнопке «Другие...». Откроется ДО «Открытие файла базы данных». Выполните в нем все необходимые настройки (имя вашей базы данных построено расширением .accdb) и нажмите кн. «Открыть». Через некоторое время на экране появится окно базы данных. Перейдите на вкладку «Режим Таблицы» и откройте таблицу «Группы». Убедитесь, что с таблицей все нормально, иначе исправьте ошибки или создайте ее заново.

Создание таблицы «Культуры».

Создание полей таблицы

Перед созданием второй таблицы нашей БД отметим особенности некоторых полей.

Код группы (Код Гр.) дублируется для установки связи с первой таблицей.

Код культуры (Код_культ) введен специально для однозначной идентификации записи. Код состоит из двух частей: первая часть кодирует обозначение культуры (четыре символа); вторая часть кодирует разновидность культуры (четыре символа). Между двумя частями установлен разделитель «-».

Для общих данных по культуре (без учета разновидности) вторая часть будет иметь вид : «ОБЩ.» (см. таблицу 4.3).

Срок созревания обозначен двумя датами (начало и конец), которые изображаются в инвертированном виде: мм.дд, где: мм — номер месяца; дд — число месяца. Тип полей — текстовой.

Использование записи неполных дат (без года) в таком виде позволит без осложнений выполнять операции сравнения, сортировки и выборки.

Для учета фактического состояния (наличия) культур на участке вводится два поля, в которых приводятся фактическое размещение культуры на участке и урожай (в кг).

Будем полагать, что если культура не высаживалась, то ее количество (или площадь) и урожай по умолчанию будут иметь значения: 999 (количество) и 0 (урожай).

Сформулируем общие требования к элементам эксплуатации таблицы «Культуры».

1. Код группы должен быть согласован с таблицей «Группы».
2. Код культуры следует вводить обязательно.
3. Записи таблицы должны быть упорядочены по коду группы и внутри групп — по коду культуры.
4. Поле «Ед. измер.» должно быть согласовано по свойствам с соответствующим полем из таблицы «Группы».

Данную таблицу мы создадим в режиме конструктора.

Отметим одно важное обстоятельство. Система требует, чтобы поля, по которым выполняется структурная связь между таблицами, имели одно имя и одинаковые значения некоторых свойств (тип, размер, формат, число десятичных знаков). Поэтому при создании поля «Код_Гр» и его свойств мы воспользуемся данными из первой таблицы (скопируем их во вторую таблицу).

Создавать таблицу будем в следующей последовательности.

1. Для копирования поля «Код_Гр» из первой таблицы таблицу «Группы» откройте в режиме конструктора.
2. Перейдите на вкладку «Создание» и в группе «Таблицы» нажмите кн. «Конструктор таблиц». Появится окно конструктора (возможно, с названием «Таблица 1») с чистым бланком.
3. Вновь перейдите в окно «Группы», например, используя ярлык вкладки. Выделите первую строку с полем «Код_Гр», нажмите ПКМ и введите команду «Копировать».
4. Вернитесь в конструктор новой таблицы, установите курсор в первую строку и введите команду «Вставить». Первое поле будет полностью определено. Однако неко-

- торые свойства этого поля надо откорректировать. Перейдите (по F6) на вкладку «Общие» и измените свойства.
- «Индексированное поле» = «Да (Допускаются совпадения)».
 - Свойства «Значение по умолчанию» и «Условие на значение» можно удалить.
5. Перейдите в верхний бланк для определения нового поля.
 6. В соответствии с данными таблицы 4.8 определите поля и значения характеристик в верхнем бланке.
 7. Далее мы должны определить первичный ключ. В этой таблице первичный ключ составной: он состоит из полей «Код_Гр» и «Код_культ». Выделите эти поля (строки) в верхнем бланке (придерживая Shift щелкните ЛКМ на соответствующих кнопках слева), найдите на панели инструментов кн. «Ключевое поле» и нажмите ее. Слева от имени полей появится изображение ключа.

Таблица 4.8

Поля и характеристики таблицы «Культуры»

Имя поля	Тип данных	Описание
1	3	4
Код_Гр	Числовой	Код группы культур
Код_культ	Текстовый	Код культуры и ее разновидности
Назв_к	Текстовый	Название культуры
Созр1	Текстовый	Срок созревания – от
Созр2	Текстовый	Срок созревания – до
Почвы1	Числовой	Отношение к кислотности почвы – от
Почвы2	Числовой	Отношение к кислотности почвы – до
Ед_измер	Текстовый	Единицы измерения количества
Кол-во	Числовой	Кол-во культур или площадь, занятая культурой
Урожай	Числовой	Урожай культуры (общий) в кг., полученный в тек. году

Определение свойств полей таблицы.

Краткое описание свойств приведено в таблице 4.9.

Будьте внимательны: маска ввода и условие на значение вводятся в одну строку без пробелов, независимо от того, как она изображена в таблице 4.9.

Самостоятельно установите основные свойства полей и придумайте тексты сообщений об ошибках для тех полей, которые имеют маску и условия на значение.

После установки свойств полей перейдите в режим отображения таблицы. В процессе перехода замените имя таблицы на «Культуры». Оцените вид таблицы и закройте ее. Закройте и таблицу «Группы», если она открыта.

Замечание.

Если Вы закрыли новую таблицу, но не изменили имя, то выделите ее в списке слева (в области переходов), нажмите ПКМ, введите команду «Переименовать» и переопределите ее имя.

Таблица 4.9.

Свойства полей таблицы «Культуры»

Имя поля	Размер и формат	Маска ввода	Подпись	Значен по ум.	Усло-вие на знач.	Обяз. поле
1	2	3	4	5	6	7
Код_Гр	P=байт, Ф=основн. ЧДЗ=АВТО	99	Код гр.			Да
Код_культ	P = 9	>CCCC– CCCC	Код культ.			Да
Назв_к	P = 40		Название			Нет
Созр1	P = 5	99\.99;0;	Созр._от			Нет
Созр2	P = 5	99\.99;0;	Созр._до			Нет
Почвы1	P=ПЛТ Ф =ФК ЧДЗ =1		Поч-вы_от	3	>=3 and <=8	Нет
Почвы2	P=ПЛТ Ф =ФК ЧДЗ = 1		Поч-вы_до	3	>=3 and <=8	Нет
Ед_измер	P = 5		Ед. измерен.			Нет
Кол-во	P=ПЛТ Ф =ФК ЧДЗ = 1		Кол-во культ.	999		Нет
Урожай	P=ПЛТ Ф =ФК ЧДЗ = 1		Урож. (общ)	0		Нет

§ 5. Определение структурной связи между таблицами

Для того, чтобы БД представляла собой единый информационный «организм», а не простой набор таблиц, необходимо выполнить еще одну настройку — связать таблицы определенным отношением.

MS Access предоставляет возможность создания различных типов связей на разных уровнях.

На уровне БД — мы определим логическую структурную связь между таблицами, отражающую основную (иерархическую) взаимосвязь данных.

На уровне запросов, форм ввода и редактирования и отчетов - могут быть установлены другие связи локального характера.

Связи отражают взаимозависимости между множествами данных в разных таблицах.

Различают следующие типы связей (отношений).

«Один к одному» (1:1): одной записи одной таблицы соответствует одна запись другой таблицы.

«Один ко многим» (1:M): одной записи одной таблицы (ее, в этом случае называют родительской) соответствует множество записей другой таблицы.

«Многие к одной» (M:1): многие записи одной таблицы связаны с одной записью другой.

Связь «Многие ко многим» данной СУБД стандартными средствами не поддерживается.

Установка связи выполняется по определенным полям, которые должны быть идентичными по типу данных, размеру и другим характеристикам. Связь обеспечивает более надежное сохранение данных при ошибках пользователя.

Наша модель данных устанавливает связь 1:M, где связующим полем является «Код_Гр» — первичный ключ одной таблицы («Группы») и идентичное поле другой (т.н. вторичный ключ). Напомним, что в таблице «Культуры» первичный ключ составной: «Код_Гр + Код_культ».

Взаимосвязь 1: M можно изобразить схемой, представленной на рисунке 4.3.

Связь между нашими таблицами определим следующим образом.

1. Закройте окна «Группы» и «Культуры», если они открыты.

2. На вкладке «Работа с базами данных щелкните на кн. «Схема данных».

3. Если вы впервые определяете связи, то появится пустое окно «Схема данных» (см. рис. 4.5, где приводится уже готовая схема). Если в этом окне уже существуют какие-то связи и таблицы, нажмите кн. «Очистить макет».

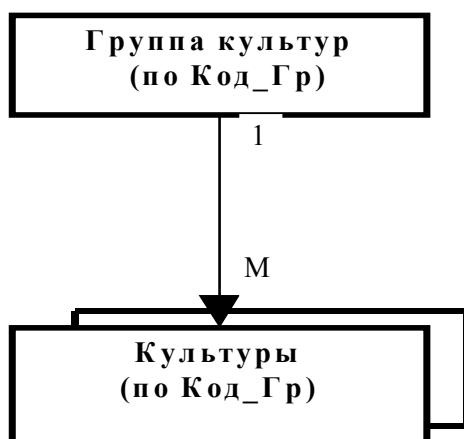


Рис. 4.3. Схема взаимосвязей таблиц

4. Далее на вкладке «Конструктор» нажмите кн. «Отобразить таблицу» и добавьте в схему данных обе наши таблицы. На схеме они будут отображаться в виде окошек с перечнем полей.

5. Закройте ДО «Добавление таблицы».

Внутри окошек таблиц поля, которые входят в первичный ключ, имеют более жирное начертание.

Создадим связь между таблицами.

1. Установите курсор мыши на поле «Код_Гр» таблицы «Группы» и нажмите ЛКМ.

2. Не отпуская ЛКМ, перетащите появившийся значок на аналогичное поле второй таблицы и отпустите ЛКМ.

3. Откроется ДО «Изменение связей» (рис. 4.4). В нижней части окна находится строка «Тип отношения», где должно быть написано «один – ко – многим». Если это не так, то вы сделали ошибку при определении свойств полей «Код_Гр». Закройте окно определения связи кнопкой «Отмена» и проверьте установку имен и свойств связующего поля в обеих таблицах.

4. В окошках «Таблица / Запрос» и «Связанная таблица / Запрос» вы увидите имена таблиц и ниже — имена полей.

5. Так как не имеет смысла хранить данные для тех групп культур, которые не существуют,) включите индикатор «Обес-

печение целостности данных». В этом случае система не позволит добавить в таблицу «Культуры» запись с отсутствующим значением кода группы и не позволит удалить код группы из таблицы «Группы», на котором «висят» записи в другой таблице.

6. Включите индикатор «Каскадное удаление связанных полей». Это позволит автоматически удалять все записи дочерней таблицы «Культуры», если в таблице «Группы» будет удалена соответствующая группа.

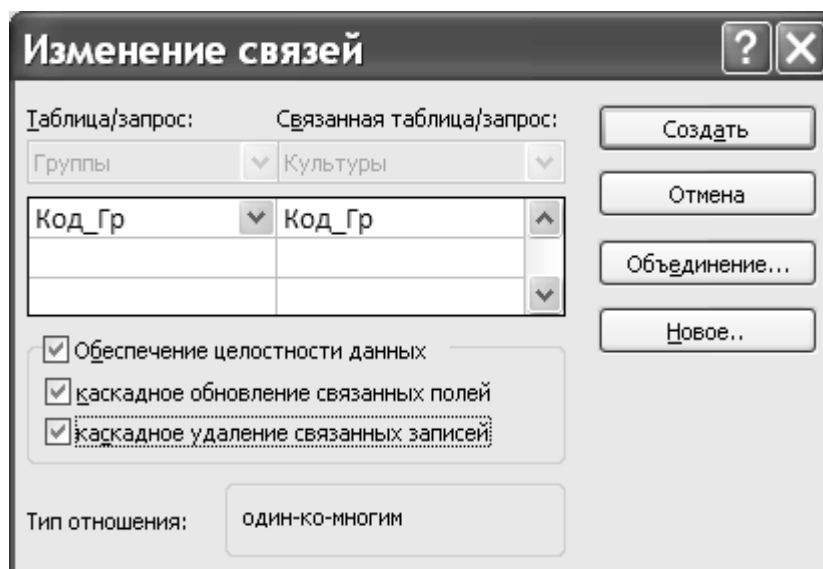


Рис. 4.4. Окно установки параметров связи

7. Включите индикатор «Каскадное обновление связанных полей». Это позволит автоматически обновлять связанные значения поля «Код_Гр» дочерней таблицы «Культуры», если в первой таблице будет изменено значение ключа.

8. После установки параметров связи нажмите кн. «Создать». Между таблицами появится линия, показывающая связь «один ко многим» в виде: «1 — :».

9. Введите команду «Файл / Сохранить» и закройте окно со схемой (рис. 4.5).

Замечания.

1) Если необходимо удалить связь, то щелкните ЛКМ на ее линии и нажмите кн. «Del».

2) Для корректировки связи щелкните 2ЛКМ на линии. При этом откроется ДО «Изменение связей».

- Откройте таблицу «Группы».

Хотим обратить Ваше внимание на появление в таблице «Группы» дополнительного столбца слева, где отображаются символы «+». При нажатии на него открываются фрагменты таблицы «Культуры» с отфильтрованными записями по коду группы. Знак при этом изменяет форму на «-». Щелкните по нему ЛКМ, и фрагмент таблицы «Культуры» закроется.

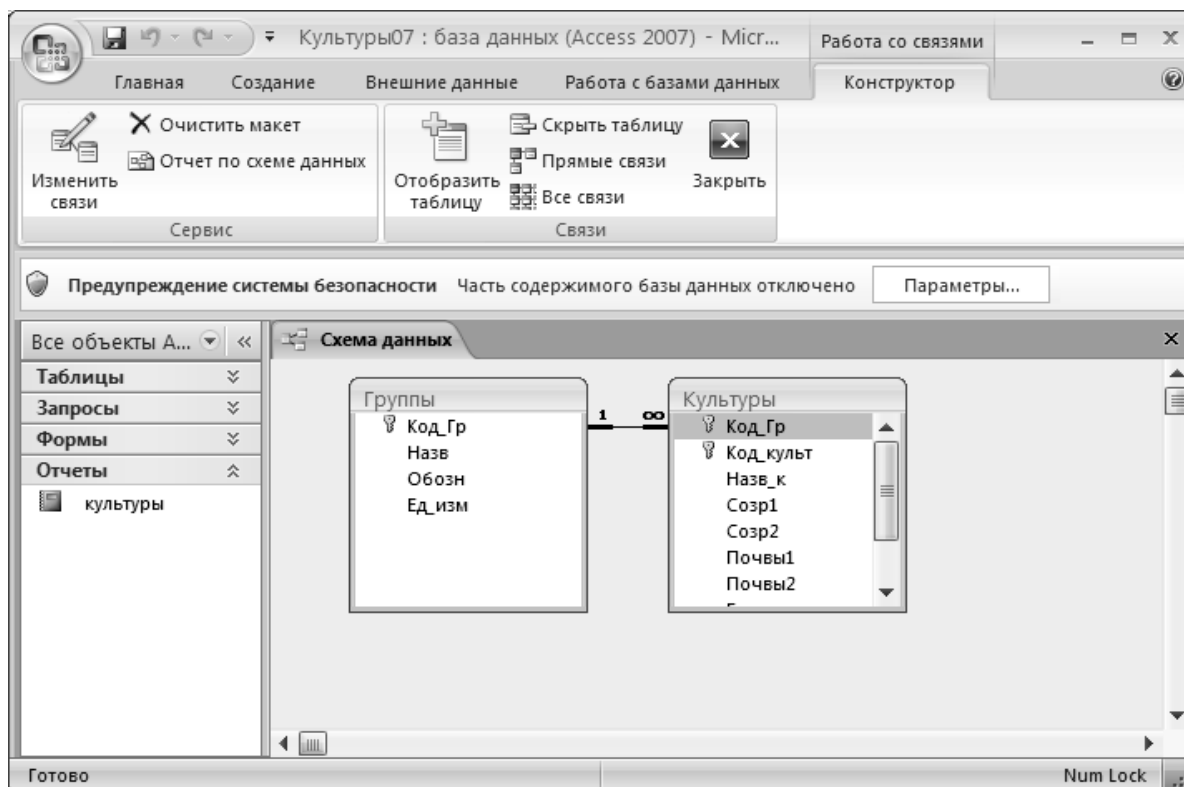


Рис. 4.5. Окно со схемой данных для БД «Культуры»

§ 6. Ввод, просмотр и корректировка данных. Подстановки

В этом разделе мы познакомимся с основными операциями обслуживания БД в режиме «Таблица».

Ввод данных в таблицы и их корректировка

Выше мы рассмотрели некоторые приемы ввода данных для таблицы «Группы» и некоторые способы управления курсором.

Начальные настройки параметров отображения таблицы.

- Откройте таблицу «Группы». Перейдите на ленте на вкладку «Главная». Здесь расположены основные инструменты форматирования таблиц и данных.
- Еще раз посмотрите на внешний вид таблицы и попробуйте его изменить. В верхней части выделены столбцы с названиями полей (Подписями). Записи и столбцы обозначены сеткой, которая выводится на печать при распечатке таблицы.
- С помощью кнопки «Сетка» в группе «Шрифт» можно управлять режимом отображения сетки. Попробуйте выполнить несколько операций с сеткой. Здесь же расположены элементы форматирования данных в ячейках таблицы (шрифт, его начертание, способы выравнивания и др.). Самостоятельно поупражняйтесь с установкой разных элементов форматирования и затем приведите все в стандартный вид: включите линии сетки; цвет фона — белый; цвет линии — черный или серый; оформление — обычное; шрифт - Times New Roman Cyr 11.
- Теперь **откройте вторую таблицу «Культуры»** и настройте ее внешний вид по своему усмотрению.
- К настоящему моменту таблица пуста. Введите пару записей из таблицы 4.3 (Яблони: зимн. и летн.). Для этого установите курсор ввода в первом поле первой (пустой) записи и введите данные первой строки (записи). Номер записи вводить не надо. Код группы вводите в виде двух цифр (например, 01). После ввода значения каждого поля нажмите кн. Tab (или кн. Enter). Код культуры можете вводить строчными буквами. В первый столбец (поле «Код_Гр.») второй записи попробуйте ввести число 20. Система выдаст сообщение об ошибке. Повторите ввод правильных данных (01).

Замечания

1. При достижении правого края поля, если значение его не умещается внутри клетки, начинается скроллинг этого значения и начальные символы «уходят» за левый край клетки.

2. Аналогичный эффект проявляется со строками таблицы: после достижения правого края окна начинается скроллинг записи.

3. При вводе данных в поля, для которых заданы начальные значения — сначала сотрите их кн. Del, а затем введите новое число.

4. Дробная часть числа отделяется от целой запятой.

5. Сроки созревания вводите полностью, включая нули. Точку вводить не надо, так как она определена маской.

Изменение макета таблицы «Культуры»

Возможно, что ввод данных показался вам не очень удобным. Ширина одних полей оказалась избыточной, а другие поля было бы желательно расширить.

При вводе значений последних полей вы уже забыли для какой культуры вводятся данные (поле с названием культуры «ушло» за экран).

Вас не устраивает вид шрифта, его размер, цвет или фон элементов таблицы и значений и т. д.

Для более удобного ввода данных нам потребуется выполнить несколько предварительных операций.

- Прежде всего, распахните полностью окно.

Для настройки вида (макета) таблицы система предлагает несколько способов. Можно использовать универсальные средства из вкладки «Главная». Для некоторых операций можно использовать оперативные визуальные средства - специальные точки таблицы, которые активизируются указателем мыши или команды всплывающего меню.

- Например, изменим ширину столбцов. Установите курсор на заголовок поля «Название», нажмите ПКМ и введите команду «Ширина столбца» и далее укажите команду «По ширине данных» (или «По содержимому»).
- Повторите операцию для других столбцов, используя другой визуальный способ с двойным щелчком мыши, который мы уже использовали при создании таблицы «Группы». Напомним, что для этого надо проводить курсор по полоске с названиями столбцов и, как только он примет

форму крестика с двунаправленными стрелками, нажать 2ЛКМ.

Для удобства ввода длинных записей система позволяет **«закрепить» некоторые столбцы** так, чтобы они постоянно находились на экране. Мы закрепим два столбца: «Код гр.» и «Код культ.».

- Установите курсор на заголовок столбца «Код гр.» И, когда появится жирная стрелка, нажмите ЛКМ. Столбец – выделится. Нажмите Shift и повторите операцию для второго столбца «Код культ.». Оба столбца должны быть выделены. Нажмите ПКМ и через меню введите команду «Закрепить столбцы». Если этот способ не сработает, воспользуйтесь командой «Главная» / кн. «Записи» / кн. «Дополнительно» / «Закрепить». Теперь эти столбцы всегда будут оставаться (при скроллинге) на экране.
- Сохраните изменения.
- После сделанных изменений введите еще одну запись (№ 3 – «ЯБЛ.–ОСЕН»).

Замечания по вводу и редактированию записей в таблицах.

Необходимо помнить, что ключевые поля (в обеих таблицах) имеют статус «Обязательное». Если вы дали команду добавить запись, а затем решили отказаться от этого, то система не «выпустит» вас из новой записи до тех пор, пока вы не введете соответствующее значение (можно ввести любое значение соответствующее типу поля). После этого можно эту запись удалить.

При изменениях значений полей будьте внимательны, система не предупреждает о сохранении новых значений.

Для внесения изменений в значение поля записи необходимо сначала найти запись и затем «войти» в нужное поле. Сделать это можно разными способами.

Для установки курсора на нужную запись вы можете пользоваться линейками прокрутки, клавишами управления курсором, щелкнуть мышью на кнопке «маркера» текущей записи или нажать клавишу F5, ввести в окошко «Запись» номер записи и нажать «Enter».

Для перехода на поле используйте мышь или клавиши «Tab» или «Shift–Tab». Текущее поле выделяется другим цветом. Если

Вы ошиблись при вводе данных, то отменить изменения (в поле) можно клавишами «Ctrl +Z».

Постарайтесь запомнить!

Возможна такая настройка системы, что курсор редактирования помещается в начало или конец значения поля без его выделения. В этом случае поле можно корректировать «посимвольно» (в режиме обычного редактирования).

Если вы хотите заменить поле целиком, а оно выделено сразу начинайте ввод нового значения.

Если поле выделено, а надо изменить отдельные символы, нажмите F2 для перехода в режим обычного редактирования, после изменений вернитесь по F2 в обычный режим.

Посимвольная корректировка значения почти не отличается от редактирования строки текста в текстовом редакторе (возможны некоторые особенности из-за наличия символов маски, исполнения функций преобразования и контроля значений).

- Откройте таблицу «Группы» и откорректируйте значения поля «Название» для 2-ой записи. Измените название «Кусты» на «Кустарник». Закройте таблицу «Группы».

Использование свойств «Подстановки» для отдельных полей таблицы «Культуры»

Как бы тщательно мы не проектировали БД и ее элементы, наступает момент, когда требуется внести в нее и в таблицы какие-то изменения.

В этой работе мы рассмотрим простые задачи по изменению свойств некоторых полей таблицы «Культуры».

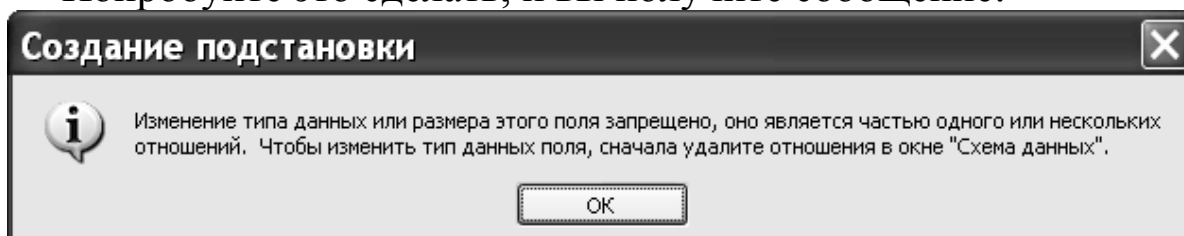
Содержательная сторона этих изменений состоит в следующем.

Если бы мы вводили данные не из готовой таблицы, а воспользовались первоисточниками (справочниками, книгами, статьями из журналов и т.п.), то при вводе «Кода группы» и указании «Единиц измерения» нам надо было бы постоянно держать перед глазами «шпаргалку» с числовыми кодами групп и обозначениями единиц измерений. Чтобы облегчить себе ввод такой информации, мы наделим эти поля особым свойством «подстановки» — создадим для них элемент управления «Поле со спи-

ском», а сам список кодов групп будем формировать на основе соответствующего поля таблицы «Группы». А для единиц измерения используем простой список.

В данной версии системы изменить свойство поля, входящего в ключ, без удаления связи нельзя.

Попробуйте это сделать, и вы получите сообщение:



- Поэтому закройте таблицы. Откройте «Схему данных» и удалите связь.

Создание поля со списком для «Кода группы» в таблице «Культуры».

- Закройте таблицу «Группы», если она открыта и откройте таблицу «Культуры» в режиме конструктора.
- В строке с кодом группы «Код_Гр» активизируйте поле «Тип данных». Откройте список типов и запустите «Мастер подстановок» (ЛКМ на соответствующем пункте).
- По этой команде запускается специальная программа «Создание подстановок» (одно из окон приводится на рис. 4.6), которая сама определит порядок работы по созданию списка.
- В первом окне «мастера» включите индикатор («☉») «Объект «столбец подстановки»...» и нажмите кн. «Далее».
- В следующем окне установите индикатор «Таблицы» и выберите таблицу («Группы») и кн. «Далее».
- В следующем окне (рис. 4.6) вы должны выбрать поля таблицы, которые войдут в список. Первым полем обязательно должно быть поле, значения которого будут переходить в поле «Код_Гр» таблицы «Культуры». Поэтому установите подсветку на поле «Код_Гр» и нажмите кн. с символом «>». Поле переместится в окно «Выбранные поля». Повторите эту операцию с полем «Обозн». Нажмите кн. «Далее».

- Следующее окно устанавливает особенности сортировки элементов списка. Мы установим сортировку по возрастанию поля «Код_Гр».
- В следующем окне включите отображение ключевого столбца. Для этого отключите индикатор («□») «Скрыть ключевой столбец». Затем подкорректируйте ширину столбцов (рис. 4.7) и нажмите кн. «Готово».

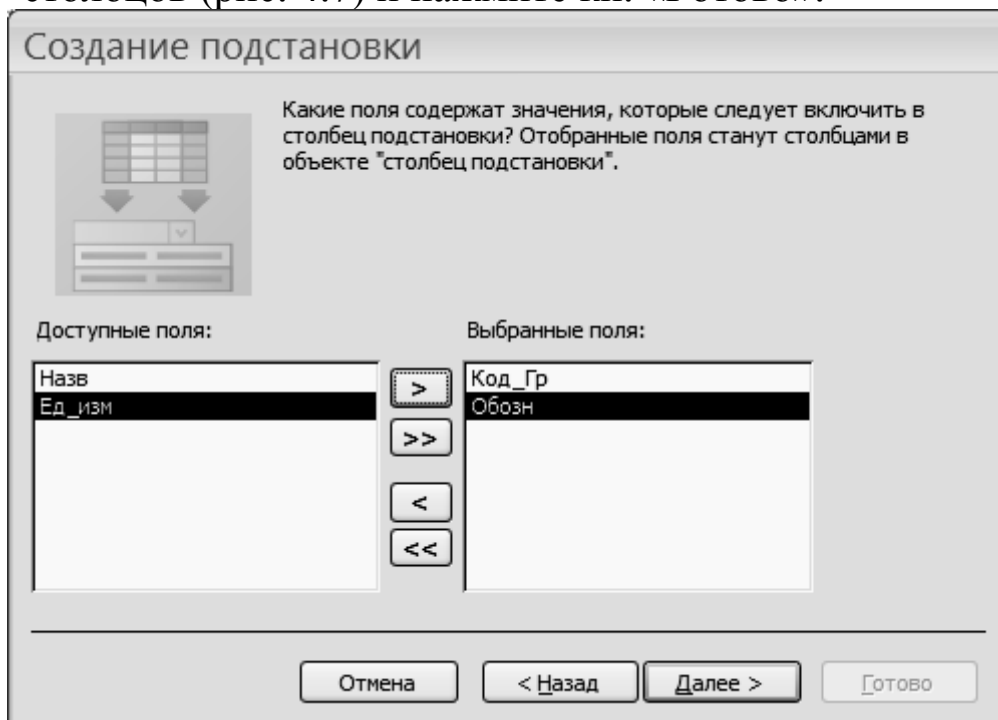


Рис. 4.6. Окно отбора полей при создании списка

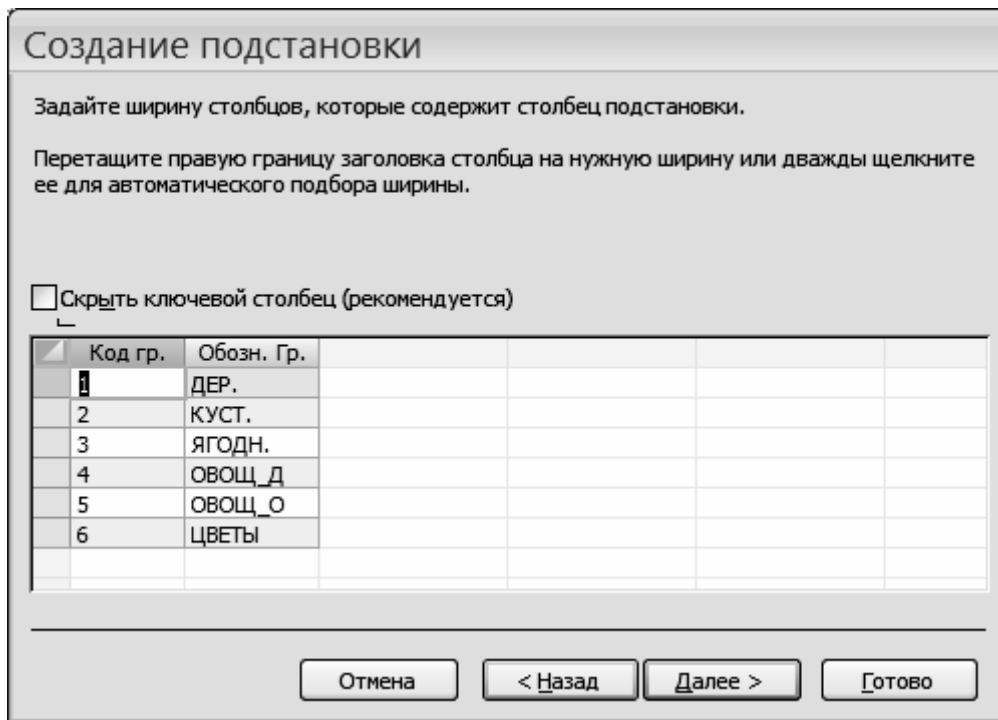


Рис. 4.7. Сформированная таблица подстановок

Окно «мастера» закроется, будет выведено сообщение о сохранении изменений, ответьте «ДА», и мы вернемся в режим конструктора таблицы.

- Откройте вкладку «Подстановка» и внимательно просмотрите значения свойств, которые установил «мастер» для поля. В окошке свойства «Источник строк» содержится запрос к таблице «Группы», сформированный на специальном языке запросов SQL. В окне ничего не меняйте.
- Перейдите в режим «Таблица» (при запросе, изменения следует сохранить). В поле столбца «Код гр.» появилась кнопка с символом «▼». Щелкните по ней ЛКМ или «Alt - ↓» и убедитесь, что список работает нормально. В противном случае повторите создание списка с помощью «мастера».
- Введите еще три записи (№4 - №6) из таблицы «Культуры» с использованием подстановки.
- **!! Обязательно восстановите связи между таблицами.**

Создание списка для поля «Единицы измерений»

Для поля «Ед_измер» мы создадим (из методических соображений) простой список, содержащий множество возможных зна-

чений этого поля. Последовательность операций та же, что и в предыдущем случае (связь можно не удалять), но в первом окне «Создание подстановки» надо включить опцию «Будет введен фиксированный набор значений» и в следующем окне ввести элементы массива: «Дер.», «Куст.», «Кв.м.» (без пробела), «Га.». Затем нажать кн. «Далее», оставить имя столбца в «Ед_измер». и нажать кн. «Готово».

- Выполните эти операции.
- При переходе в режим таблицы сохраните макет и ведите еще две записи (№ 7 и № 8) из таблицы 4.2.
- Закройте таблицу. Все изменения сохраните.

§ 7. Создание форм для ввода/редактирования данных в таблицах

Общие сведения о формах

Формы являются специальными окнами (объектами) БД, которые позволяют создать специальные средства ее обслуживания, более эффективные, чем при традиционном табличном представлении данных.

Наиболее эффективно формы проявляют себя при вводе и редактировании отдельных записей с большим количеством полей.

Формы могут создаваться различными способами:

- с помощью мастера форм;
- с использованием «Автоформы»;
- вручную (в режиме конструктора).

По характеру использования они создаются для обслуживания:

- отдельной таблицы;
- группы взаимосвязанных таблиц;
- различных запросов.

Система позволяет создавать три базовых типа форм (в режиме «Автоформа»).

1. Форма типа «Формуляр» (в «один столбец»). При использовании этого типа один «лист» формы обслуживает поля одной

записи таблицы. Если все поля записи не умещаются на одном экране, то можно создать многостраничную форму. Данный тип используется наиболее часто.

2. Ленточная форма. Это — удобный вид формы представления данных в виде таблицы для обработки нескольких записей. Отличается от традиционной (табличной) наличием специального стиля оформления записей и разделов, где можно разместить элементы управления.

3. «Разделенная форма». Эта форма состоит из двух частей. В верхней части отображаются поля текущей записи (в столбец). В нижней — таблица с отфильтрованными записями (например, по коду группы). Обе части синхронизированы.

Главная особенность (и отличие от стандартного представления) формы состоит в том, что мы можем внутри окна формы создавать различные элементы управления, в т.ч. кнопки, вычисляемые поля и др. элементы управления, и работать с данными без использования главного меню системы, панелей инструментов и других средств управления среды СУБД.

Структура формы (в режиме конструктора)

Форма первого и второго типов содержит несколько разделов, в которых могут размещаться заголовки (текстовые строки), вычисляемые поля, элементы управления — кнопки, поля БД, комментарии, рисунки и другие объекты.

Окно формы соответствует общим стандартам окон системы Windows.

Поле окна формы имеет, как правило, три раздела:

- заголовок;
- область данных;
- раздел примечаний.

Если форма одностраничная (типичный случай), то все разделы равноправны, если многостраничная, то заголовок присутствует на каждой странице, а примечание — в конце последней страницы.

Создание форм.

При наличии двух связанных таблиц можно использовать два варианта их построения с использованием, например, мастера или «Автоформы».

Первый способ. Каждая форма создается независимо, а затем их связывают с помощью специальной кнопки, которая вставляется в ведущую (главную) форму (в нашем случае – в форму таблицы «Группы»). Эта кнопка программируется на открытие второй формы («Культуры»).

Второй способ. Обе формы создаются мастером «за один проход». При этом система, руководствуясь схемой данных (определенной ранее взаимосвязью таблиц), сама установит связь между формами и разместит форму «дочерней» таблицы внутри «родительской» формы в виде т.н. подчиненной таблицы.

Из методических соображений воспользуемся сначала первым способом.

Сначала создадим форму «формуляр» для таблицы «Культуры» самым простым способом.

- Для создания формы откройте БД, если она закрыта, и последовательно выполните следующие действия.

1. Закройте таблицы. Проверьте схему данных и, если она нарушена (что возможно после создания полей со списками), то восстановите ее.

2. Выделите в области переходов т. «Культуры». Откройте на ленте вкладку «Создание» и щелкните ЛКМ на кн. «Форма». Система создаст форму по своему шаблону или с учетом тех настроек, которые были сделаны при создании предыдущей формы, и отобразит ее в режиме «макета» (см. рис 4.8). Вид окна и его содержимое может немного отличаться от того, что вы видите на рисунке.

В форме перечислены все поля (даны их подписи) и справа расположены окошки для ввода значений. В нижней части находятся кнопки «навигации», с помощью которых вы можете передвигаться по записям таблицы. Кнопка с символами «▶*» позволяет добавить в таблицу новую запись. Вы можете передвигаться по записям таблицы и с помощью клавиш PageUp или PageDown, а внутри записи с помощью клавиш со стрелками. После ввода новых значений или исправления старых нажимайте «Enter».

В левом верхнем углу формы находится символ макета – квадрат с пересекающимися стрелками.

- Для того чтобы уменьшить размер полей, установите курсор справа на их границе (курсор примет форму двунаправленной стрелки), нажмите ЛКМ и перетащите границу вправо. Обратите внимание, что форма унаследовала некоторые свойства из соответствующей таблицы.

Посмотрите на вкладку «Формат», с помощью которой можно отформатировать макет формы по своему усмотрению. При этом форматирование будет распространяться на все сгруппированные элементы формы.

- Закройте форму, сохранив макет и установите для формы имя «Культуры».

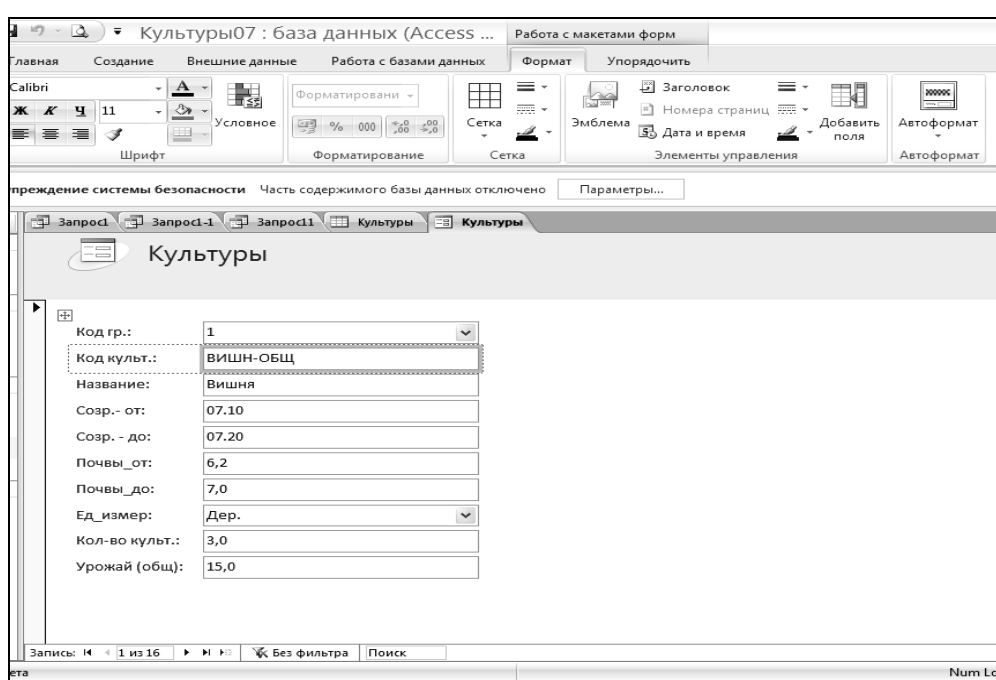


Рис. 4.8. Вид формы «Культуры» в режиме «Макета»

Если система разместит имя формы в разделе «Все таблицы» окна области переходов, то откройте раздел «Формы» и переместите имя созданной формы в этот раздел.

Используя форму введите еще три записи (№ 9, 10, 11) из таблицы 4.3. Проверьте правильность ввода переходом в режим «Таблицы».

Создание формы для таблицы «Группы»

- Создадим форму для таблицы «Группы» с помощью «мастера».

1. На вкладке «Создание» нажмите кн. «Другие формы...» и выберите способ «Мастер форм».

2. Активизируется первое окно мастера «Создание форм», где следует установить имя таблицы.

3. В нижней части находятся два окна – списка. В левом окне перечислены поля таблицы. Правое окно пустое.

4. Нам надо все поля вставить в форму (сделать выбранными), т. е. переместить из левого окна в правое. Для этого нажмите кн. с символами «>>». Все поля нашей таблицы должны переместиться в правое окно. После этого нажмите кн. «Далее».

5. Откроется окно второго шага — выбора внешнего вида формы. Включите опцию «В один столбец» и нажмите кн. «Далее». Откроется окно третьего шага — выбора стиля оформления формы. Установите стиль «Стандартный» («Обычный») или любой другой (но учтите, что система при этом может использовать другой шрифт и размер полей) и нажмите кн. «Далее».

6. На четвертом шаге открывается последнее окно с именем формы «Группы», которое мы изменять не будем. Установите опцию «Открытие формы для просмотра...» и нажмите кн. «Готово». Система выведет на экран окно формы в режиме макета. На рис. 4.9 приводится вид формы в «рабочем» режиме «Форма». Подкорректируйте размеры полей. На рис. 4.10 приводится вид формы в режиме конструктора.

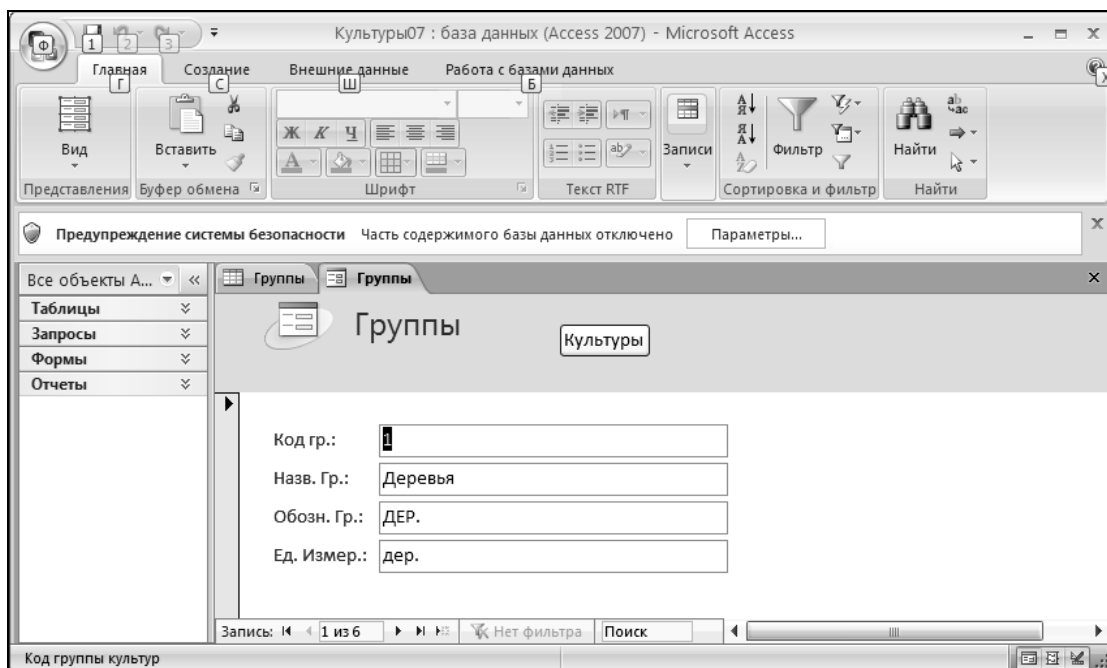


Рис. 4.9. Вид формы «Группы» в режиме «Форма».

Изменение свойств форм в режиме конструктора

Все элементы, размещенные в форме, называются **элементами управления (ЭУ)**.

В авторежиме система разместила в наших формах только имена полей или «Подпись», сами поля из базовых таблиц, кнопки навигации и текстовые строки, показывающие номер текущей записи и количество записей в наборе.

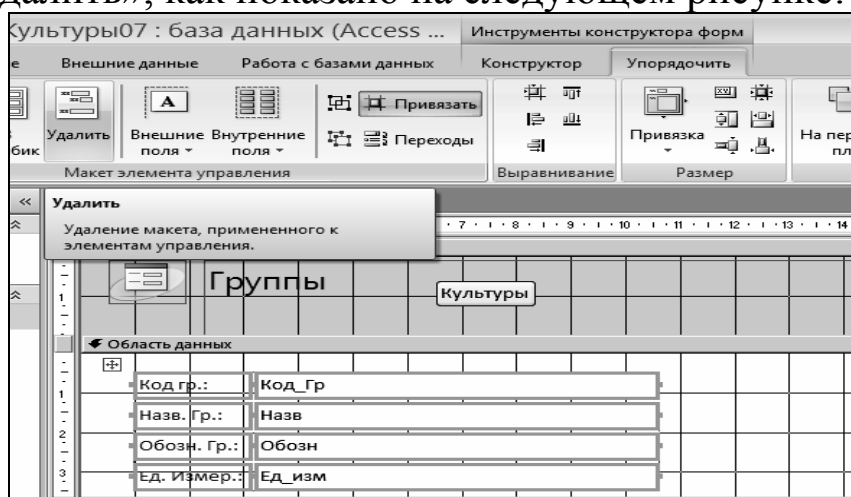
Поле — основной ЭУ, который присутствует практически в каждой форме. Другим важным ЭУ считается **командная кнопка (КК)**, которая позволяет выполнить определенную команду или программу, инициализируемую при нажатии на нее.

И сама форма и каждый ЭУ имеет набор характерных свойств. Свойства объединяются в группы. Количество групп и набор свойств в группах существенно зависят от вида ЭУ.

Поля базовых таблиц в форме наследуют большинство своих свойств. Например, поля «Код гр.» и «Ед_измер.» формы «Культуры» унаследовали свойства «Подстановки» (списки) из соответствующей базовой таблицы. Но такие свойства, как например, размер (ширина) полей и шрифт система установила по своему усмотрению.

Рассмотрим тот инструментарий, который позволяет оценить свойства ЭУ, откорректировать некоторые из них, используя разные средства и способы.

- Откройте форму «Группы», если она закрыта, и перейдите любым способом в режим конструктора. В этом режиме вы увидите макет формы с тремя разделами (см. рис. 4.10, где уже выполнены некоторые операции форматирования и вставка кнопки). Окно свойств формы закрыто. В разделе «Заголовок формы» приводится пиктограмма формы и название. Раздел «Примечания» пуст. На ленте появляется вкладка «Конструктор» с необходимыми инструментами форматирования и конструирования.
- Квадратик с крестиком из стрелок в левом верхнем углу области данных (по аналогии с режимом макета) говорит о том, что все поля и надписи сгруппированы, т. е. при их изменении они ведут себя, как единое целое.
- Если вы хотите разгруппировать ЭУ и отформатировать их по своему, то выделите весь блок элементов щелчком на крестике (появится толстая рамка), откройте вкладку «Упорядочить» и в группе «Макет» введите команду «Удалить», как показано на следующем рисунке:



После этого элементы управления становятся доступными для редактирования. Однако, например, поля и подписи (надписи) остаются сгруппированными. Это хорошо видно на рис. 4.10. Если Вы хотите форматировать их отдельно, то надо нажать кн. «Разгруппировать».

- Щелкните 2ЛКМ на квадратике, расположенном слева от горизонтальной линейки конструктора. Откроется окно свойств с вкладками (группами свойств). Бегло просмотрите список свойств на вкладках, ничего не меняйте и закройте это окно.

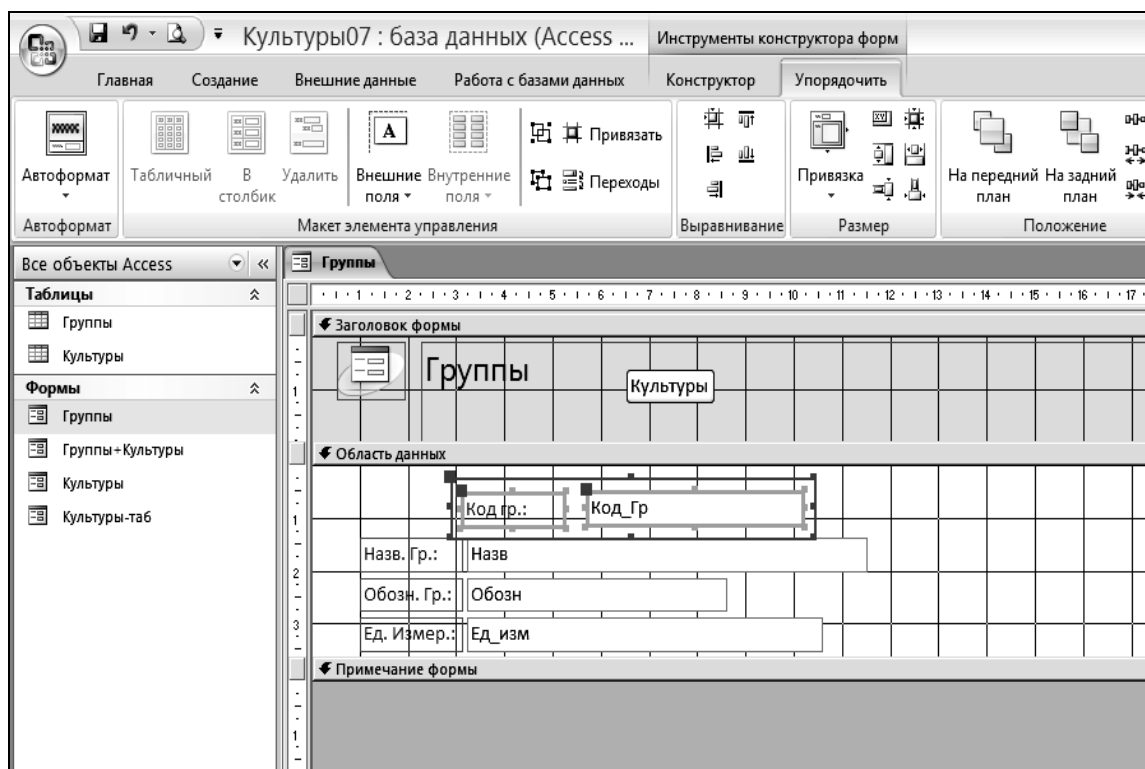


Рис. 4.10. Вид формы «Группы» в режиме «Конструктора»

Для корректировки свойств или местоположения существующих ЭУ их надо выделить. Для выделения надо привести указатель на ЭУ и щелкнуть ЛКМ. Он выделится маркерами, а курсор примет другую форму (в зависимости от его положения). С помощью маркеров можно изменить размеры элемента. Для перетаскивания ЭУ в другое место формы надо привести курсор на линию элемента между маркерами, нажать ЛКМ и, не отпуская ЛКМ, отбуксировать элемент в нужное место.

При корректировке поля надо учитывать, что этот ЭУ состоит из двух частей: «Надписи» и «Поле» (в режиме конструктора в поле отображается имя поля). Они обычно выделяются и перемещаются совместно (см. рис. 4.10). Если вы хотите отделить надпись от поля, то надо их разгруппировать, затем зацепить маркер в левом верхнем углу поля или надписи (он имеет боль-

ший размер, чем остальные маркеры) и переместить ЭУ в новое место.

Для выделения группы элементов можно использовать стандартные приемы: придерживать клавишу Shift при выделении или построить указателем мыши рамку, охватывающую или пересекающую нужную группу элементов. Этот прием надо использовать, когда вы хотите, например, сгруппировать ЭУ.

- Попробуйте самостоятельно изменить некоторые свойства (геометрические или местоположения) элементов полей. При ошибках можно отменить изменения, например, клавишами «Ctrl – Z».
- Оцените ваши изменения, перейдя в режим формы.

Вставка командной кнопки для открытия формы «Культуры».

При наличии ведущей таблицы («Группы») и соответствующей ей формы мы можем создать такой режим работы с БД, что пользователь, открыв форму «Группы», сможет из нее открывать форму «Культуры».

Для реализации такого режима мы используем специальный элемент управления – командную кнопку (КК).

Таблица «Группы» должна находиться в режиме конструктора.

- Кнопку вставим в раздел «Заголовок формы» (рис. 4.10).
- Щелкните ЛКМ на пустом месте раздела и нажмите в группе «Элементы управления» конструктора кн. «Кнопка». Курсором постройте в правой части раздела небольшой прямоугольник на 2 – 3 квадрата. После отпускания ЛКМ появится кнопка с названием «КнопкаNN», где NN – номер кнопки (возможно, он будет равен 1). Одновременно на экран система выведет первое окно мастера «Создание кнопок» (рис. 4.11).
- Установите категорию «Работа с формой» и в окошке «Действия» определите «Открыть форму» → кн. «Далее». На втором шаге выберите форму «Культуры».
- На следующем шаге включите опцию «Открыть форму для отобранных записей».

- Далее - выберите поля для отбора записей: в окнах «Группы» и «Культуры» укажите поле «Код_Гр».
- На следующем шаге включите опцию «Текст» и замените надпись на кнопке «Культуры».
- На последнем шаге оставьте имя кнопки, установленное системой, и нажмите кн. «Готово».

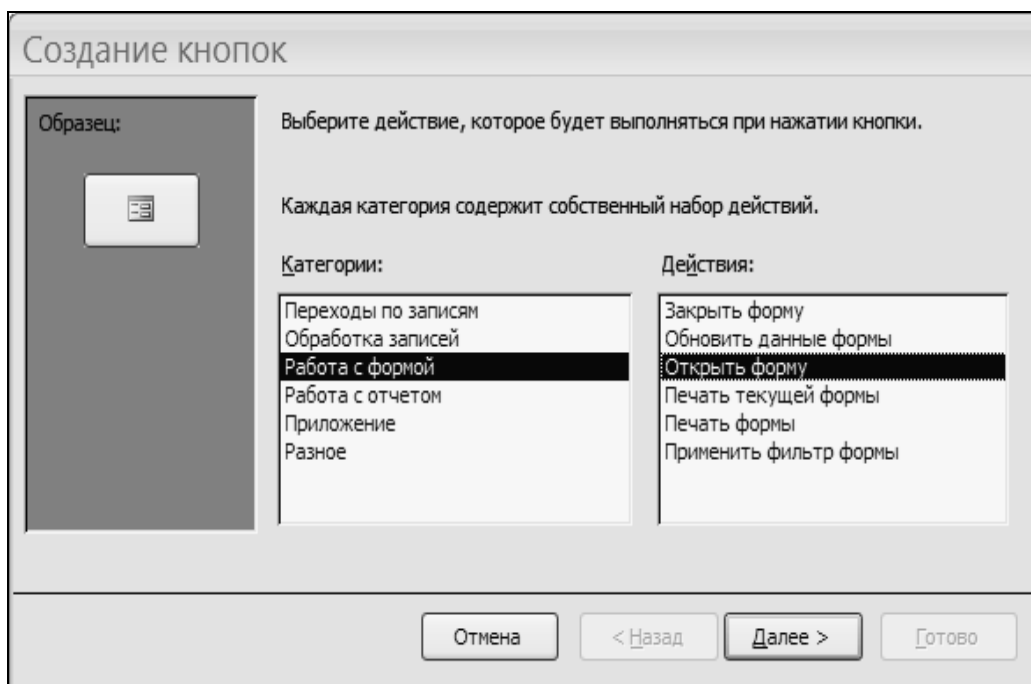


Рис. 4.11. Начальное окно создания кнопок

- После выхода в конструктор откорректируйте размер, шрифт и местоположение кнопки по своему усмотрению. Перейдите в режим «Формы» и оцените ваши построения. Если необходимо, откорректируйте. Опробуйте действие кнопки. Введите в т. «Культуры» оставшиеся три записи из таблицы 4.3 (код группы вводить обязательно).
- Попробуйте самостоятельно вставить кнопку для закрытия формы «Культуры (действие «ЗакретьФорму»).

Рассмотренный способ создания «автономных» форм при наличии структурных связей между таблицами не является оптимальным. Возможно, что в этом случае пользователю будет более удобно использовать «комбинированную» форму, где данные «родительской» формы («Группы») будут представлены одной записью, а все записи «дочерней» формы («Культуры»), соответ-

ствующие значению ключевого поля («Код_Гр»), будут отфильтрованы и представлены в виде например, таблицы, расположенной внутри «родительской» формы. Причем состояние форм должно быть синхронизировано (рис. 4.12).

Форма – таблица, размещенная внутри «родительской», называется **подчиненной**.

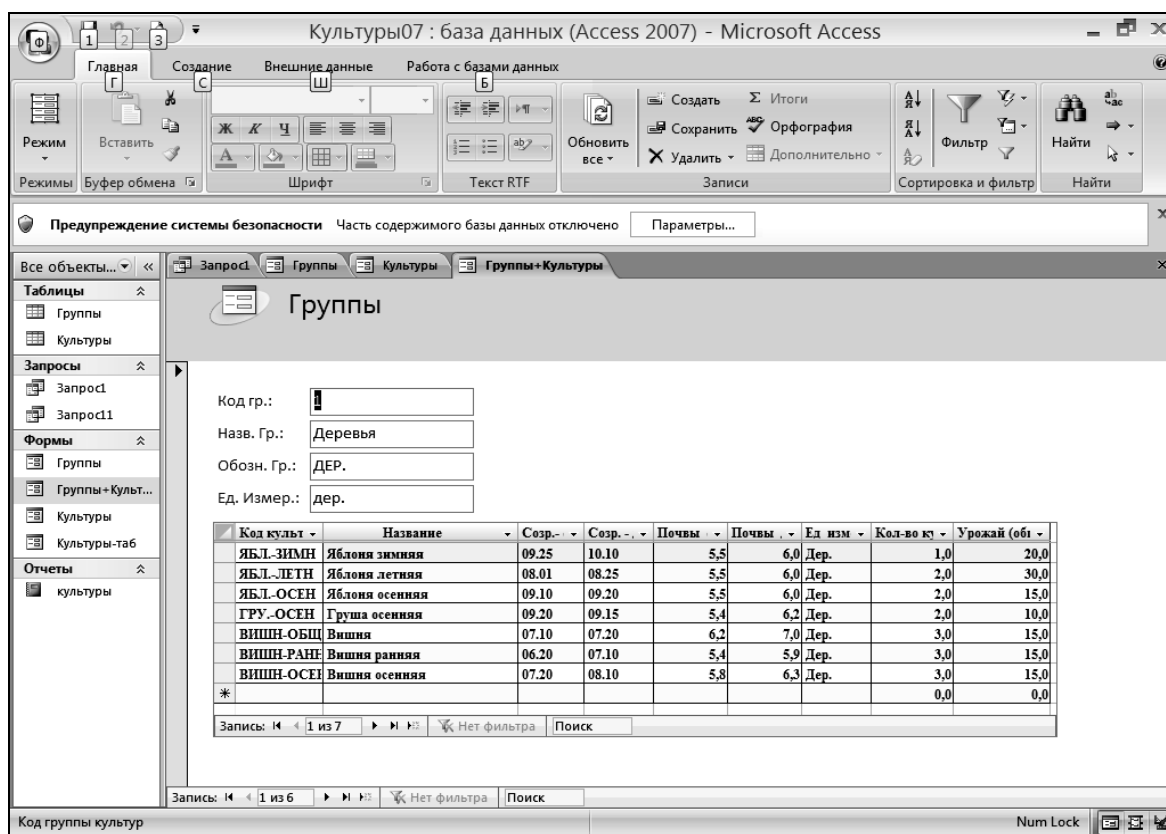


Рис. 4.12. Вид комбинированной формы (с подчиненной)

- Для создания такой формы (если вы не собираетесь менять состав полей) выделите таблицу «Группы» и запустите «Мастер форм» (командой: «Создание / гр. Формы / Конструктор форм / Мастер форм»).

Система сама определит взаимосвязь таблиц, состав и структуру полей и создаст нужную форму по умолчанию (рис. 4.12).

Если вы хотите управлять процессом создания формы, то не указывайте родительскую таблицу.

§ 8. Вывод на печать характеристик и содержимого таблиц.

Индивидуальное задание для выполнения расчетно-графической работы (РГР)

Дальнейшие операции с БД целесообразно выполнять после внесения изменений в ее содержание в соответствии с вариантом, который определит для вас преподаватель. В качестве базового задания предлагается: добавить в таблицу «Культуры» две записи по культурам 1–ой группы со следующими данными:

Группа	Код культ.	Название	Созр. от (R)	Созр. до	Почвы от	Почвы до
1	ВИШН–РАНН	Вишня ранняя	06.20	07.10	$5,4 + 0,1*N$	$R + 0,5$
1	ВИШН–ОСЕН	Вишня осенняя	07.20	08.10	$5,8 + 0,1*N$	$R + 0,5$

где N — номер варианта (от 1 до 40), R — значение показателя «Почвы от». Значения остальных полей копируются из записи «ВИШН.–ОБЩ.» («Вишня»).

Преподаватель может изменить вид или вариант задания и выдать студенту другое задание, например, самостоятельно разработать базу данных определенного содержания.

В любом случае, вы должны сформировать отчет в соответствии с требованиями преподавателя и указаниями, изложенными в заключении.

Если вы выполняете задание по нашему варианту, то советуем формировать отчет по РГР параллельно с выполнением упражнений.

Все данные из таблиц, которые будут приводиться дальше, соответствуют варианту $N = 0$.

Поэтому, если преподаватель не задал вам вариант по данному практикуму, введите две записи в таблицу «Культуры» для «нулевого» варианта.

Замечание.

Содержимое таблиц можно «перебросить» в отчет Word-а стандартным способом. Следует выделить в БД открытую табли-

цу, ввести команду «Копировать», перейти в окно отчета по РГР и дать команду вставки.

Анализ описания структуры таблиц

Информация о таблицах является частью общего документирования БД, которое в MS Access выполняется с помощью специальной программы «Документатор» (или «Архивариус»). Эта программа анализирует БД и ее элементы и составляет специальный отчет, который можно просмотреть на экране и распечатать.

- Откройте окно «База данных», а все таблицы закройте.
- Введите команду «Работа с БД /раздел Анализ / кн. Архивариус». Откроется ДО «Архивариус».
- В ДО откройте вкладку «Таблицы» и включите индикаторы у таблиц «Группы» и «Культуры». Нажмите кн. «Параметры». В ДО «Печать описания таблицы» включите следующие опции:
 - «связи» в разделе «Включить в описание таблицы»;
 - «имена, типы данных и форматы» — в следующем разделе (свойства печатать не будем);
 - в разделе «Включить в описание индексов» — установите опцию «Ничего».
- После установки опций нажмите кн. «ОК» в текущем окне и в ДО «Архивариуса».

Через некоторое время документатор откроет окно предварительного просмотра «Описание объектов» и поместит в него описание структуры (2–4 листа).

- Внимательно просмотрите отчет, разберитесь с его содержанием. Если необходимо, сохраните отчет архивариуса в виде отдельного файла типа .rtf (нажмите ПКМ и введите команду: «Экспорт / Файл Word...»)

Пояснения по содержанию отчета здесь не приводятся.

Печать содержимого таблиц

Для предварительного анализа тех данных, которые мы ввели в нашу БД, можно распечатать содержимое таблиц. Эта операция выполняется на начальных этапах построения БД и служит для оценки структуры таблицы, характеристик и свойств полей, а

также для отработки различных запросов на небольшом количестве записей контрольного примера.

В дальнейшем для вывода на печать информации из БД разрабатываются и **создаются специальные формы отчетов**, где данные представляются в удобочитаемом виде.

Таблицы распечатываются в том виде (по составу столбцов и структуре), в котором они заданы в текущий момент.

- Выделите таблицу «Группы» и введите команду «кн. Office / Печать / Предварительный просмотр». Система выведет в окно предварительного просмотра таблицу.
- Используя инструменты масштабирования и прокрутки окна, посмотрите таблицу. Настройте макет страницы. В соответствующем диалоговом окне подберите параметры страницы так, чтобы таблица удобно располагалась на листе бумаги. Попробуйте установить альбомную ориентацию. При печати (книжной ориентации) вам понадобится 2–3 листа бумаги. Распечатайте таблицу или поместите ее в соответствующий раздел отчета по РГР.

§ 9. Сортировка, фильтрация и организация выборки данных

Общие сведения об операциях с данными

Одно из основных назначений СУБД (и элементов эксплуатации БД) — находить и выдавать пользователю ту информацию, которую он в данный момент желает получить.

Для этого пользователь должен сформулировать некоторый запрос, затем представить его в формальном виде, определить в СУБД условия отбора или сортировки записей и выполнить запрос.

Например, мы можем потребовать отсортировать данные в таблице «Культуры» по полю сроков созревания, или отобразить в таблице только те культуры, которые можно разместить на нейтральных почвах и близких к ним (с РН от 6,2 до 7,2), или найти в таблице запись по определенной культуре.

Для решения таких задач (поиска, отбора и отображения записей, удовлетворяющих определенным условиям в запросах) система предоставляет несколько способов.

Сортировка записей. Эта операция может быть выполнена как для всей таблицы, так и для записей, составляющих результат выполнения запроса (см. ниже).

Поиск. Эта операция позволяет найти в таблице нужную запись или группу записей.

Фильтрация. Позволяет найти и отобразить в таблице только те записи, которые удовлетворяют определенным условиям. Здесь мы можем воспользоваться следующими фильтрами.

- Фильтрами по значению и по «выделенному». Их называют также фильтрами по «образцу».

- Фильтром по форме. В этом фильтре можно использовать несколько критериев отбора записей (здесь не рассматривается).

- Расширенным фильтром (фильтром по условию). Это более гибкий инструмент фильтрации записей, но и более сложный в построении запроса.

Запрос на выборку. Данная форма отбора нужных записей имеет более сложный механизм выполнения операций фильтрации, но предоставляет пользователю и более мощный аппарат для выполнения запросов.

Сортировка таблицы и простейшие приемы фильтрации

Сортировка записей.

- Откройте таблицу «Культуры».

При открытии таблицы система выводит записи в последовательности, определенной значениями первичного ключа, который, в нашем случае состоит из полей «Код_Гр» и «Код_культ».

Если этот порядок нас не устраивает, то необходимо отсортировать таблицу по другим полям.

- Отсортируем нашу таблицу по начальному сроку созревания культур в возрастающем порядке значений этого поля.
- Установите курсор на любой записи в столбце «Созр. — от» или выделите этот столбец. В разделе «сортировка и фильтрация» меню «Главная» найдите и нажмите кн. «А-

Я» («Сортировка по возрастанию»). Посмотрите результаты сортировки. Для возврата к исходному состоянию нажмите кнопку «А-Я» (с «ластиком»).

- Повторите операцию с другими полями.

Поиск записей.

Допустим, нам надо быстро найти и посмотреть информацию о смородине (т.е. выполнить поиск записей по одному полю).

- Выделите поле «Название» или поместите в него курсор вставки.
- Введите команду «Найти». Откроется ДО «Поиск и замена» (в поле).
- Установите в этом ДО (на вкладке «Поиск») следующие параметры.
 - Введите в окошко «образец» первые буквы «смор» (без кавычек) и нажмите кн. «Больше». Откроются дополнительные опции.
 - В окошке «Просмотр» установите опцию «Все», т.е. выполнить поиск по всей таблице.
 - В окошке «Совпадение» установите значение «С начала поля».
 - Отключите индикаторы «С учетом регистра» и «С учетом формата полей» (иначе прописные и строчные буквы будут различаться).

Укажите «Поиск в» (поле) - «Название».

- Нажмите кн. «Найти далее». Если найденная запись загромождается окном, то перетащите его в другое место.
- Еще раз нажмите кн. «Найти далее». Система выделит в таблице БД другую запись со словом «Смородина».
- Закройте ДО «Поиск в поле...».
- Самостоятельно выполните поиск по другим полям (например, найдите записи с уровнем кислотности почв = 5,6).

Замечание.

С помощью опции «Совпадение» можно установить поиск по «Любой части поля». В этом случае, например, при образце «ня», будут высвечиваться записи со словами «Яблоня», «зимняя», «ранняя» и т.д.

- Выполните такой поиск.

Простая фильтрация по выделенному (по выбору)

Это самая простая фильтрация. Для ее выполнения необходимо установить образец, т.е. выделить в таблице нужный столбец, указать в нем значение, по которому надо отобрать записи, и ввести соответствующую команду.

Эта фильтрация не может работать с диапазонами значений полей и с набором разных полей.

Допустим, мы хотим работать только с записями, соответствующими первой группе культур «Деревья».

- Выделите в первом столбце «Код гр.» поле со значением 1 (щелкните внутри него ЛКМ).
- В разделе «Сортировка и ...» нажмите кн. «Выделение» и введите команду «Равно 1». Система отфильтрует записи и выведет на экран таблицу с записями первой группы. Далее в столбце «Название» аккуратно выделите слово «Яблоня», нажмите ПКМ. Откроется контекстное меню (рис. 4.13). Введите команду «Начинается с Яблоня». Система «сузит» диапазон записей до трех, оставив только записи по яблоням. Таким образом, данную команду можно повторять многократно для разных полей. Причем для текстовых полей можно задавать только часть значения.

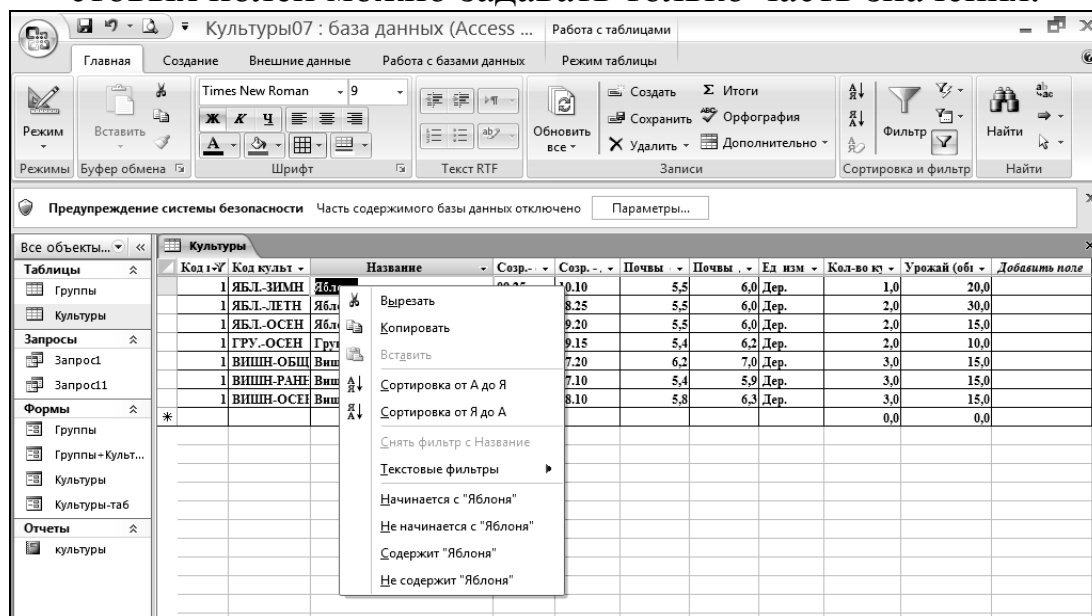


Рис. 4.13. Окно с меню для фильтрации по образцу

- Вернитесь к исходной таблице БД, удалив фильтр командами «Снять фильтр» или «Удалить фильтр». Между эти-

ми командами существует различие. Первая – полностью удаляет фильтр. Вторая только отключает его. В этом случае этот фильтр можно использовать многократно.

- Способ фильтрации «по значению» рассмотрите самостоятельно. Для его использования надо выделить столбец, щелкнуть на треугольнике в названии столбца. Откроется ДО отбора значений полей. Выделите нужные значения и нажмите «ОК».

Использование расширенного фильтра.

Фильтр данного вида используется для быстрого построения условий отбора записей при использовании нескольких критериев и полей (при использовании логических выражений).

Замечание.

Определение логических (булевских) выражений будет рассмотрено в следующем разделе.

Допустим, мы хотим отобрать и распечатать записи по культурам, для которых почва должна иметь слабокислую, нейтральную или близкую к ней реакцию с рН от $P1 = 5,6$ до $P2 = 7,2$ ($P1$ и $P2$ — условные обозначения, которые мы используем в постановке данной задачи). Из методических соображений потребуем, чтобы нормативный диапазон, т.е. соответствующие значения полей «Почвы1» («Почвы_от») и «Почвы2» («Почвы_до»), полностью включал в себя заданный. Таким образом, критерий отбора будет иметь следующий вид: если ($Почвы1 \geq P1$) И ($Почвы2 \leq P2$), то данная запись удовлетворяет нашему условию.

Используем для решения данной задачи «Расширенный фильтр».

- Откройте таблицу «Культуры» и удалите фильтры, если они не удалялись.
- Введите команду «"Главная" / раздел "Сортировка и..." / кн. "Параметры расш. фильтра" / команда "Расширенный фильтр"». Откроется специальное ДО (см. рис. 4.14) с бланком запроса (на рисунке отображается состояние окна после определения условий).
- Изучите состояние окна. В верхней части расположено окошко с перечнем полей таблицы «Культуры». В нижней

– окно бланка для построения запроса. В бланке находятся строки: «Поле», «Сортировка», «Условия отбора» и множество строк типа «Или». В ячейках «Поле» должны быть указаны имена полей, участвующих в логических выражениях и для сортировки отображаемой таблицы (результата выполнения запроса).

- Если в бланке остались данные от предыдущего фильтра, то нажмите ПКМ и введите команду «Очистить бланк».
- В строку «Поле», используя открывающиеся списки, включите поля: «Почвы1», «Почвы2», «Код_Гр» и «Код_культ». Установите для последних полей сортировку «по возрастанию».

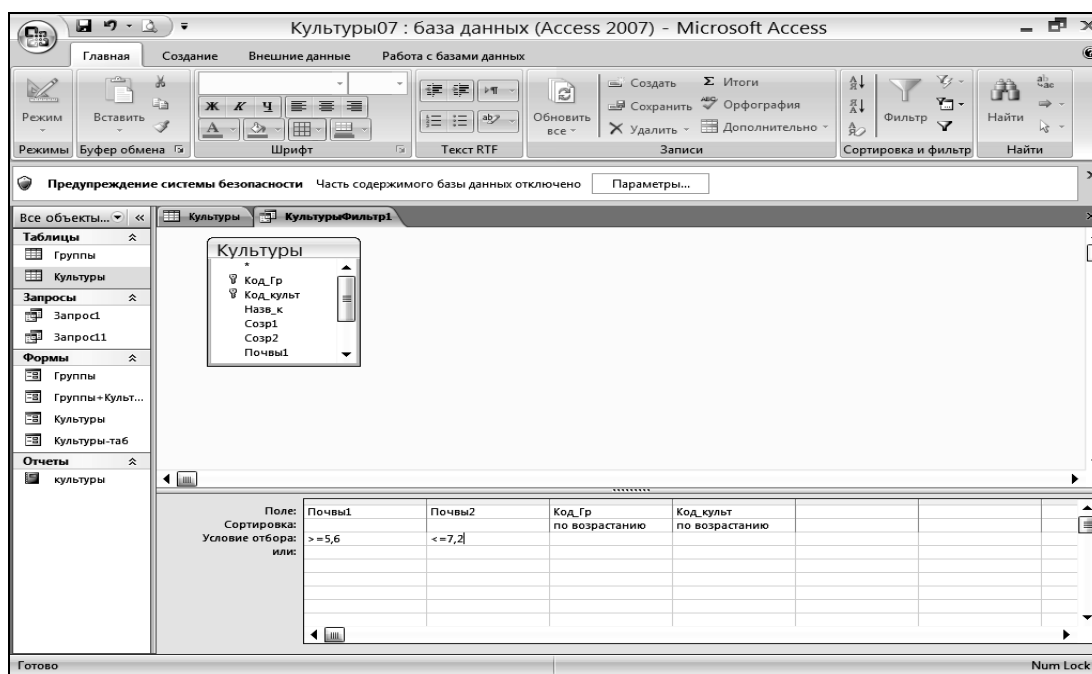


Рис. 4.14. Окно конструктора расширенного фильтра

- Установите для полей «Почвы1» и «Почвы2» критерии отбора записей в строку «Условие отбора». Для «Почвы1» введите (без кавычек) «>=5,6», для «Почвы2» введите «<=7,2».

Код к	Код культ	Название	Созр.-	Созр.-	Почвы	Почвы	Ел. изм.	Кол-во ку	Урожай (об
	1	ВИШН-ОБЩ	07.10	07.20	6,2	7,0	Дер.	3,0	15,0
	2	ВИШН-ОСЕН	07.20	08.10	5,8	6,3	Дер.	3,0	15,0
	2	СМОР-КРАС	07.15	07.25	5,6	6,0	Куст.	2,0	5,0
	2	СМОР-ЧЕРН	07.20	07.25	5,6	6,0	Куст.	4,0	8,0
	4	КАП.-Б.Р.	07.20	08.10	6,5	7,2	Кв.м.	20,0	50,0
	4	МОФК-ОБЩ	07.10	09.20	6,0	7,0	Кв.м.	5,0	50,0
	5	КАРТ-ОБЩ	08.10	09.15	6,0	6,5	Кв.м.	100,0	250,0
	5	ТОМ.-ПОЗД	08.25	09.20	6,2	6,9	Кв.м.	30,0	50,0
	5	ТОМ.-РАН.	08.15	08.25	6,3	6,8	Кв.м.	40,0	30,0
	*							0,0	0,0

Рис. 4.15. Результат выполнения расширенного фильтра

- Выполните фильтрацию и оцените результат (см. рис. 4.15). Если этот фильтр надо сохранить, то щелкните ПКМ на заголовке вкладки и введите команду «Сохранить» и далее укажите имя фильтра.
- Удалите фильтр (физически он остается до создания следующего фильтра). Закройте таблицу.

§ 10. Создание и выполнение запросов на выборку данных

В отличие от простых способов фильтрации запрос на выборку предоставляет пользователю более мощный и универсальный инструмент для поиска и отбора нужных сведений из базы данных.

Предыдущие способы фильтрации обладают существенными ограничениями:

- фильтр может работать только с одной таблицей;
- выводятся все столбцы таблицы;
- фильтр может выполнить относительно простую операцию фильтрации (например, без использования вычисляемых полей).

При использовании запросов можно:

- использовать более одной таблицы;
- сформировать таблицу запроса с использованием только необходимых полей;
- располагать поля в необходимой последовательности.
- ограничить количество исходных записей;
- использовать вычисляемые поля.

Сформированный запрос может быть сохранен как объект БД (аналогично таблицам, формам, отчетам и др.). Результат выполнения запроса помещается в отдельную таблицу, структура которой определяется параметрами установленными при создании запроса.

Конструирование запроса.

Пусть нам необходимо получить (и распечатать) справочную информацию по двум группам культур : «Деревья» и «Кустарники» со сроком созревания после «07.18».

При отображении таблицы после выполнения запроса мы не будем включать поля с фактическими данными (поля «Урожай» и «Кол-во»), но добавим вычисляемое поле «Урож-ть» (характеризует урожайность культуры), значение которого вычисляется следующим образом: «Урож-ть» = «Урожай» / «Кол-во». Поля с кодом группы и кодом культуры будут использованы только для отбора записей, но выводиться на экран не будут. Вместо кода группы будет выводиться ее обозначение из таблицы «Группы». Запрос будем создавать на основе двух наших таблиц.

- Закройте все таблицы. На вкладке «Создание» в разделе «Другие» щелкните на кн. «Конструктор запросов». Система выведет на экран окно конструктора и на его фоне ДО «Добавление таблицы». Если верхнее поле конструктора содержит окошки таблиц с установленными связями, то добавлять таблицы не надо, иначе в этом ДО последовательно выберите обе наши таблицы (выделить и нажать кн. «Добавить»). Закройте ДО добавления таблиц. В результате мы попадаем в конструктор. В главном меню появилась вкладка «Конструктор»

Окно конструктора состоит из двух частей (см. рис. 4.16):

- в верхней части находятся окошки таблиц с перечнем полей;
- в нижней части — бланк конструирования запроса.

В каждом столбце бланка мы определим поле таблицы и вид сортировки, если хотим упорядочить записи по этому полю, установим признак вывода поля на экран и условия, которые определяют критерий отбора записей по полю.

Первым шагом при создании запроса является выбор полей, которые должны быть включены в таблицу запроса.

Выполнить эту операцию можно разными способами. Мы используем самый простой – перевод (перетаскивание) поля из окошка таблицы в бланк запроса (по аналогии с установкой связи).

При переводе указатель мыши превращается в прямоугольник.

- Переведите в строку «Поле» бланка: из таблицы «Группы» поле «Обозн»; из таб. «Культуры» поля: «Назв_к», «Созр1», «Созр2», «Почвы1», «Почвы2», «Ед_измер». «Код_Гр», «Код_Культ» (два последних поля будут после создания вычисляемого поля перемещены см. рис. 4.16)

Замечания

1. Если вы ошиблись при определении поля в бланке, то выделите это поле в соответствующем столбце бланка и удалите кн. Delete. После этого повторите определение поля заново.

2. Если вы не сохраняли запрос и не дали ему имя при закрытии (даже то, которое предлагает система), то после выхода из системы запрос будет потерян.

3. Поля, выводимые в наборе записей запроса, наследуют свойства, заданные в исходной таблице, но вы можете переопределить некоторые из них: «Описание», «Формат поля», «Маску ввода», «Подпись». Чтобы задать новые свойства некоторого поля, выделите его в бланке и введите команду «Свойства», используя контекстное меню или команду «Страница свойств» из раздела «Показать или скрыть». На экране появится окно свойств, где необходимо указать необходимые свойства.

Создадим вычисляемое поле «Урожайность».

- Установите курсор в клетку «поле» нового столбца введите арифметическое выражение:
Урож-ть:[Культуры]![Урожай]/[Культуры]![Кол-во], где «Урож-ть» – условное название нового поля. После знака

равенства записано арифметическое выражение (на языке системы), определяющее частное от деления значения поля «Урожай» на значение поля «Кол-во» из таблицы «Культуры». Мы указываем имя таблицы «Культуры» явно, т.к. не включили соответствующие поля в бланк запроса.

Переместите при помощи мыши вычисляемое поле влево (рис. 4.16) и включите его отображение на экране.

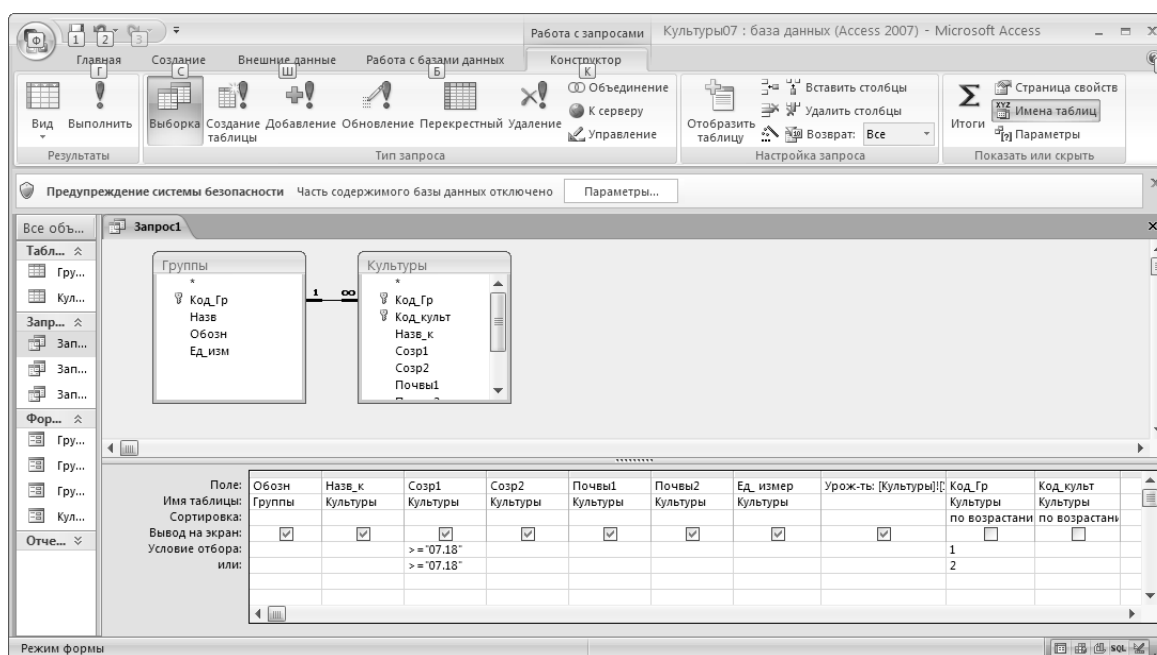
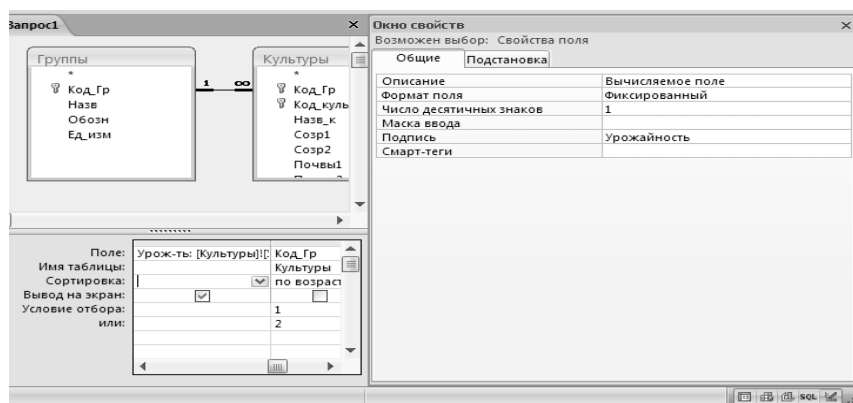


Рис. 4.16. Окно конструктора запросов с условиями.

- Для проверки правильности построения запроса на данном этапе сначала сохраните запрос с именем «Запрос1» (кн. «Сохранить» на панели быстрого доступа). Выполните запрос: кн. «Выполнить» в разделе «Результаты». Оцените свои построения и обратите внимание на заголовок и значения вычисляемого поля. Если заметили ошибку, удалите выражение и введите еще раз, соблюдая правильный ввод имен таблицы, полей и знаков.

Если вас не устраивает заголовок столбца и формат отображения значений, то вы должны переопределить соответствующие свойства. Для этого вернитесь в конструктор запроса.

- Для изменения свойств поля «Урожай-ть» откройте ДО свойств поля. Щелкните ПКМ на поле и в меню выберите пункт «свойства». Откроется окно свойств:



- На вкладке «Общие» установите: «Формат» = «фиксированный»; «Число дес. знаков» = 1; «Подпись» = «Урожайность» (без кавычек). Закройте ДО и еще раз проверьте результат изменений в режиме таблицы. Вернитесь в конструктор.

На следующем шаге надо определить, какие поля будут отображаться в таблице и по каким полям следует проводить сортировку.

Для «отключения» полей при отображении результата необходимо, щелкнув ЛКМ, убрать галочку в строке «Вывод на экран».

- Отключите вывод на экран кода группы (поле «Код_Гр») и кода культуры («Код_культ»).

Для определения вида сортировки надо перейти к соответствующему полю бланка, открыть список и выбрать вид сортировки.

- Укажите в строке «Сортировка» упорядочение в возрастающем порядке по полям «Код_Гр» и «Код_культ».
- Перед выполнением следующих упражнений сохраните запрос. В дальнейшем выполняйте эту операцию регулярно.

Замечание.

Если вы закроете запрос и опять откроете, то неотображаемые поля в бланке будут смещены вправо и помещены после поля «Урожайность».

Конструирование критерия отбора записей.

Критерий формируется заданием условий на значения полей таблицы. Ввод условия (логического выражения) аналогичен заданию условия на значение для поля таблицы при ее создании.

Логические выражения.

Условие, или, более правильно, **логическое выражение (ЛВ)** служит для проверки истинности или ложности какого-либо утверждения, которое формально записывается с помощью операций сравнения и логических (булевских) операторов.

В общем виде, простейшее ЛВ можно записать так: **V1 ОП V2**, где V1 и V2 — выражения числового или текстового типа или поля таблицы (V1 и V2 называются операндами); ОП — операция сравнения из множества: { = — равно, > — больше, < — меньше, >= — больше или равно, <= — меньше или равно, <> — не равно }.

Более сложные ЛВ формируются с помощью логических операторов:

- AND (And) — логическое «И»;
- OR (Or) — логическое «ИЛИ».

Выражение A AND B имеет значение «Истина», только в том случае, если и A и B имеют истинное значение. Во всех остальных случаях значение выражения — «Ложно».

Выражение A OR B имеет значение «Ложно» только тогда, когда ложны значения обоих операндов (A и B).

Вид логического выражения, которое используется в условиях отбора записей, немного отличается от общего. Дело в том, что в качестве одного из операндов (обычно V1) выступает поле таблицы. Поэтому выражение получается «укороченным». Например, если надо отобразить записи, соответствующие огородным культурам (двухлетним и однолетним), то в соответствующей строке бланка поля «Код_Гр» необходимо записать условие (без кавычек): « = 4 Or = 5 ».

Пример 1. Введите условие « = 4 Or = 5 ». в соответствующую строку поля и выполните запрос.

Оцените действие команды и вернитесь в режим конструктора. Удалите условие (сотрите его кл. «Delete»).

Если в условии отбора включается несколько полей и все ЛВ (для каждого поля) находятся в одной строке, то все они объединяются логической операцией AND. Запись отбирается в запрос, если значение всего ЛВ принимает значение «Истина».

Пример 2. Выполним отбор записей, соответствующих первой группе («Деревья») со сроком созревания от 08.01 и $pH \geq 5,5$.

Для этого введите следующие условия (все условия вводятся без кавычек): в поле «Код_Гр» условие «=1», в поле «Созр1» — условие « ≥ 08.01 », в поле «Почвы1» — условие « $\geq 5,5$ ». Выполните запрос. Оцените результат. Сотрите условия в бланке запроса.

Если для поля задается несколько условий, объединенных операцией «ИЛИ» (OR), то каждое условие записывается в отдельную строку «ИЛИ» бланка.

Пример 3. Установите первое условие, используя второй способ: в первую строку «Условие отбора» введите (для поля «Код_Гр») «=4», а во вторую строку («ИЛИ») — условие «=5». Выполните запрос, оцените результат и удалите условия из бланка.

Если в условиях отбора необходимо указать диапазон значений, то логическое выражение формируется с помощью операций сравнения и логического оператора And.

Например, если необходимо отобрать записи с культурами, у которых значения поля «Почвы от» («Почвы1») находятся в диапазоне от 5,6 до 6,2, то мы должны ввести условие « $\geq 5,6$ And $\leq 6,2$ ».

Введите это условие и выполните запрос. Удалите условие.

Основной запрос

Сформируем условия отбора для исходного запроса.

По полю «Код группы» («Код_Гр») мы должны отобрать записи со значениями равными 1 или 2.

Поле «Срок созревания от...» («Созр1») — текстовое. Нижняя граница, определяющая условие отбора может быть записана в виде « ≥ 07.18 ».

Полное логическое выражение, соответствующее нашему случаю, будет иметь вид:

(Код_Гр =1 OR Код_Гр =2) AND (Созр1 >= 07.18).

Из методических соображений мы преобразуем его к виду:

(Код_Гр=1 AND Созр1 >= 07.18) OR (Код_Гр=2 AND Созр1 >= 07.18).

- Введем условия в бланк запроса:
- в соответствующей строке поля «Код_Гр» укажите «=1» и в поле «Созр1» >=07.18;
- перейдите в строку «ИЛИ» и укажите в соответствующих полях «=2» и «>=07.18».
- Выполните запрос. Результат представлен в таблице 4.10. Запрос сохраните.

Обратите внимание, что данная таблица перенесена в Word стандартным способом: выделяем таблицу в Access, затем в контекстном меню выбираем команду «копировать», переходим в Word и вводим команду «Вставить».

Замечание.

При такой вставке таблицы в Word возможны некоторые изменения в отображении количества десятичных знаков. Это связано с настройкой самого текстового редактора.

Таблица 4.10.

Результат выполнения запроса на выборку

Запрос1							
Обозн. Гр.	Название	Созр.-от	Созр. - до	Почвы_от	Почвы_до	Ед_измер	Урожай-ность
ДЕР.	Вишня осенняя	07.20	08.10	5,80	6,30	Дер.	5,00
ДЕР.	Груша осенняя	09.20	09.15	5,40	6,20	Дер.	5,00
ДЕР.	Яблоня зимняя	09.25	10.10	5,50	6,00	Дер.	20,00
ДЕР.	Яблоня летняя	08.01	08.25	5,50	6,00	Дер.	15,00
ДЕР.	Яблоня осенняя	09.10	09.20	5,50	6,00	Дер.	7,50
КУСТ.	Смородина черная	07.20	07.25	5,60	6,00	Куст.	2,00

§ 11. Создание отчетов

Основные сведения об отчетах

Отчет является объектом БД (так же, как и форма) и используется для организации и представления данных в виде печатного документа.

По сравнению с «распечаткой» содержимого базовой таблицы (или запроса) отчет обладает рядом специфических свойств.

- Он предоставляет широкие возможности для группирования и вычисления промежуточных и общих итогов.
- Отчет может быть красиво оформлен с помощью дополнительных средств. Например, каждая группа записей может иметь отдельные заголовки и примечания, можно вставить дополнительные надписи и рисунки и т.д.
- В отчете можно производить сложные вычисления не только внутри групп, но и по нескольким группам одновременно.

Структура отчета

Отчет содержит несколько разделов, которые могут иметь разное содержание и различные свойства.

1. Заголовок отчета. Содержит общее название документа и печатается только один раз.

2. Верхний колонтитул. Содержит сведения, которые необходимо печатать на каждой странице отчета в верхней части страницы (дату, краткое название отчета, номер страницы и т.д.). Если отчет должен иметь форму таблицы и не содержит группированных данных, то в этой части можно разместить шапку таблицы.

3. Заголовок группы. Используется, если в отчете выполняется группировка записей и начало каждой группы должно содержать какие-то данные. Часто здесь повторяют шапку таблицы (в этом случае в колонтитуле ее не используют).

4. Область данных. Этот раздел присутствует в любом отчете и содержит значения полей записи таблицы (или запроса).

5. Примечание группы. Используется для вычисления и отображения итоговых значений по полям записей, которые объединяются в группы. Если в отчете вводится группировка с под-

ведением итогов, то этот раздел должен присутствовать обязательно.

6. Нижний колонтитул. Используется аналогично верхнему колонтитулу для вывода дополнительной информации в нижней части страницы.

7. Примечание отчета. Этот раздел используется для отображения информации, которая должна выводиться в самом конце отчета (один раз). Часто сюда помещают общие итоговые данные и примечания.

Замечание.

Наличие разделов в отчете определяется способом создания и наличием группировки данных. Пустой макет отчета, если его создавать вручную, содержит только разделы колонтитулов и область данных. Остальные разделы добавляются по мере необходимости.

Способы создания отчета

Создание отчета может выполняться разными способами и с использованием «мастеров» разного уровня.

1. Вручную, с помощью **«Конструктора»**. Это самый универсальный, но и самый трудоемкий способ, требующий высокой квалификации и хорошего знания структуры БД.

2. С использованием **«Мастера отчетов»**. Этот способ позволяет быстро создать хороший отчет для дальнейшей модификации в конструкторе с использованием информации из разных таблиц.

3. С использованием **«Автоотчета»**. Он строится на основе одной таблицы с помощью кнопки «Отчет» на вкладке «Создание». В результате мы увидим аналог таблицы с некоторыми элементами дополнительного форматирования. Такой отчет очень скудный по оформлению, но не требует сложных манипуляций с параметрами.

- Откройте таблицу «Культуры» и создайте отчет этим способом. Изучите его форму и содержание. Посмотрите элементы управления окна предварительного просмотра.

4. С использованием **«Мастера почтовых наклеек»**. Этот способ позволяет сформировать отчет специального вида для распечатки данных таблицы в виде отдельных карточек (накле-

ек). При этом мастер предлагает выбрать один из распространенных шаблонов.

Отчеты созданные в режимах «мастера» и «автоотчета» можно изменять в режимах «макета» или «конструктора» (по аналогии с формами).

Создание отчета с помощью мастера.

В нашу задачу входит создать отчет для вывода на печать данных из таблицы «Культуры» Для сортировки и группировки записей по группам культур используем поле «Назв» из таблицы «Группы». По некоторым числовым полям вычислим значения некоторых функций (средних, максимальных, минимальных и итоговых).

- Закройте таблицы.
- В меню «Создание в разделе «Отчеты» нажмите кн. Мастер отчетов».

На экране появится знакомое (по построению формы) окно «Создание отчетов». Далее мастер будет последовательно выводить эти окна с инструкциями для конструирования отчета.

Процесс создания состоит из ряда шагов.

Шаг 1. Окно выбора полей (рис. 4.17).

В окошке «Таблицы и запросы» установите имя таблицы «Группы» и переведите в правое окно поле «Назв». Затем укажите т. «Культуры и переместите (кнопкой «>>») все поля в окно «Выбранные поля». Затем кнопкой «<<» верните обратно поле «Код_Гр». Нажмите кн. «Далее». Откроется следующее окно мастера «Определение вида представления данных»

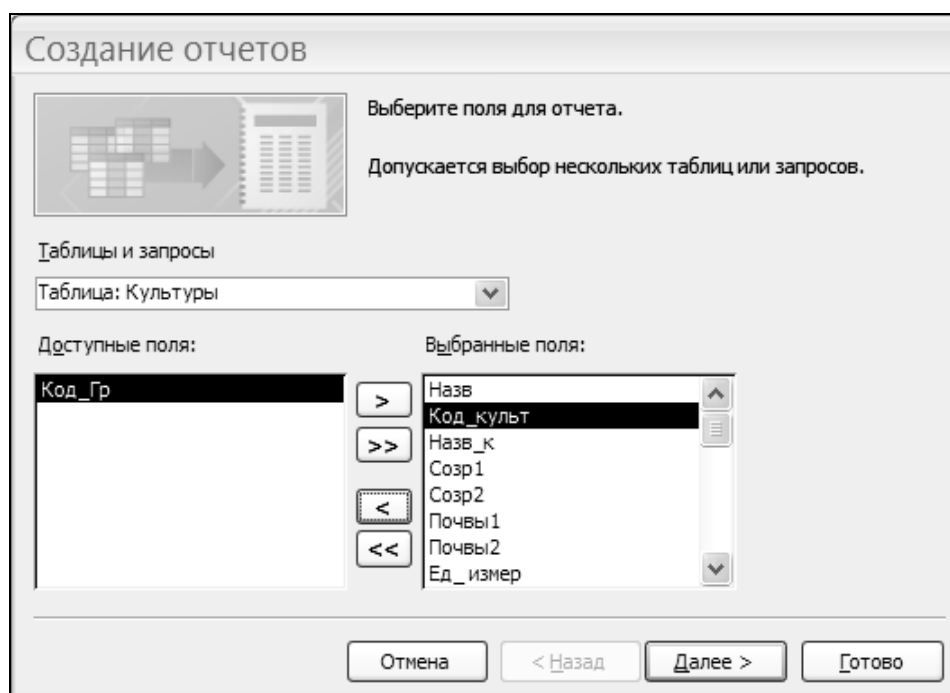


Рис. 4.17. Окно мастера отчетов для выбора полей

Шаг 2. Определение вида представления данных и установка уровня (порядка) группировки.

В этом окне слева представлен перечень таблиц, справа – перечень выбранных полей. Установите курсор на таблицу «Группы» и Вы увидите уровни группировки отчета и поля, которые определяют группировку (в нашем случае это поле «Назв») Нажмите кн. «Далее».

В следующем окне можно добавить уровни группировки по другим полям, но нам этого делать не надо. Нажмите кн. «Далее». Откроется следующее окно мастера.

Шаг 3. Дополнительные установки по сортировке записей.

В этом окне мы установим дополнительную сортировку записей внутри группы по полю «Код_культ». Для этого откройте список полей в строке ввода (под № 1) и установите это поле. Порядок сортировки оставьте от А до Я.

Нажмите кн. «Итоги...». Откроется дополнительное ДО «Итоги» (рис. 4.18).

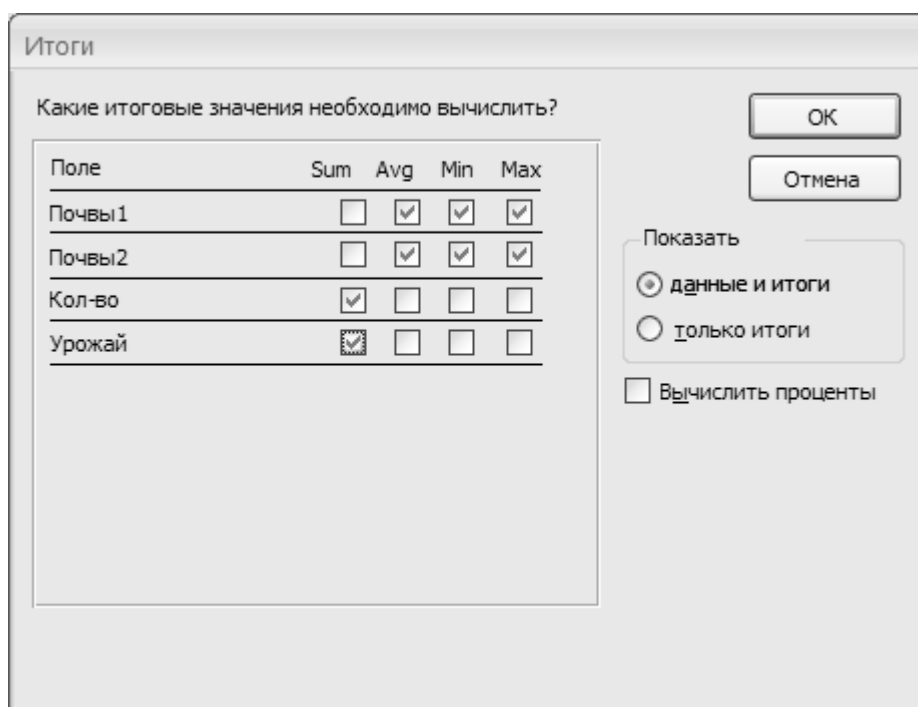


Рис. 4.18. Окно для определения итогов

Шаг 4. Установка видов итоговых значений по группам.

В ДО «Итоги» перечислены поля (числового типа), по которым в разделе примечаний группы отчета можно вычислить и вывести значения ряда функций.

Включите индикаторы суммирования (Sum) по полям «Кол-во» и «Урожай». Включите также индикаторы для вычисления функций Avg, Min, Max для полей «Почвы1» и «Почвы2». В разделе «Показать» должна быть включена опция «Данные и итоги». Опция «Вычислять проценты» должна быть отключена. После этих установок нажмите кн. «ОК». Вы опять вернетесь в окно шага 3. Нажмите кн. «Далее».

Шаг 5. Выбор макета отчета.

Здесь система предлагает выбрать: ориентацию листа бумаги (книжную или альбомную); макет из предлагаемого списка; дополнительные параметры.

Установите книжную ориентацию и макет типа «Ступенчатый». Включите опцию «Настроить ширину полей...» и нажмите кн. «Далее».

Шаг 6. В новом окне система предлагает выбрать стиль отчета. Выберите стиль «Стандартный» и нажмите кн. «Далее».

Шаг 7. Последнее окно создания отчета позволяет ввести имя отчета. Введите имя «Культуры», включите опцию «Просмотр отчета» и нажмите кн. «Готово».

Система перейдет в режим предварительного просмотра, где вы можете оценить результаты вашей работы и вывести отчет на печать.

Если вы все сделали правильно, то при распечатке вы получите отчет (БД) фрагмент которого представлен на рис. 4.19.

The screenshot shows the Microsoft Access interface with a report titled 'Культуры'. A dialog box 'Экспорт в файл RTF' is open, indicating the export of the selected object to an RTF file format. The report table contains the following data:

Назв. гр.	Код культ.	Название	Созр.-от	Созр.-до	Почвы_от	Почвы_до	Ед_измер	Кол-во культ.	Урожай (общ)
Деревья									
	ВИШН-ОБИ	Вишня	07.10	07.20	6,2	7,0	Дер.	3,0	15,0
	ВИШН-ОСЕ	Вишня осенняя	07.20	08.10	5,8	6,3	Дер.	3,0	15,0
	ВИШН-РАН	Вишня ранняя	06.20	07.10	5,4	5,9	Дер.	3,0	15,0
	ГРУ-ОСЕН	Груша осенняя	09.20	09.15	5,4	6,2	Дер.	2,0	10,0
	ЯБЛ.-ЗИМН	Яблоня зимняя	09.25	10.10	5,5	6,0	Дер.	1,0	20,0
	ЯБЛ.-ЛЕТН	Яблоня летняя	08.01	08.25	5,5	6,0	Дер.	2,0	30,0
	ЯБЛ.-ОСЕН	Яблоня осенняя	09.10	09.20	5,5	6,0	Дер.	2,0	15,0
Итого для 'Код_Гр' = 1 (7 записей)									
	Sum							16,0	#
	Avg				5,6	6,2			
	Min				5,4	5,9			
	Max				6,2	7,0			
Кустарни									
	СМОР-КРАС	Смородина красная	07.15	07.25	5,6	6,0	Куст.	2,0	5,0
	СМОР-ЧЕРН	Смородина черная	07.20	07.25	5,6	6,0	Куст.	4,0	8,0
Итого для 'Код_Гр' = 2 (2 записей)									
	Sum							6,0	#
	Avg				5,6	6,0			
	Min				5,6	6,0			
	Max				5,6	6,0			
Овощны									

Рис. 4.19. Фрагмент окна с отчетом и меню экспорта

В окне предварительного просмотра вы можете выполнить различные действия по просмотру, форматированию, печати, а также экспортировать отчет в разные программы (Excel, Share Point, Word и другие).

На рисунке показано меню экспортирования отчета в файл формата RTF, который может быть использован для вставки отчета в окно документа текстового редактора (например, Word).

- Постарайтесь поработать с элементами форматирования окна предварительно просмотра, распечатать отчет и экспортировать его в документ Word-а.

- Советуем также посмотреть конструктор отчета и разобраться с его работой.

Заключение

В данной главе мы рассмотрели только часть этапов проектирования и создания базы данных в среде СУБД MS Access.

Ряд элементов СУБД, связанных с автоматизацией работы БД и ее компонентов (например, создание кнопочных форм (типа «меню»), синхронизацию работы форм БД, элементы программирования и др.), мы оставили без внимания. Вы можете познакомиться с ними по соответствующей литературе.

Далее мы приводим материал, который поможет вам выполнить индивидуальную работу и создать отчет по РГР по данной теме.

Требования к порядку выполнения и содержанию индивидуальной расчетно-графической работе (РГР).

Для выполнения индивидуальной работы необходимо получить у преподавателя задание с постановкой задачи на создание БД определенного содержания и структуры.

При выполнении индивидуального задания студент должен выполнить следующее.

- Сделать постановку задачи по созданию базы данных (БД) и внести необходимые исправления по указанному варианту.
- Разработать структуру таблиц и базы данных с использованием нормализации и согласовать с преподавателем.
- Создать базу данных (таблицы и связи) с указанной структурой.
- Заполнить БД необходимым количеством записей (не менее 15).
- Распечатать (или сохранить в отдельных файлах например, типа .rtf) информацию о БД и содержимое таблиц БД и затем включить их в отчет по РГР.
- Создать элементы эксплуатации (формы и отчеты).
- Выполнить сортировку и фильтрацию. Результаты сортировки и фильтрации вставить в отчет.
- Создать запросы на выборку и выполнить их. Результаты поместить в отчет.

- Создать отчет СУБД по таблицам или запросам.
- Составить и оформить отчет о работе по созданию БД и ее элементов в среде редактора «Word».

База данных и все необходимые данные (в том числе и отчет) должны находиться на личной дискете, которая предъявляется преподавателю при защите работы.

Рекомендуем, получив задание, сразу создать файл, титульный лист и первый раздел «Постановка задачи...», написать (используя редактор «Word») из задания исходную формулировку задачи с таблицей, внести в нее необходимые изменения, соответствующие варианту и добавить свои записи.

Используя процесс нормализации, доработать структуру таблиц и структуру базы данных. Заранее наметить способы и методы решения задач эксплуатации (критерии сортировки, фильтрации, запросов и структуры форм и отчетов).

Установить типы столбцов, точность представления данных, форму и вид контроля значений данных при вводе, значения по умолчанию, маски и т.д.

Далее необходимо оформить таблицы с характеристиками полей (например, как показано в таблицах (а) и (б)).

Таблица (а)

Типы и описание полей таблицы:

Имя поля	Тип данных	Описание

Таблица (б).

Свойства полей таблицы:

Имя поля	Размер и формат	Маска ввода	Подпись	Значен. по ум.	Условие на значение.	Обяз. поле
1	2	3	4	5	6	7

Распечатать таблицу с данными и с характеристиками полей и представить материалы преподавателю для утверждения.

После утверждения варианта постановки задачи и структуры таблиц, студент приступает к созданию базы данных и средств ее эксплуатации.

Для оценки работы, студент предъявляет преподавателю отчет и диск с базой данных

Отчет должен содержать следующие разделы.

- **Титульный лист с указанием:**
- названия факультета, номера группы, фамилии И.О. студента;
- номера задачи и название задачи и варианта.
- должности, звания и Ф. И. О. преподавателя.
- даты написания и защиты отчета.
- **Текстовую часть.**

Текстовая часть должна содержать следующие разделы.

1. Описание постановки задачи.

В разделе дается краткое описание задания и приводятся следующие материалы.

- Исходная форма таблицы с внесенными изменениями по варианту с полным набором записей (контрольными и своими), а также краткое описание процесса нормализации и его результат: таблицы с данными и структурная схема взаимосвязей таблиц. При представлении нормализованных таблиц в шапке каждой таблицы указываются исходные названия столбцов и имена полей, которые будут использованы в БД.
- Характеристика полей таблиц (см. выше). Таблицы могут быть дополнены необходимым текстовым описанием особенностей свойств отдельных полей.

2. Описание создания базы данных.

В этом разделе приводятся:

- имя БД и место ее хранения;
- описание особенностей создания БД;
- описание изменений настроек среды и макета таблицы;
- описание результата анализа структуры таблицы с использованием «Архивариуса». Все необходимые установки для «Архивариуса» можно выполнить по аналогии с указаниями параграфа 8. Распечатку отчета «Архивариуса» поместить в отчет (рекомендации см. ниже);
- вид и содержимое таблиц с полными данными.

3. Описание и вид форм для ввода и редактирования данных.

4. Сортировка и фильтрация.

В этом разделе Вы должны привести описание критериев сортировки, фильтрации и распечатки соответствующие выполненным операциям.

Как минимум, в отчет необходимо включить:

- одну распечатку с отсортированными данными;
- одну распечатку с отфильтрованными данными.

5. Создание и выполнение запросов.

В этот раздел необходимо включить:

- исходное описание запроса и формализованный критерий отбора записей;
- вид критерия для бланка запроса;
- распечатку результата выполнения запроса на выборку данных.

6. Описание и вид отчетов по БД.

Внутреннее оформление отчета (использование шаблонов, стилей, рисунков и т.д.) определяет сам студент.

Замечания по переносу информации из Access в Word.

Для вставки в отчет по РГР содержимого таблиц, элементов эксплуатации (например, форм, отчетов, бланков запросов и др.) можно использовать разные способы.

- Для вставки таблиц можно использовать способ, изложенный в параграфе 10 (см. таблицу 4.10).
- Для вставки изображений форм и бланков запросов можно использовать стандартный способ «фотографирования», используя комбинацию клавиш «Alt+Print Screen».
- Результаты «Архивариуса» и отчета БД могут быть размещены в приложениях в виде распечаток или экспортированы в Word с помощью соответствующего инструментария (вкладка «Внешние данные» главного меню). При таком способе экспортирования вид и содержание документа могут быть искажены. При экспортировании результат помещается в промежуточный файл формата .rtf, из которого надо данные скопировать в отчет.

Глава 5. Основы работы в среде Интернет

§ 1. Введение

Интернет — глобальная сеть, состоящая из каналов связи и управляющих компьютеров, коммутирующих эти каналы связи с целью быстрой передачи цифровой информации из одной точки Земли в другую. Терминалами Интернета, как правило, являются персональные компьютеры.

В последнее десятилетие Интернет буквально изменил жизнь большей части населения Земли. Что дает Интернет? С помощью Интернета можно получить доступ к самой разной информации. В частности, можно найти полезные программы для своего компьютера. Можно также распространить информацию о себе как специалисте или, наоборот, найти нужного сотрудника для своей фирмы.

Интернетом можно пользоваться для пересылки почты, причем не только текстов. Можно пересылать изображения, звуки, другими словами, — любую информацию, которую научились кодировать цифрами. По Интернету можно слушать радио и смотреть телевизионные программы. Благодаря Интернету появились целые научно-практические направления, например телемедицина.

Под телемедициной понимается система консультативно-диагностической медицинской помощи, при которой пациент и врач, непосредственно проводящий обследование пациента, находятся на значительном расстоянии от врача-консультанта. При этом для получения, обработки и пересылки результатов обследования используются специальное компьютеризированное цифровое медицинское оборудование и средства Интернета.

Появились даже интернет-магазины. Они позволяют изучать предложения и делать покупки в спокойной обстановке, сидя у себя дома. Интернет дает возможность создания клубов по интересам, в частности профессиональным, причем члены клуба могут быть разделены тысячами километров.

Основные аспекты использования Интернета в деятельности московских банков:

- сотрудники банков используют Интернет для получения курсов валют, макроэкономических и финансовых показателей, деловых и политических новостей, различных аналитических обзоров, а также сведений о конкурентах, потенциальных клиентах и изменениях в законодательстве;

- банки используют Интернет для привлечения новых клиентов, размещая на своих сайтах (т. е. в собственных разделах Интернета) информацию о предоставляемых услугах, тарифах и своей политике;

- банки на своих сайтах часто размещают информацию, необходимую их клиентам (биржевые котировки, деловые и политические новости, различные справочники);

- некоторые банки на своих сайтах создают системы, предоставляющие клиентам возможность проводить операции по своим счетам и торговать на биржах, а также системы просмотра выписок по расчетным счетам и банковским картам.

Количество пользователей Интернета в России превысило 40 миллионов человек.

Изучение принципов работы в Интернете с помощью данного учебного пособия возможно лишь в процессе работы на IBM PC-совместимом компьютере, подсоединенном к Интернету и работающем под управлением операционной системы Windows XP, в состав которой входят программы Internet

Explorer и Windows Explorer (Проводник). Необходим также текстовый процессор Word 2007.

В данной главе отсутствуют изображения окон и пиктограмм, т. к. при работе на компьютере в этих иллюстрациях нет необходимости. Напомним, что при отсутствии имени пиктограммы на экране дисплея его можно «высветить», подведя к пиктограмме указатель мышки. Используются термины «кликнуть», «клик» и «правый клик», соответственно означающие «щелкнуть левой кнопкой мышки», «щелчок левой кнопкой мышки» и «щелчок правой кнопкой мышки».

При обозначении последовательности операций иногда используются «стрелочка» и «многоточие». Их смысл поясним на следующем примере: запись **Файл** → **Печать** → ... означает, что надо кликнуть на слове **Файл**, затем на пункте **Печать** открывшегося списка и далее выполнить очевидные операции.

В данной главе часто говорится о выделении объекта на экране дисплея. Для выделения следует «наехать» на объект указателем мышки, после чего кликнуть. При этом объект изменит свой цвет.

В приложении 1 сформулировано 5 заданий, выполнение которых необходимо для закрепления материала. В приложении 2 приведена таблица, позволяющая записывать русские тексты латинскими (английскими) буквами. Включение в главу этого приложения объясняется тем, что в настоящее время не все персональные компьютеры за рубежом оснащены файлами, позволяющими работать с русскими текстами. В приложении 3 приведен список полезных сайтов, о которых не говорилось в предыдущих разделах главы.

§ 2. История Интернета

В 1968 г. в Министерстве обороны США возникла необходимость связать множество компьютеров, расположенных далеко друг от друга. Было решено создать специальную сеть. Разработка проекта сети и его осуществление были поручены ARPA

(Advanced Research Projects Agency) — Управлению передовых исследований Министерства обороны. Сеть должна была:

- способствовать научным исследованиям в военно-промышленной сфере;
- быть объектом исследования методов построения сетей, устойчивых к частичным повреждениям, например в результате ракетно-ядерного удара.

Такая сеть появилась через пять лет. Называлась она ARPAnet. Передача данных в сети была организована на основе Межсетевого протокола IP (Internet Protocol) — описания правил и принципов работы сети. Благодаря этому в сети могли использоваться компьютеры различных марок, оснащенные различными операционными системами. Требовалось лишь, чтобы компьютеры подсоединялись к сети согласно IP.

Сеть проектировалась так, чтобы от пользователей не требовалось никаких знаний о ее структуре, которая в каждый момент могла измениться. Чтобы послать сообщение, компьютер (выражаясь фигурально) должен был поместить данные в некий «конверт», указать на «конверте» конкретный адрес и передать получившийся пакет в сеть согласно IP.

В 70-е годы развитие сетей происходило незаметно. Их услугами пользовались лишь специалисты по вычислительной и военной технике.

В начале 80-х годов появились и стали быстро завоевывать популярность локальные сети, т. е. сети, связывающие близко расположенные компьютеры с целью оптимального их использования при решении прикладных задач (типичный пример прикладной задачи — прогноз погоды).

Во второй половине 80-х годов Национальный научный фонд США создал суперкомпьютерные центры в шести университетах пяти штатов для использования их в фундаментальных и прикладных исследованиях. Из-за дороговизны этих центров их следовало эксплуатировать кооперативно. Возникла проблема связи: требовалось соединить эти центры и предоставить доступ к ним различным пользователям.

Идея использования имеющихся военных коммуникаций ARPAnet вызвала возражения чиновников военного ведомства.

Тогда Национальный научный фонд разработал и создал свою собственную сеть. Суперкомпьютерные центры были соединены телефонными линиями с пропускной способностью (скоростью передачи информации) 56 килобит в секунду (kbps). Позже в сети были использованы телефонные линии с пропускной способностью порядка 1 мегабита в секунду. На более быстрые были заменены и сетевые управляющие компьютеры. Появился термин «Интернет».

Напомним, что бит — это минимальный элемент информации. Дело в том, что любую информацию можно записать как последовательность нулей и единиц, например, число 17 можно записать как 10001. Каждый элемент последовательности (а здесь их пять) называется битом. Элементом информации является также байт, равный 8 битам.

Возвращаясь к Интернету, следует подчеркнуть, что он появился благодаря усилиям Национального научного фонда США и других научных организаций всего мира. Причиной быстрого роста Интернета явилась его доступность.

В 1989 г. Тим Бернерс-Ли (сотрудник Европейской лаборатории физики элементарных частиц CERN) представил руководству документ, в котором показал достоинства текстовой системы с перекрестными ссылками (так называемой нелинейной текстовой системы). На основе этой теоретической работы он создал программу с эргономичным пользовательским интерфейсом (т. е. средством общения человека с программой), с помощью которого пользователь в диалоговом режиме мог обращаться к различным базам данных.

Такой интерфейс был использован в Интернете, благодаря чему он приобрел современный вид «Всемирной паутины» (World Wide Web, WWW).

§ 3. Интернет в России

В 1990 г. сотрудниками Института атомной энергии им. И. В. Курчатова была разработана и создана небольшая сеть для общения ученых нескольких московских научных организаций.

Сеть получила название Relcom (от «reliable communications» — «надежная связь»).

Сеть Relcom быстро расширялась, в основном, за счет подключения к ней локальных сетей научных институтов физического и математического профиля. Вскоре было решено подключиться к Интернету, что не явилось проблемой, т. к. сеть Relcom создавалась на основе стандартов Интернета.

Появление Интернета в России имело огромное значение, т. к. для многих ученых он явился единственным каналом связи с зарубежными коллегами.

В настоящее время происходит развитие Интернета в России, как и Интернета вообще.

§ 4. Адреса и имена серверов

Сетевые управляющие компьютеры (серверы, домены) в Интернете имеют числовые адреса (так называемые IP-адреса), которые записываются в виде четырех десятичных чисел (от 0 до 255 каждое), разделенных точками. Пример: 32.22.235.45 — числовой адрес сервера еженедельника «Аргументы и факты».

Числовые адреса серверов аналогичны индексам предприятий связи обычной почты. Они удобны для связи компьютеров, для людей же удобна система адресов, в которой серверам присваиваются имена.

Имя сервера — слово, однозначно обозначающее сервер в Интернете. Оно состоит из нескольких информационных полей, которые отделяются друг от друга точками. Пример: www.aif.ru — имя сервера еженедельника «Аргументы и факты».

Если в последнем информационном поле имени сервера две буквы, то они кодируют страну, в которой расположен сервер:

ru — Россия;

us — США;

ca — Канада;

uk — Великобритания;

de — Германия;

fr — Франция;

it — Италия;

jp — Япония;

cn — Китай;

и т. д.

Если в последнем информационном поле три буквы, то они кодируют тип организации, которой сервер принадлежит:

com — коммерческие учреждения;
org — некоммерческие учреждения;
edu — образовательные учреждения;
net — сетевые организации;
int — международные организации;
mil — военные организации;
gov — правительственные организации.

Один и тот же сервер может иметь несколько имен. Так, например, сервер Эрмитажа в Санкт-Петербурге имеет имена `www.hermitage.ru` и `www.hermitagemuseum.org`.

Имеется соответствие между именами и числовыми адресами серверов. Компьютеры, в которых хранятся таблицы соответствия, называются серверами DNS (Domain Name System).

§ 5. Виды доступа в Интернет

Доступ в Интернет пользователь получает через поставщика услуги — так называемого провайдера. Каждый провайдер, как правило, обеспечивает один способ подключения пользователя к Интернету.

Самый простой способ подключения к Интернету — так называемый модемный доступ dial-up («набираем номер телефона», «дозваниваемся»). Для него кроме компьютера нужно иметь телефонную линию и модем (часто он входит в состав материнской платы компьютера). Соединить их воедино и заставить работать не очень сложно — вызывать настройщика не придется. После подключения не пользователь, а модем станет дозваниваться до провайдера, открывая тем самым доступ в Интернет. При этом телефонная линия будет занята.

Этот вид подсоединения к Интернету подходит тем, кому Интернет нужен для операций с небольшими объемами информации: для проверки электронной почты, чтения новостей, посещения сайтов.

Существует способ подключения к Интернету через телефонную линию, при котором телефон остается свободным. Это — так называемый стандарт DSL (Digital Subscriber Line — цифровая абонентская линия). Такой способ подключения характеризуется высокой скоростью передачи информации.

Подключение по выделенной линии также характеризуется высокой скоростью передачи информации. В этом случае выбор компании, с помощью которой пользователь будет выходить в Интернет, зависит от географии: жители соседних домов объединяются в локальную сеть и подключаются к одному сетевому кабелю. Если дом уже подключен к выделенной линии, пользователю достаточно заключить договор с компанией-провайдером, и настройщик подсоединит пользователя к общей сети. Иметь домашний телефон для подобного подключения необязательно.

Такой способ подключения к Интернету нужен активным пользователям, работающим в Интернете часто и подолгу.

Подключиться к Интернету можно через мобильный телефон, если он поддерживает функцию GPRS (General Packet Radio Services — пакетная радиосвязь общего назначения). В справочной службе оператора мобильной связи объяснят, как настроить телефон. Считывать и отправлять большие объемы информации таким способом разорительно.

§ 6. World Wide Web — «Всемирная паутина»

World Wide Web переводится как «Всемирная паутина» и обозначается WWW. Она основана на понятии гипертекста.

Гипертекст — это текст со вставленными в него словами-командами (гиперссылками). Иллюстрированный текст, в котором командами являются не только слова, но и другие объекты (например, точки или пиктограммы), называется гипермедиа (от слов «гипертекст» и «мультимедиа»). Однако обычно используется термин «гипертекст», даже в случае, когда речь идет о гипермедиа.

На экране дисплея гиперссылки выделяются, например, цветом. Кроме того, на гиперссылке указатель мышки меняет форму. Если кликнуть на гиперссылке, то программа поддержки гипертекста выполнит некое действие, связанное с данным словом, например, на экране появится текст, на который ссылается это слово, фотография или другой гипертекст.

В дальнейшем гиперссылку, как правило, для краткости будем называть ссылкой.

Гипертекстовые документы, связанные между собой ссылками, образуют гипертекстовую систему.

Гипертекстовые документы «Всемирной паутины» называются Web-страницами, а гипертекстовые системы — наборами Web-страниц, сайтами или Web-узлами.

Web-страница является минимальным адресуемым элементом информации «Всемирной паутины». Страницы объединяются в сайты по смыслу. Физически страницы одного сайта могут размещаться как на одном компьютере, так и на разных, расположенных по соседству или на значительном расстоянии друг от друга.

По ссылке можно не только попасть в другой документ, но, например, запустить какую-нибудь программу на компьютере, находящемся в другом полушарии. Результаты выполнения программы будут автоматически переданы на компьютер, где была сделана ссылка.

Наглядным примером гипертекста является географическая карта с выделенными точками. Кликнув на такой выделенной точке, пользователь попадает на соответствующий сайт, который демонстрирует достопримечательности данной местности, даже со звуковым сопровождением.

§ 7. URL и протоколы доступа к ресурсам Интернета

Текстовая строка, с помощью которой компьютеру отдается приказ на использование того или иного ресурса Интернета, называется унифицированным указателем ресурсов URL (Uniform Resource Locator). Структура URL такова: сначала идет имя протокола доступа к ресурсу (т. е. правил его использования), далее

— адрес ресурса (иногда адресом ресурса называют весь URL. Рассмотрим некоторые протоколы доступа к ресурсам.

Протокол HTTP (Hypertext Transfer Protocol — протокол передачи гипертекстов) разработан для установления связи компьютера пользователя с HTTP-сервером (см. ниже) и эффективной загрузки в компьютер пользователя Web-страницы, в общем случае задаваемой несколькими файлами — основным (с расширением .htm или .html) и вспомогательными.

URL содержит следующие элементы:

- 1) http — протокол доступа к ресурсу;
- 2) адрес ресурса:
 - имя HTTP-сервера, содержащего нужную Web-страницу (после ://);
 - путь к файлам страницы в дереве каталогов (после /);
 - имя основного файла (после /).

В адресе ресурса последние составляющие могут отсутствовать. Рассмотрим два примера URL.

1. <http://statserv.relcom.ru/statserv/infoline.html> — означает загрузку в компьютер пользователя Web-страницы, находящейся на HTTP-сервере statserv.relcom.ru, причем:

- страница находится в директории statserv;
- основной файл страницы имеет имя infoline.html.

2. <http://www.netscape.com> — установление связи компьютера пользователя с HTTP-сервером www.netscape.com.

Протокол FTP (File Transfer Protocol — протокол передачи файлов) предназначен в основном для установления связи между компьютерами и пересылки файлов с одного компьютера на другой.

Пример URL: <ftp://ftp.cdrom.com> — означает установление связи компьютера пользователя с FTP-сервером ftp.cdrom.com (об FTP-серверах см. ниже).

Протокол Telnet является протоколом эмуляции терминала. С его помощью пользователь (сидя за персональным компьютером) может подключиться к удаленному компьютеру и производить действия над его файлами точно так же, как если бы работал на этом компьютере, сидя перед его дисплеем и клавиатурой

(терминалом). Но для этого надо иметь соответствующие права, т. е. знать имя (login) и пароль (password).

Чтобы воспользоваться протоколами HTTP и FTP, требуются специальные программы, так называемые HTTP- и FTP-клиенты. Эти программы входят в состав браузера Internet Explorer (см. ниже).

§ 8. HTTP- и FTP-серверы

Существует множество HTTP- и FTP-серверов, на которых собрано большое количество файлов. HTTP-серверы являются хранилищами Web-страниц. На FTP-серверах, например, могут храниться архивы программ крупных компаний или программы, распространяемые бесплатно. Тип сервера определяется протоколом, используемым при загрузке с него информации.

При соединении компьютера пользователя с сервером или попытке загрузить с него файл могут запрашиваться имя пользователя и пароль. В Интернете принято соглашение об универсальном способе входа на сервер, который был назван анонимным. При этом пользователь любому серверу сообщает одно и то же имя — *anonymous*. Паролем обычно служит произвольное слово, которое сервером игнорируется. Естественно, при анонимном способе предоставляется доступ только к файлам, предназначенным для всех. Другие разделы файловой структуры сервера, защищенные настоящими паролями, доступными не будут.

В принципе HTTP- и FTP-серверами могут быть любые компьютеры. Для этого нужно всего лишь установить на них соответствующее программное обеспечение.

§ 9. Браузер Internet Explorer

Браузер (browser), или обозреватель, — это программа на компьютере пользователя, являющаяся основным средством обмена информацией между пользователем и Интернетом. Наиболее распространенным браузером является Internet Explorer. Рас-

смотрим шестую версию этой программы, входящую в состав операционной системы Windows XP.

Для запуска программы Internet Explorer достаточно сделать двойной клик на ее пиктограмме. Internet Explorer можно запустить и любым другим стандартным способом. При запуске появляется Web-страница, называемая домашней (home page).

Наиболее часто используемыми ресурсами Интернета являются Web-страницы. Поэтому, в основном, будем говорить о работе пользователя с Web-страницами.

Рассмотрим обязательные элементы окна Internet Explorer.

1. Заголовок окна на синем фоне.

2. Строка меню **File (Файл)**, **Edit (Правка)**, **View (Вид)**, **Favorites (Избранное)**, **Tools (Сервис)**, **Help (Справка)**. С помощью меню можно сохранять Web-страницы в дисковой памяти компьютера, осуществлять поиск на странице, переходить от страницы к странице, пользоваться папкой избранных ссылок и формировать ее, изменять свойства броузера и т. д.

3. Область показа текущей страницы. Выделенные фрагменты страницы — это ссылки. Ссылку можно также отличить по виду указателя мышки: если он принимает форму руки, значит, перед вами ссылка. Клик на ссылке перемещает вас на страницу, соответствующую этой ссылке. Когда вы вернетесь на исходную страницу, броузер напомнит, какой ссылкой вы воспользовались, изменив ее цвет.

Кроме перечисленных обязательных элементов окна Internet Explorer, имеются также следующие необязательные элементы.

1. Панели инструментов, предоставляющие быстрый доступ к наиболее часто используемым командам. Выводятся и убираются, как в текстовом процессоре Word: **View (Вид)** → **Toolbars (Панели инструментов)** → ...

2. Строка состояния *Status Bar*. Выводится и убирается аналогично панели инструментов: **View (Вид)** → **Status Bar (Строка состояния)**. Она находится в нижней части окна. В строке состояния отображается информация о том, чем занят броузер в данный момент. В частности, наблюдая процесс увеличения длины зеленой планки, можно следить за загрузкой Web-страницы.

Рассмотрим основные панели инструментов.

- *Standard Buttons (Обычные кнопки)* — включает в себя пиктограммы часто используемых команд, например, **Back (Назад)** и **Forward (Вперед)** для быстрого перехода от одной Web-страницы к другой.

- *Address Bar (Адресная строка)* — состоит из текстового поля *Address (Адрес)*, скрытого списка адресов и кнопки **Go (Переход)**. Текстовое поле *Address (Адрес)*, во-первых, показывает адрес (URL) используемого ресурса Интернета (например, Web-страницы), а во-вторых, служит для ввода адреса нужного ресурса. Для того чтобы открыть скрытый список адресов, надо кликнуть на стрелочке ▼ справа от текстового поля *Address (Адрес)*. При клике на нужном адресе (в этом списке) он появляется в текстовом поле. При клике на кнопке **Go (Переход)** или нажатии клавиши «Enter» программа Internet Explorer начинает использовать ресурс, адрес которого указан в текстовом поле *Address (Адрес)*.

Панель инструментов *Standard Buttons (Обычные кнопки)* включает в себя следующие пиктограммы.

1. **Back (Назад)** и **Forward (Вперед)** — для быстрого перехода от одной недавно загруженной Web-страницы к другой.

2. **Stop (Остановить)** — для прекращения загрузки страницы.

3. **Refresh (Обновить)** — для повторной загрузки страницы (если пользователь считает, что страница доставлена с ошибками, и хочет получить ее еще раз).

4. **Home (Домой)** — для возврата на домашнюю страницу.

5. **Search (Поиск)** — для нахождения Web-страницы в Интернете по заданному признаку.

6. **Favorites (Избранное)**. При клике на этой пиктограмме открывается папка избранных ссылок. Отпадает необходимость каждый раз при обращении к «любимой» Web-странице набирать ее адрес, надо лишь найти ее пиктограмму в открытой папке и дважды кликнуть на ней. В папку избранных ссылок можно добавлять страницы и целые директории, можно исключать их.

7. **History (Журнал)** — для возврата к Web-страницам, загруженным ранее.

8. **Mail (Почта)** — для вызова почтовой программы, чтобы отправить текущую Web-страницу по электронной почте.

Перечисленные команды, естественно, можно выполнять и без панели инструментов *Standard Buttons* (*Обычные кнопки*) с помощью строки меню. Рассмотрим некоторые полезные команды, не вынесенные на эту панель инструментов.

1. Запоминание текущей Web-страницы в дисковой памяти компьютера: **File (Файл) → Save As (Сохранить как)**. В текстовом поле *Save as type (Тип файла)* появившегося окна следует установить *Web Page, complete (Web-страница полностью)*. После задания имени файла и директории, где этот файл следует разместить, надо кликнуть на кнопке **Save (Сохранить)**.

Помимо файла с расширением .htm или .html в выбранной директории появится папка (или несколько папок) со вспомогательными файлами. Когда потребуется посмотреть Web-страницу, надо дважды кликнуть на пиктограмме соответствующего файла с расширением .htm или .html в окне программы Windows Explorer (Проводник).

2. Замена домашней Web-страницы (home page): **Tools (Сервис) → Internet Options (Свойства обозревателя) → вкладка General (Общие) → ...**

3. Поиск на текущей Web-странице: **Edit (Правка) → Find on This Page (Найти на этой странице) → ...**

Internet Explorer производит «кэширование» данных — сохранение в дисковой памяти компьютера пользователя (в так называемом кошельке — кэше) загруженных Web-страниц. Если со времени последнего посещения на странице не произошло никаких изменений, то при обращении к этой странице она считывается из дисковой памяти компьютера. Это позволяет не перегружать каналы связи Интернета.

Для соединения с нужным FTP-сервером в текстовом поле *Address (Адрес)* программы Internet Explorer следует набрать его адрес, например ftp://ftp.netscape.com. Если при клике на кнопке **Go (Переход)** или нажатии клавиши «Enter» происходит соединение с сервером, то окно Internet Explorer приобретает вид окна программы Windows Explorer (Проводник). Далее, если позволяет сервер, можно просмотреть файловую структуру на сервере, произвести копирование и удаление файлов и т. д., причем с программой Internet Explorer можно работать, как с программой Windows Explorer (Проводник).

§ 10. Поисковые системы Интернета

Поисковые системы (поисковики, поисковые машины) призваны помочь пользователю найти нужные Web-страницы и сайты (наборы Web-страниц), не утонув в море информации «Всемирной паутины».

Как отдельные Web-страницы, так и сайты будем называть документами.

Специальные программы-роботы поисковиков (известные также как «пауки») периодически обследуют Интернет, проводя индексацию найденных документов, т. е. составляя своеобразные каталоги. Кроме того, каталоги составляются вручную.

Каталоги последнего типа работают на основе добровольной регистрации документа Web-мастером (создателем документа). Они включают в себя следующие сведения:

- 1) название и адрес документа;
- 2) категорию, к которой относится документ, как то: наука и образование, музей, учебное заведение и т. д.;
- 3) аннотацию и ключевые слова.

Пользователь поисковика в рамках имеющегося интерфейса формулирует запрос. После, нажатия клавиши «Enter» запрос обрабатывается системой с помощью имеющихся каталогов и выдаются результаты поиска — имена и адреса документов, удовлетворяющих запросу.

Исследование возможностей поисковиков показывает, что реальная полнота охвата ресурсов «Всемирной паутины» одной системой не превышает 30%. Поэтому, если вам не удалось отыскать интересующую вас информацию с помощью одной системы, попробуйте воспользоваться другой. Каждая поисковая система имеет свои особенности, и качество полученного результата зависит от предмета поиска и формулировки запроса.

Помимо отдельных поисковиков, имеются системы метапоиска (параллельного поиска). Они передают сформулированный пользователем запрос разным поисковикам, что позволяет выполнять поиск сразу в нескольких поисковых системах. Неизбежные повторы документов, как правило, автоматически ис-

ключаются из предоставляемых пользователю результатов поиска.

Очень важно точно сформулировать запрос на поиск. Рассмотрим основные виды запросов пользователя.

- Слово.
- Фраза.
- Несколько слов с операторами включения (+) и/или исключения (–).

Например, запрос *+картошка –курица* означает, что в искомым документах должно содержаться слово «картошка» и не должно содержаться слово «курица».

- Несколько слов, связанных логическими операциями AND, OR, NOT (И, ИЛИ, НЕ).

В каждой поисковой системе имеется раздел *Справка, Помощь* или *Help*, с помощью которого можно узнать, какими видами запросов можно пользоваться. С его изучения и следует начинать использование поисковика.

Логические операции AND, OR и NOT имеют следующий порядок старшинства: NOT — самый старший, AND — средний, OR — самый младший. Скобки при составлении запросов имеют тот же смысл, что и в математике: они меняют порядок выполнения операций.

Посмотрим, как поисковая система поймет такие запросы:

- 1) *японская AND кухня*;
- 2) *(японская OR китайская) AND кухня*;
- 3) *(японская OR китайская) AND кухня AND NOT мебель*.

В первом случае будут найдены документы, в которых речь идет о японской кухне, как в смысле кулинарии, так и мебели, во втором случае — документы, в которых речь идет о японской или китайской кухне, как в смысле кулинарии, так и мебели. В третьем случае будут найдены документы, в которых речь идет о японской или китайской кухне в смысле кулинарии.

Примеры поисковых систем: www.yandex.ru — Яндекс; www.rambler.ru — Рамблер; www.google.com — Гугл. Остановимся на поисковике Яндекс.

Работа с Яндексом начинается с ввода *http://www.yandex.ru* (или *http://www.yandex.ru/*, или *www.yandex.ru*) в текстовое поле

Адрес (Address) браузера Internet Explorer. После клика на кнопке **Go (Переход)** или нажатия клавиши «Enter» в ваш компьютер загрузится домашняя страница Яндекса.

Запрос вводится в текстовое поле *Я ищу*. Для начала поиска следует кликнуть на кнопке **Найти** или нажать клавишу «Enter». Результатом поиска являются имена и адреса документов, удовлетворяющих запросу.

Основные операторы запросов Яндекса

Обозначение	Смысл оператора	Пример запроса
Пробел или &	логическое И (в пределах предложения)	лечебная физкультура
&&	логическое И (в пределах документа)	рецепты && (плавленый сыр)
 	логическое ИЛИ	фото фотография снимок
+	обязательное наличие слова в документе	+быть или +не быть
()	группировка слов	(технология изготовление) (сыра творога)
~	оператор И-НЕ (в пределах предложения)	банки ~ закон
~~ или -	оператор И-НЕ (в пределах документа)	путеводитель по Парижу ~~ (агентство тур)
«»	поиск фразы	«красная шапочка»

Помимо строгого поиска (с помощью операторов, приведенных в таблице, и др.) в Яндексе реализован и простой способ: ему можно задать вопрос так же, как всезнайке-эрудиту. Например, *где раки зимуют* или *как выбрать компьютер*. Независимо от того, в какой форме в запросе вы употребили русское или английское слово, при поиске учитываются все его формы по правилам соответствующего языка.

Ниже (с сокращениями и изменениями) приводятся советы по поиску в Яндексе, сформулированные в разделе *Помощь*, куда вы попадете, кликнув на ссылке **Помощь**. Многие из советов полезны при использовании любой поисковой системы.

Проверяйте орфографию. Если не найден ни один документ, то вы, возможно, допустили орфографическую ошибку в написании слова. Проверьте правильность написания. Если при поиске вы использовали несколько слов, то посмотрите (в результатах поиска), сколько раз каждое из слов встречается в найденных документах. Какое-то из слов не встречается ни разу? Скорее всего, его вы и написали неверно.

Используйте синонимы. Если список найденных документов слишком мал или не содержит полезные документы, попробуйте изменить слово. Например, вместо *рефераты*, возможно, больше подойдет *курсовые работы* или *сочинения*. Попробуйте задать для поиска три-четыре слова-синонима сразу. Для этого перечислите их через вертикальную черту. Тогда будут найдены документы, где встречается хотя бы одно из слов-синонимов. Например, вместо *фотографии* попробуйте *фотографии* | *фото* | *снимки* | *фотоизображения*.

Ищите больше, чем по одному слову. Слово *психология* или *продовольствие* дадут при поиске поодиночке множество бессмысленных ссылок. Добавьте одно или два ключевых слова, связанных с искомой темой. Например, *психология Юнга* или *продажа и покупка продовольствия*. Рекомендуем также сужать область запроса. Если вы интересуетесь автомобилями ВАЗ, то запрос *автомобиль ВАЗ* выдаст более подходящие документы, чем запрос *легковые автомобили*.

Не пишите большими буквами. Начиная слово с большой буквы, вы не найдете слов, написанных с маленькой буквы, если это слово не первое в предложении. Поэтому не пишите обычные слова с большой буквы, даже если с них начинается ваш вопрос Яндекс. Заглавные буквы в запросе рекомендуется использовать только в именах собственных. Например, *телепередача Здоровье*.

Используйте знаки плюс и минус. Чтобы исключить документы, где встречается определенное слово, поставьте перед ним знак минус. И наоборот, чтобы определенное слово обязательно присутствовало в документе, поставьте перед ним плюс. (Обратите внимание на то, чтобы между знаком плюс или минус и словом не было пробела.) Например, если вам нужно описание Парижа, а не предложения многочисленных туристических агентств, имеет смысл сформулировать запрос следующим образом: *путеводитель по Парижу –агентство –тур*.

Попробуйте использовать язык запросов. С помощью специальных знаков вы сможете сделать запрос более точным.

Для изучения синтаксиса языка запросов (что нужно для строгого поиска) следует кликнуть на ссылке **Язык запросов** раздела *Помощь*.

§ 11. E-mail — электронная почта

Пересылка писем — исторически первый и по сей день один из важнейших ресурсов Интернета. Достоинствами электронной почты (e-mail) являются высокая скорость передачи информации и низкая стоимость. К недостаткам следует отнести отсутствие конфиденциальности. Впрочем, этот недостаток устраним: имеются надежные программы шифрования электронных документов.

Устройство почтовой службы достаточно простое. При модемном доступе в Интернет оно выглядит следующим образом.

Пользователь имеет закрепленное за ним пространство дисковой памяти на сервере провайдера — так называемый почтовый ящик. Новой считается почта, пришедшая с момента последнего обращения пользователя к почтовому ящику. Она складывается в почтовый ящик и ждет запроса пользователя на доставку в его компьютер. Причем, если пользователем подготовлена исходящая почта, то при запросе сначала отправляется она, а затем проверяется содержимое почтового ящика и в компьютер пользователя загружаются имеющиеся письма. Письмо пользователя пересылается на сервер, указанный в адресе получателя. В случае ошибочного адреса письмо вернется отправителю с указанием ошибки.

Адрес e-mail (электронный адрес) состоит из названия почтового ящика (например, gav), знака @ («at», «собака») и имени сервера (например, caravan.ru). Таким образом, получается gav@caravan.ru. Чтобы послать кому-либо письмо, необходимо точно указать адрес электронной почты корреспондента. Если он неизвестен, то для его определения следует воспользоваться какой-либо службой поиска электронных адресов из перечисленных: www.whowhere.com; www.switchboard.com; www.four11.com.

Электронное письмо может иметь приложение. Им может быть один или несколько произвольных файлов. Так, например, имеется возможность пересылать фотографии или зашифрованные содержательные документы.

Большой популярностью пользуются телеконференции. Телеконференцией называется общение достаточно широкого круга

людей, объединенных общими интересами, посредством электронной почты. В телеконференциях используются рефлекторы почты.

Рефлектор почты (mail reflector) — специальный почтовый адрес. Электронная почта, посланная туда, автоматически рассылается по списку рассылки. Чтобы быть включенным в список рассылки (т. е. стать участником телеконференции), надо послать на компьютер с нужным рефлектором почты соответствующую просьбу.

Программа, позволяющая использовать электронную почту при работе в Интернете, называется почтовой.

§ 12. Почтовая программа Outlook Express

Наиболее распространенной почтовой программой является Outlook Express. Мы рассмотрим шестую версию этой программы, входящую в состав операционной системы Windows XP.

Программа Outlook Express оперирует со следующими пятью папками компьютера пользователя (local folders):

- *Inbox (Входящие)* — содержит поступившие письма (сообщения);
- *Outbox (Исходящие)* — содержит письма, подготовленные для отправки, но еще не отправленные;
- *Sent Items (Отправленные)* — содержит отправленные письма;
- *Deleted Items (Удаленные)* — содержит удаленные письма (своеобразная мусорная корзина);
- *Drafts (Черновики)* — содержит неоконченные письма.

В папку *Drafts (Черновики)* письмо поступает при закрытии окна, используемого для написания письма. Чтобы к письму вернуться, надо дважды кликнуть на нем в перечне писем папки *Drafts (Черновики)*. Аналогично можно вернуться к удаленному письму, находящемуся в папке *Deleted Items (Удаленные)*.

Для запуска программы Outlook Express достаточно сделать двойной клик на соответствующей пиктограмме. Можно ее запустить каким-либо другим стандартным способом.

Вход в нужную папку (из пяти перечисленных) может производиться разными методами. Наиболее общий метод: **View (Вид)** → **Go to Folder (Перейти в папку)** → клик на названии нужной папки → **ОК**. Если на экране (под панелью инструментов) имеется одна из шести кнопок **Outlook Express**, **Inbox (Входящие)**, **Outbox (Исходящие)**, **Sent Items (Отправленные)**, **Deleted Items (Удаленные)** и **Drafts (Черновики)**, то в нужную папку можно войти, кликнув на этой кнопке, а затем — на названии нужной папки.

Если вы вошли в одну из пяти папок, то в окне Outlook Express увидите следующие элементы.

1. Заголовок окна на синем фоне.
2. Строка меню **File (Файл)**, **Edit (Правка)**, **View (Вид)**, **Tools (Сервис)**, **Message (Сообщение)**, **Help (Справка)**.
3. Панель инструментов, включающая пиктограммы **Create Mail (Создать сообщение)**, **Reply (Ответить отправителю)**, **Reply to All (Ответить всем)**, **Forward (Переслать)**, **Print (Печать)**, **Delete (Удалить)**, **Send and Receive All (Доставить почту)**, **Address Book (Адресная книга)**, **Find (Найти)**. Эта панель инструментов предоставляет быстрый доступ к часто используемым командам.
4. Строка состояния (в нижней части окна). В ней отображается информация о том, чем занята программа Outlook Express в данный момент.
5. Перечень писем, содержащихся в текущей папке.
6. Содержание письма, выделенного в перечне.

Для задания вида окна Outlook Express (в частности, настройки панели инструментов) следует выполнить команды **View (Вид)** → **Layout (Раскладка)** → ...

При клике на пиктограмме **Create Mail (Создать сообщение)** на экране монитора появляется окно *New Message (Создать сообщение)*, предназначенное для создания нового письма. В нем содержатся следующие составные части.

1. Настраиваемая панель инструментов. Настройка: **View (Вид)** → **Toolbars (Панели инструментов)** → ...
2. Текстовое поле *To (Кому)* — для задания адреса корреспондента. Если корреспондентов несколько, то адреса отделяются друг от друга запятой или точкой с запятой.

3. Текстовое поле *Cc* (*Копия*) — для задания адресов, в которые отправляются копии письма (сокращение *Cc* образовано от «*Courtesy copy*»).

4. Текстовое поле *Subject* (*Тема*) — для задания предмета письма, например, *Hi from Moscow*.

5. Область документа. В этом текстовом поле набирается письмо, причем его можно редактировать и проверять, как это делается в текстовом процессоре Word. В текст письма можно вставить целый файл. Для этого надо выполнить команды **Insert** (**Вставка**) → **Text from File** (**Вложение файла**) → ... Можно также набрать письмо в Word, а потом перенести его в область документа средствами буфера обмена. Для этого следует:

- в Word выделить нужный текст, а затем выполнить команды **Главная** → **Копировать**;
- далее в окне *New Message* (*Создать сообщение*) программы Outlook Express выполнить команды **Edit** (**Правка**) → **Paste** (**Вставить**).

Процесс ввода информации в текстовые поля *To* (*Кому*) и *Cc* (*Копия*) можно упростить, если кликнуть на слове «*То*» («*Кому*») или «*Сс*» («*Копия*»). При этом откроется окно *Select Recipients* (*Выбрать получателей*). Далее следует нужное число раз выполнить следующую последовательность операций:

- выделить адрес или адресата в списке (в левой части окна);
- кликнуть на одной из кнопок со стрелочкой → (в центре окна).

По окончании формирования списков адресов (в правой части окна) надо кликнуть на кнопке **ОК**.

Следует иметь в виду, что корреспонденту, адрес которого указан в текстовом поле *To* (*Кому*), будет известно, что вы послали письмо и по адресам, перечисленным в текстовом поле *Cc* (*Копия*).

Иногда удобно использовать текстовое поле *Bcc* (*Скрытая*). Чтобы оно появилось, в окне *New Message* (*Создать сообщение*) надо выполнить **View** (**Вид**) → **All Headers** (**Все заголовки**). Каждый из адресатов, перечисленных в текстовом поле *Bcc* (*Скрытая*), получит копию письма, но остальные об этом не узнают. (Сокращение *Bcc* образовано от «*Blind courtesy copy*».)

Теперь рассмотрим режимы Reply (Ответить отправителю) и Reply to All (Ответить всем).

Вы работаете с папкой *Inbox* (*Входящие*), и надо ответить на письмо, выделенное в перечне. Для этого кликните на пиктограмме **Reply** (**Ответить отправителю**). При этом появится окно, похожее на окно *New Message* (*Создать сообщение*), но со следующими особенностями.

- В текстовом поле *To* (*Кому*) уже будет адрес корреспондента, которому ответ предназначен.
- В текстовом поле *Subject* (*Тема*) будет указан предмет письма, на которое ответ производится, но с приставкой *Re:*, означающей «Reply», т. е. «Ответ».
- В области документа будет текст письма, на которое производится ответ, причем строки текста будут помечены символом >. Текст ответа набирается в той же области документа, в удобных местах. Включение исходного письма в текст ответа помогает следить за нитью дискуссии. Лишние строчки можно удалить средствами редактирования.

Режим Reply to All (Ответить всем) аналогичен режиму Reply (Ответить отправителю), но количество адресов больше: в их число включаются адреса, перечисленные в текстовом поле *Cc* (*Копия*) того письма, на которое производится ответ. Для использования этого режима надо кликнуть на пиктограмме **Reply to All** (**Ответить всем**).

Вы по-прежнему работаете с папкой *Inbox* (*Входящие*), и требуется переслать полученное письмо. Для этого кликните на пиктограмме **Forward** (**Переслать**). При этом на экране появится окно, похожее на окно *New Message* (*Создать сообщение*), но со следующими особенностями:

- в текстовом поле *Subject* (*Тема*) будет указан предмет пересылаемого письма, но с приставкой *Fw:*, означающей «Forward», т. е. «Пересылка»;
- в области документа будет присутствовать текст пересылаемого письма, каждая строчка которого помечена символом >.

В текстовое поле *To* (*Кому*) следует ввести адрес корреспондента.

Благодаря конструкции приложения (attachment) можно пересылать любые файлы. Изображение канцелярской скрепки в перечне писем говорит о наличии у письма приложения. Чтобы его открыть, надо выполнить следующие действия.

1. Дважды кликнуть на имени письма со скрепкой в перечне писем. При этом появится новое окно, в поле *Attach (Вложение)* которого — пиктограммы приложенных файлов.

2. Дважды кликнуть на пиктограммах приложенных файлов.

Следует иметь в виду, что компьютер получателя письма должен быть оснащен программами, способными открывать приложенные файлы. Пусть, например, приложена электронная таблица, созданная с помощью табличного процессора Excel. Тогда компьютер получателя должен быть оснащен программой Excel или другой программой, способной открыть электронную таблицу.

Приложенный файл можно открепить (т. е. поместить в нужную директорию компьютера), не открывая. Для этого на его пиктограмме делается правый клик, а затем — клик на пункте **Save As (Сохранить как)** открывшегося меню ...

Чтобы в окне *New Message (Создать сообщение)* приложить к письму файл, необходимо выполнить следующие операции.

1. **Insert (Вставка)** → **File Attachment (Вложение файла)** или кликнуть на изображении скрепки на панели инструментов.

2. Найти и выделить файл, который надо приложить к письму, и кликнуть на кнопке **Attach (Вложить)**.

Для приложения еще одного файла эти операции следует повторить.

После того как письмо составлено, его можно отправить, кликнув на пиктограмме **Send (Отправить)**. В процессе отправки письма вы также получите почту из почтового ящика, который находится на сервере. Можно также выполнить команды **File (Файл)** → **Send Later (Отправить позже)**. При этом письмо будет только записано в папку *Outbox (Исходящие)*. Отправить его можно позже, например, когда будет дешевле тариф Интернета или когда в папке *Outbox (Исходящие)* накопится несколько писем.

Программа Outlook Express снабжена адресной книгой, призванной обеспечить быстрый доступ к адресам. Чтобы ее от-

крыть, надо выполнить команды **Tools (Сервис) → Address Book (Адресная книга)**. При этом открывается окно *Address Book — Main Identity (Адресная книга — Главная идентификационная)*. Графы *Name (Имя)* и *E-Mail Address (Адрес электронной почты)* этого окна заполняются автоматически при появлении нового корреспондента. Чтобы заполнить графы *Business Phone (Служебный телефон)* и *Home Phone (Домашний телефон)*, а также внести другую полезную информацию (например, дату рождения, состав семьи), надо выполнить следующие операции.

1. Выделить нужный адрес в окне *Address Book — Main Identity (Адресная книга — Главная идентификационная)*.

2. Кликнуть на пиктограмме **Properties (Свойства)** или выполнить команды **File (Файл) → Properties (Свойства)**.

3. Внести имеющуюся информацию в текстовые поля вкладок.

4. Кликнуть на кнопке **ОК**.

Чтобы в адресную книгу ввести новый адрес вручную, надо кликнуть на пиктограмме **New (Создать)** в окне *Address Book — Main Identity (Адресная книга — Главная идентификационная)*. Далее следует кликнуть на пункте **New Contact (Создать контакт)** открывшегося меню, заполнить нужные текстовые поля, а затем последовательно кликнуть на кнопках **Add (Добавить)** и **ОК**.

Чтобы изменить порядок расположения писем в их перечне, надо кликнуть на заголовке одного из шести столбцов перечня (восклицательном знаке, скрепке или др.). Это относится к любой из пяти папок.

Для распечатки писем или их фрагментов (предварительно выделенных с помощью мышки с нажатой левой кнопкой) используется пиктограмма **Print (Печать)**, для удаления писем — пиктограмма **Delete (Удалить)**. Пиктограмма **Find (Найти)** используется для нахождения письма по заданным признакам.

Следует подчеркнуть, что с программой Outlook Express можно работать в режиме offline, т. е. будучи отключенным от Интернета. Режим online следует включить после клика на пиктограмме **Send (Отправить)** или **Send and Receive All (Доставить)**. После обмена информацией с почтовым ящиком, находящемся на сервере провайдера, компьютер пользователя перехо-

дит в режим offline или остается в режиме online в зависимости от настройки Outlook Express.

Из сказанного следует, что программа Outlook Express очень удобна, если вы пользуетесь Интернетом в условиях, когда компьютер пользователя соединяется с Интернетом на короткое время, например, с помощью мобильного телефона.

§ 13. Почтовая система Mail.ru

В настоящее время на многих сайтах можно завести бесплатный почтовый ящик. С его помощью можно отправлять и получать электронные письма. При этом естественно пользоваться почтовой программой сайта, а не рассмотренной выше программой Outlook Express.

Почтовыми программами сайтов можно пользоваться только в режиме online. Поэтому при доступе в Интернет через мобильный телефон использование сайтовских почтовых программ требует больших денежных затрат, чем использование Outlook Express. Однако при этом увеличивается ваша мобильность: при перемещениях из города в город, из страны в страну вы можете менять провайдера, и при этом ваш почтовый ящик всегда будет доступен под теми именем и паролем, которые вы ему назначили.

В качестве примера почтовой программы рассмотрим почтовую систему Mail.ru сайта www.mail.ru. На домашней странице этого сайта, появляющейся при входе на сайт, находятся текстовые поля *Имя* и *Пароль*. Это говорит о том, что системой могут пользоваться лишь те, кто прошел регистрацию.

Процесс регистрации начинается с клика на ссылке **Регистрация в почте**. На появившейся странице вам надо заполнить форму регистрации, представляющую собой текстовые поля с пояснениями. В ходе регистрации вы задаете имя почтового ящика и пароль. Процесс регистрации оканчивается кликом на кнопке **Зарегистрировать почтовый ящик**. Если вы предоставили всю требуемую информацию и выбранное вами имя почтового ящика не занято, то будет выдано сообщение об успешной регистрации.

Чтобы начать работу с почтовой системой, следует кликнуть на ссылке **@Mail.ru**. Далее надо набрать имя почтового ящика и пароль в соответствующих текстовых полях и кликнуть на кнопке **Войти**. Вы попадете в первое личное окно системы Mail.ru, в котором перечислены папки, находящиеся в вашем распоряжении, и имеется ссылка **Папки**.

Система Mail.ru оперирует с папками, которые представлены соответствующими ссылками. В начале работы с системой Mail.ru может быть только одна папка, *Входящие*, для входящих в ваш адрес писем.

Чтобы получить доступ непосредственно к письмам, нужно кликнуть на названии папки. При этом на экране появится ее содержимое в виде таблицы. Рассмотрим элементы таблицы.

1. Заголовки столбцов. Это не просто слова: если вы кликнете на любом из заголовков, то упорядочите письма в папке соответствующим образом.

2. Квадратик для галочки. Поставив галочку перед письмом (кликнув на квадратике), вы тем самым его отметите, т. е. дадите системе понять, что намереваетесь совершить над письмом некое действие (из описанных ниже). Поставив галочку в заголовке столбца, вы отметите сразу все письма.

3. Канцелярская скрепка обозначает письмо, к которому приложены файлы. Для получения доступа к этим файлам следует открыть письмо.

Над таблицей и под ней находится ряд кнопок, причем всегда присутствуют кнопки **ОК** и **Удалить**. Их назначение состоит в следующем.

- Кнопка **ОК** позволяет переместить отмеченные письма в другую папку. Эта папка может быть стандартной или созданной вами (см. ниже). Для этого следует ввести название папки в текстовое поле *Переместить в*, кликнув на стрелочке ▼ и выделив нужный пункт открывшегося списка, и далее кликнуть на кнопке **ОК**.

- Кнопка **Удалить** позволяет удалить отмеченные письма.

Чтобы открыть письмо, нужно кликнуть на адресе или имени отправителя письма. При этом открывается окно с содержимым письма. В окне также находятся следующие ссылки.

- **Ответить.** После клика на этой ссылке вы можете создать новое письмо, которое будет адресовано отправителю письма, открытого в данный момент. В теме письма будет присутствовать *Re:*, что означает «reply», т. е. «ответ».

- **Переслать** — для пересылки письма, открытого в данный момент. В теме письма будет присутствовать *Fw:*, что означает «forward», т. е. «пересылка». В пересылаемое письмо можно ввести новый текст, к нему можно приложить файлы.

- **Перенаправить** — для пересылки письма без каких-либо добавлений.

- **Печать** — позволяет распечатать текст письма.

- **Удалить** — для ликвидации письма, открытого в данный момент.

Если к письму приложены файлы, то они будут перечислены ниже текста письма. Для открепления какого-либо из этих файлов, т. е. пересылки его с сайта в ваш компьютер, следует кликнуть на соответствующей ссылке **Скачать**. Файл можно удалить, кликнув на ссылке **Удалить**.

Чтобы создать и отправить и/или сохранить новое письмо, выполните следующие действия.

1. Кликните на ссылке **Написать письмо**.

2. Заполните текстовые поля *Кому*, *Копия*, *Скрытая* и *Тема* (их назначение такое же, как в программе Outlook Express). Слова «Кому», «Копия» и «Скрытая» являются ссылками, с помощью которых адреса можно ввести в текстовые поля из адресной книги.

3. Введите текст письма в самое большое текстовое поле — область документа.

4. Установите галочку около текста *Сохранить копию письма в папке «Отправленные»*, если вы хотите оставить себе копию отправленного письма. Если папка *Отправленные* не существует, она будет создана. Позже, открыв эту папку, вы сможете отправить письмо повторно или удалить его.

5. Кликните на кнопке **Отправить**, **В черновики** или **Очистить**.

Назначение последних трех кнопок состоит в следующем.

- Кнопка **Отправить** — для отправки письма по указанному адресу.
- Кнопка **В черновики** позволяет сохранить письмо в папке *Черновики*, не отправляя его. Если папка *Черновики* не существует, она будет создана. Отправить письмо можно позже, открыв папку *Черновики*, а затем — письмо.
- Кнопка **Очистить** — для очистки всех текстовых полей.

С письмом вы можете переслать любой файл. Для этого нужно приложить его к письму.

Чтобы приложить файл к письму, выполните следующие действия.

1. Укажите нужный файл. Для этого воспользуйтесь кнопкой **Browse (Обзор)** или введите полное имя файла (т. е. имя файла с путем и расширением) вручную в текстовое поле, находящееся слева от кнопки **Browse (Обзор)**.

2. Кликните на кнопке **Прикрепить**. Файл будет скопирован в ваш почтовый ящик и приложен к письму.

По окончании процесса прикрепления файла появятся его номер, имя и размер. Кнопка **Убрать** позволяет удалить приложенный файл.

Повторите перечисленные выше действия для всех файлов, которые вы хотите приложить к письму.

По окончании процесса формирования письма не забудьте кликнуть на кнопке **Отправить** или **В черновики**.

Чтобы посмотреть, какие папки находятся в вашем распоряжении, следует кликнуть на ссылке **Папки**, находящейся в первом личном окне системы Mail.ru. В появившейся таблице напротив названия каждой папки находятся ссылки, предназначенные для выполнения соответствующих действий.

- Чтобы удалить папку, кликните на соответствующей ссылке **Удалить**.

- Чтобы очистить папку, кликните на соответствующей ссылке **Очистить**.

- Чтобы создать папку, кликните на ссылке **Создать новую папку**. Далее в текстовое поле *Новая папка* введите название папки и кликните на кнопке **Создать**. Вновь созданная папка появится в списке среди остальных.

Адресная книга нужна для того, чтобы при создании письма можно было быстро заполнить текстовые поля *Кому*, *Копия* и *Скрытая*. Для входа в адресную книгу надо кликнуть на ссылке **Адреса**. Вы увидите окно, в котором можете пополнять список адресов новыми, редактировать информацию об уже имеющихся адресах или удалять их из списка.

- Чтобы добавить новый адрес, кликните на ссылке **Контакт**, находящейся в разделе *Добавить*. Вам будет предложено ввести электронный адрес и, по желанию, многое другое. По окончании добавления адреса следует кликнуть на кнопке **Добавить контакт**.

- Чтобы внести изменения, следует поставить галочку в соответствующей строке адресной книги и кликнуть на ссылке **Редактировать**. При этом откроется окно, аналогичное тому, в котором вы добавляли адрес. По окончании внесения изменений кликните на кнопке **Сохранить изменения**.

- Чтобы удалить какой-либо адрес из адресной книги, следует поставить галочку в соответствующей строке адресной книги и кликнуть на ссылке **Удалить**.

Если вы находитесь в адресной книге, то создание нового письма можно начать, поставив галочку напротив выбранного адреса. Тем самым вы указываете, куда направляется письмо. После клика на кнопке **Написать** можно приступить к созданию письма.

Кликнув на ссылке **Настройки**, находящейся справа от раздела *Добавить* адресной книги, вы получите доступ к параметрам почтовой системы и при необходимости сможете внести изменения в ее настройку.

§ 14. Программы архивации и разархивации

Программы архивации позволяют сжимать информацию в дисковой памяти компьютера, т. е. создавать копии файлов меньшего размера. Эти программы позволяют также упаковывать целые папки (директории), создавая при этом один архивный

файл. Программы разархивации производят разархивацию файлов, т. е. операцию, обратную архивации.

Применение программ архивации и разархивации позволяет сократить время и стоимость передачи информации средствами Интернета.

Иногда функции архивации и разархивации совмещаются в одной программе. Такими программами архивации/разархивации являются WinZip и WinRar.

Чтобы воспользоваться программой WinZip для упаковки файла или папки, надо указатель мышки подвести к пиктограмме этого файла или папки в окне программы Windows Explorer (Проводник). Далее следует открыть контекстное меню, сделав правый клик. В контекстном меню надо кликнуть на пункте **Add to Zip file (Добавить в Zip)**. Далее следует в текстовое поле *Add to archive (Добавить в архив)* ввести полное имя создаваемого архивного файла с расширением .zip (или согласиться с предложенным именем). После клика на кнопке **Add (Добавить)** получится архивный файл.

Архивный файл можно переслать по электронной почте как приложение. Напомним, что приложением может быть произвольный файл, в частности с расширением .zip.

Допустим, что вы по электронной почте получили файл с расширением .zip в качестве приложения и открепили его. Для разархивации этого файла надо указатель мышки подвести к его пиктограмме в окне программы Windows Explorer (Проводник) и открыть контекстное меню, сделав правый клик. В контекстном меню надо кликнуть на пункте **Extract to (Извлечь в)**. Далее следует в текстовое поле *Extract to (Извлечь в)* ввести путь к результирующему файлу или папке (или согласиться с предложенным путем). После клика на кнопке **Extract (Извлечь)** получится файл или папка — результат разархивации.

Архивацию и разархивацию можно производить с помощью программы WinRar. Используется она почти так же, как программа WinZip.

§ 15. Текстовый процессор Word и Интернет

Вернемся к «Всемирной паутине». World Wide Web можно рассматривать как систему документов, связанных между собой ссылками, причем эти документы могут находиться в одном компьютере или в разных компьютерах в разных частях мира. Ссылки позволяют переходить от одного документа к другому. Документы могут включать тексты, рисунки, фотографии, аудио- и видеофайлы.

Файлы Word обычно имеют расширение .doc, но могут иметь расширение .htm или .html, т. е. Word может работать с Web-страницами.

Если вы сомневаетесь в расширении файла, то его можно проверить следующим образом:

- в окне программы Windows Explorer (Проводник) на файле сделать правый клик;
- в открывшемся контекстном меню выполнить команды **Open With (Открыть с помощью) → Choose Program (Выбрать программу)**;
- в верхней части открывшегося окна *Open With (Выбор программы)* будет указано имя файла с расширением.

Если в Word открыта Web-страница, то при ее сохранении расширение не меняется. Для преобразования документа с расширением .doc в Web-страницу в Word выполняется следующая последовательность операций:

- **Office → Сохранить как**;
- указать диск, папку и новое имя файла (или согласиться с предложенным);
- в текстовое поле *Тип файла* ввести *Веб-страница*;
- кликнуть на кнопке **Сохранить**.

Полученный документ можно просматривать браузером Internet Explorer.

Формат ссылки (гиперссылки) включает адрес объекта ссылки и текст ссылки, которые задаются при ее создании.

Для создания ссылки в документе с именем, например, index и расширением .htm или .html, открытом в Word, надо выполнить следующую последовательность операций:

- мигающий курсор установить на том месте документа, где должна находиться ссылка;

- выполнить команды **Вставка** → **Связи** → **Гиперссылка**;
- в открывшемся окне *Вставка гиперссылки* в текстовое поле *Адрес* ввести адрес объекта ссылки, например, <http://www.microsoft.com/ms.htm> (адрес Web-страницы Интернета) или [test.htm](#) (имя файла, находящегося в той же папке, что и документ *index*); при этом в текстовом поле *Текст* появится предлагаемый текст ссылки, который можно изменить;
- кликнуть на кнопке **ОК**.

В документе *index* на месте мигающего курсора появится выделенный цветом текст ссылки, ранее присутствовавший в текстовом поле *Текст*.

После сохранения откроем этот документ браузером Internet Explorer. При клике на тексте ссылки в окне Internet Explorer будет открыт объект, на который указывает ссылка.

В документе *test.htm*, ссылка на который была вставлена в документ *index*, могут присутствовать ссылки на другие документы. Таким образом, можно построить гипертекстовую систему или набор Web-страниц, т. е. сайт.

§ 16. Размещение набора Web-страниц на сайте Narod.ru

После того, как набор Web-страниц принял задуманный вид, его можно разместить на том или ином сайте. Соответствующие инструкции обычно можно найти в разделе помощи сайта.

Рассмотрим метод загрузки набора Web-страниц на сайт Narod.ru, имеющий адрес www.narod.yandex.ru. Сразу отметим, что имена файлов набора могут содержать лишь латинские буквы, цифры и символ подчеркивания, причем имя основного файла должно быть *index.html* или *index.htm*.

Предварительно вы должны зарегистрироваться. Для этого на домашней странице Narod.ru следует задать имя своего сайта и кликнуть на кнопке **Занять**. Если имя окажется занятым, следует задать другое. Далее надо заполнить предложенную анкету. В процессе регистрации вы должны задать пароль. Имя пользователя (логин) совпадает с именем вашего сайта.

После успешной регистрации следует кликнуть на кнопке **Сохранить**. После этого вы попадаете в Мастерскую.

В разделе *Редактирование и управление* Мастерской есть ссылка **Загрузка файлов на сайт**. Кликнув на ней, вы попадаете в окно с 10 кнопками **Browse (Обзор)**. Кликните на первой кнопке **Browse (Обзор)** и в появившемся окне найдите первый из файлов набора страниц, имеющих расширение .htm или .html. После двойного клика на этом файле его имя появится в текстовом поле слева от первой кнопки **Browse (Обзор)**. С помощью второй кнопки **Browse (Обзор)** введите имя второго файла в соответствующее текстовое поле и т. д. Когда имена всех файлов будут введены, кликните на кнопке **Загрузить файлы**. После этого появится список файлов.

На этом процесс загрузки набора Web-страниц на сайт Narod.ru оканчивается. Ваш личный сайт становится доступным всем пользователям Интернета.

При работе с личным сайтом может потребоваться подключение вашего компьютера к Narod.ru по FTP. Для этого в текстовое поле *Internet Explorer Address (Адрес)* нужно ввести следующее: ftp://ваш_логин:ваш_пароль@ftp.narod.ru. Например, если ваш логин — myname, а пароль — 12345, то текстовая строка должна выглядеть так: ftp://myname:12345@ftp.narod.ru. Когда соединение будет установлено, в окне браузера, принявшем вид окна программы Windows Explorer (Проводник), появятся файлы вашего сайта.

§ 17. Создание сайтов средствами Мастерской Narod.ru

Сайты можно создавать средствами сайта **Narod.ru** с адресом www.narod.yandex.ru. Для этого, прежде всего, надо зарегистрироваться (см. предыдущий раздел). После клика на кнопке **Сохранить** вы попадаете в Мастерскую сайта **Narod.ru**.

Кликните на одной из ссылок раздела *Создание сайта*, например на ссылке **Резюме**. Вы увидите предлагаемые шаблоны. Выбранный шаблон будет определять общий облик будущей Web-страницы.

Перед выбранным шаблоном следует поставить точку. После клика на кнопке **Дальше** вы попадаете в окно, где и создается страница.

По окончании создания Web-страницы следует кликнуть на кнопке **Готово**. После этого вы увидите созданную вами страницу в окне с кнопками **Еще редактировать** и **Готово**. Если вы недовольны видом страницы, вам следует кликнуть на кнопке **Еще редактировать**, в противном случае — на кнопке **Готово**. После клика на кнопке **Готово** Web-страница становится доступной для всеобщего обозрения.

Если вы хотите вернуться к своей странице для внесения изменений, вам следует в текстовых полях *Логин* и *Пароль* домашней страницы Narod.ru набрать зарегистрированное имя и пароль и кликнуть на кнопке **Войти**. В разделе *Редактирование и управление* Мастерской, куда вы попадаете, надо кликнуть на ссылке **Управление файлами и HTML-редактор**. В появившемся окне следует кликнуть на кнопке **Редактировать** (с карандашом) слева от имени файла, который вы собираетесь редактировать. После этого можно приступить к внесению изменений, которые оканчиваются кликом на кнопке **Готово**.

Созданную Web-страницу можно назвать одностраничным сайтом. Если вы хотите создать многостраничный сайт, то надо кликнуть на ссылке **Главная страница** (а не **Резюме**) раздела *Создание сайта* Мастерской. Далее следует выбрать шаблон и кликнуть на кнопке **Дальше**.

В появившемся окне надо заполнить текстовые поля, например, следующим образом:

- в текстовое поле *Текст* ввести *Мне нравится преподавание информатики*;
- в текстовое поле *или введите* первой ссылки ввести текст *http://www.guz.ru*, а в текстовое поле *Текст ссылки* ввести *ГУЗ*;
- в текстовое поле *или введите* второй ссылки ввести текст *http://www.gzgarber.narod.ru*, а в текстовое поле *Текст ссылки* ввести *Г.З.Гарбер*.

По окончании создания многостраничного сайта следует кликнуть на кнопке **Готово**.

Подчеркнем следующие три момента:

- в текстовые поля *или введите* можно вводить адреса не только чужих сайтов, но и собственных;
- мы рассмотрели лишь один из методов создания многостраничного сайта средствами Мастерской;
- создавая и редактируя сайт средствами Мастерской, вы работаете в режиме online, поэтому денежные затраты могут быть значительными.

Как только вы создали сайт, он становится доступен программам-роботам поисковых систем. Если по какой-либо причине вы не хотите, чтобы ваш сайт могли найти поисковики, нажмите кнопку **Никогда не индексировать** на странице *Индексирование в поисковых системах*. Эта страница появляется при клике на ссылке **Индексирование в поисковых системах** в разделе *Редактирование и управление* Мастерской.

Для изучения статистики посещений вашего сайта пользователями Интернета следует войти на соответствующую страницу, кликнув на ссылке **Статистика посещений** раздела *Редактирование и управление* Мастерской.

§ 18. Чат и чат-серверы

Слово «чат» в русском языке произошло от английского «chat», которое переводится как «беседа». Беседа в Интернете ведется при помощи обычных персональных компьютеров: каждый участник чата вводит свое сообщение в специальное текстовое поле и отправляет его; это сообщение видят другие участники чата и могут на него ответить. Таким образом, чат является аналогом телеконференции.

Чат-сервер представляет собой компьютер, на котором установлено специальное программное обеспечение, рассылающее группе пользователей Интернета текстовое сообщение, полученное от одного из членов этой группы. В установке специального программного обеспечения на компьютерах пользователей нет необходимости, т. к. все нужное для чата содержится на специальных Web-страницах, которые могут загружаться браузером. Чат-сервер может обслуживать несколько таких страниц, обычно

называемых «комнатами» или «залами». Как правило, каждая из таких комнат объединяет людей с общими интересами.

Большинство чат-серверов требует предварительной регистрации участников чата, которая сводится к заполнению формы. Главная информация — это псевдоним (nickname). Он отображается перед сообщением, что позволяет другим участникам чата идентифицировать автора сообщения. При регистрации может потребоваться задание имени (login) и пароля (password). В дальнейшем они вводятся при входе в чат.

После регистрации вы можете приступить к чату. Для передачи сообщения нужно его ввести с клавиатуры персонального компьютера в предназначенное для этого текстовое поле и кликнуть на соответствующей кнопке или нажать клавишу «Enter». При этом на экране монитора появится введенный текст. Остальные участники чата также увидят это сообщение и могут ответить на него. По окончании чата требуется из него выйти.

Примером чат-сервера является «Отель у Максима». Для установления связи компьютера пользователя с этим сервером следует использовать URL вида <http://chat.cnt.ru>.

§ 19. Интернет-магазины и интернет-услуги

Интернет-магазин — это способ приобретения товаров через Интернет. Один из возможных способов оплаты покупки состоит в следующем.

В особом сетевом банке за обычные деньги (скажем, \$ 200 + 2%, где второе слагаемое — плата банку за услуги) вы покупаете некий сертификат, который будет представлять \$ 200. Точнее, вы получаете не сам сертификат, а только его серийный номер. Тот, кто знает номер, владеет сертификатом. Далее, если вы хотите потратить \$ 200 в интернет-магазине, то посылаете ваш сертификат продавцу по электронной почте. При его получении продавец проверяет сертификат и доставляет вам товар.

Серьезная проблема состоит в сохранении секретности номера сертификата при пересылках по Интернету. Для обеспечения секретности придуманы сложные системы шифрования данных.

Часто «товар» может быть доставлен по Интернету. Речь идет об услугах переводчиков, программистов, врачей и т. д.

Одной из интернет-услуг является так называемое дистанционное обучение, при котором учитель и ученики находятся в разных помещениях, возможно, на большом расстоянии друг от друга. Общение (платное для учеников и оплачиваемое для учителя) происходит в реальном масштабе времени и напоминает чат, причем оно может быть дополнено аудиовизуальным контактом учителя с учениками.

Имеются интернет-магазины и в России. Оплата, как правило, производится наличными деньгами при получении товара.

Примером интернет-магазина является книжный магазин Ozon.ru, имеющий адрес www.ozon.ru.

§ 20. Безопасность работы в Интернете

При работе в Интернете следует иметь в виду следующее: насколько ресурсы этой сети открыты законопослушному пользователю, настолько же эти ресурсы, при определенных условиях, открыты всем, кто обладает необходимыми средствами.

Для пользователя этот факт не играет особой роли, но знать о нем необходимо, чтобы не допускать действий, нарушающих законодательство тех стран, на территории которых расположены серверы Интернета. К таким действиям относятся вольные или невольные попытки нарушить работоспособность компьютерных систем, попытки взлома защищенных систем, использование и распространение программ (в частности, компьютерных вирусов), нарушающих работоспособность компьютерных систем.

Работая в Интернете, следует помнить о том, что все действия, как законные, так и незаконные, протоколируются специальными программами и информация о них обязательно где-то сохраняется. Таким образом, к обмену информацией в Интернете следует подходить, как к обычной переписке с использованием почтовых открыток. Информация свободно циркулирует в обе стороны, но в общем случае она доступна всем пользователям Интернета. Поэтому существует потребность защиты информации в Интернете.

Сегодня Интернет является не только средством общения и универсальной справочной системой — в нем циркулируют договорные и финансовые обязательства, необходимость защиты которых как от просмотра, так и от фальсификации очевидна. Начиная с 1999 года Интернет становится средством обеспечения розничного торгового оборота, а это требует защиты данных кредитных карт и других электронных платежных средств. В Интернете защитой является шифрование информации.

Суть шифрования состоит в том, что к документу применяется некий метод (назовем его ключом), после чего документ становится недоступен для чтения обычными средствами. Его может прочитать только тот, кто знает ключ. Аналогично происходит шифрование ответного сообщения. Если в процессе обмена информацией для шифрования и чтения пользуются одним и тем же ключом, то такой криптографический процесс называется симметричным.

Основной недостаток симметричного процесса заключается в том, что, прежде чем начать обмен информацией, надо выполнить передачу ключа, а для этого опять-таки нужна защищенная связь, т. е. проблема повторяется, хотя и на другом уровне. Если рассмотреть оплату клиентом товара или услуги с помощью кредитной карты, то получается, что торговая фирма должна создать по одному ключу для каждого своего клиента и каким-то образом передать им эти ключи. Это крайне неудобно. Поэтому в настоящее время в Интернете используются несимметричные криптографические системы, основанные на использовании не одного, а двух ключей. Происходит это следующим образом. Компания для работы с клиентом создает два ключа: один — открытый (public — публичный) ключ, а другой — закрытый (private — личный) ключ. На самом деле это как бы две половинки одного ключа, связанные друг с другом.

Ключи устроены так, что сообщение, зашифрованное одной половинкой, можно расшифровать только другой половинкой (не той, которой оно было закодировано). Создав пару ключей, торговая компания широко распространяет публичный ключ (открытую половинку) и надежно сохраняет закрытый ключ (свою половинку).

Как публичный, так и закрытый ключи представляют собой некую кодовую последовательность. Публичный ключ компании может быть опубликован на ее сайте, откуда каждый желающий может его скачать. Если клиент хочет сделать фирме заказ, он возьмет ее публичный ключ и с его помощью закодирует свое сообщение о заказе и данные своей кредитной карты. После кодирования это сообщение может прочесть только владелец закрытого ключа. Никто из участников цепочки, по которой пересылается информация, не в состоянии это сделать. Даже сам отправитель не может прочитать собственное сообщение, хотя ему хорошо известно содержание.

Если фирма хочет отправить клиенту квитанцию о том, что заказ принят к исполнению, она закодирует ее своим закрытым ключом. Клиент сможет прочитать квитанцию, воспользовавшись имеющимся у него публичным ключом данной фирмы. Он может быть уверен, что квитанцию ему отправила именно эта фирма, а не кто-то иной, поскольку никто не имеет доступа к закрытому ключу фирмы.

Защита публичным ключом (впрочем, как и большинство других видов защиты информации) не является абсолютно надежной. Дело в том, что поскольку каждый желающий может получить и использовать чей-то публичный ключ, то он может сколь угодно подробно изучить алгоритм работы механизма шифрования и попытаться установить метод расшифровки сообщения, т. е. реконструировать закрытый ключ.

Это настолько справедливо, что алгоритмы кодирования публичным ключом даже нет смысла скрывать. Обычно к ним есть доступ, а часто они просто широко публикуются. Тонкость заключается в том, что знание алгоритма еще не означает возможность провести реконструкцию закрытого ключа в приемлемые сроки.

Защиту информации принято считать достаточной, если затраты на ее преодоление превышают ожидаемую ценность самой информации. В этом состоит принцип достаточности защиты, которым руководствуются при использовании несимметричных средств шифрования данных. Он предполагает, что защита не абсолютна и приемы ее снятия известны, но она достаточна для того, чтобы сделать это мероприятие нецелесообразным. При появ-

лении иных средств, позволяющих получить зашифрованную информацию в разумные сроки, изменяют принцип работы алгоритма, и проблема повторяется на более высоком уровне.

Разумеется, не всегда реконструкцию закрытого ключа производят методами простого перебора комбинаций. Для этого существуют специальные методы реконструкции закрытого ключа по опубликованному открытому ключу. Область науки, посвященная этим исследованиям, называется криптоанализом, а средняя продолжительность времени, необходимого для реконструкции закрытого ключа, называется криптостойкостью алгоритма шифрования.

Для многих методов несимметричного шифрования криптостойкость, определенная в результате криптоанализа, существенно отличается от величин, заявляемых разработчиками алгоритмов на основании теоретических оценок. Поэтому во многих странах вопрос применения алгоритмов шифрования данных находится в поле законодательного регулирования. В частности, в России к использованию в государственных и коммерческих организациях разрешены только те программные средства шифрования данных, которые прошли государственную сертификацию в административных органах, в частности в Федеральном агентстве правительственной связи и информации (ФАПСИ).

Мы рассмотрели вопрос, как клиент может переслать организации свои конфиденциальные данные, например номер электронного счета. Точно так же он может общаться и с банком, отдавая ему распоряжения о перечислении своих средств на счета других лиц и организаций. Однако здесь возникает следующая проблема: как банк узнает, что распоряжение поступило от данного лица, а не от злоумышленника, выдающего себя за него? Эта проблема решается с помощью так называемой электронной подписи.

Принцип создания электронной подписи тот же, что и рассмотренный выше. Если надо создать электронную подпись, следует с помощью специальной программы, полученной из банка, создать те же два ключа — закрытый и публичный. Публичный ключ передается банку. Если теперь надо отправить поручение банку на операцию с расчетным счетом, оно кодируется публичным ключом банка, а подпись под ним кодируется закрытым

ключом поручителя. Банк поступает наоборот: он читает поручение с помощью своего закрытого ключа, а подпись — с помощью публичного ключа поручителя. Если подпись читаема, это означает, что поручение отправлено данным лицом.

Приложение 1. Темы рекомендуемых лабораторных работ

1. Виртуальное посещение музеев (2 часа)

Задание: средствами программы Internet Explorer войти на сайты Эрмитажа в Санкт-Петербурге (www.hermitage.ru) и Государственного музея изобразительных искусств им. А. С. Пушкина в Москве (www.museum.ru/gmii) и с помощью гиперссылок начать знакомство с экспозициями этих музеев.

2. Поиск необходимой информации (4 часа)

Задание: с помощью рассмотренных поисковых систем найти Web-страницу с описанием строительного объекта, введенного в строй за последние 10 лет в заданном городе.

Найденную Web-страницу, которая потребуется в дальнейшем, сохранить полностью и показать ее преподавателю.

3. Организация почтового ящика, отправка и получение электронных писем (4 часа)

Задание:

1) завести индивидуальный почтовый ящик на сайте www.mail.ru;

2) обменяться письмами (с приложениями и без) с товарищами по группе; копии выслать преподавателю по заданному адресу;

3) выслать преподавателю Web-страницу с описанием строительного объекта, найденную в предыдущей лабораторной работе, в заархивированном виде.

Требование: в текстовом поле *Тема* электронных писем должны присутствовать фамилия и имя автора, а также номера группы и лабораторной работы.

4. Создание набора Web-страниц и размещение его на сайте (4 часа)

Задание:

1) по заданной теме создать многостраничный сайт из собственных и чужих Web-страниц, расставляя гиперссылки в документах Word;

2) созданный сайт разместить на сайте www.narod.yandex.ru;

3) адрес сайта выслать преподавателю.

Требование: в текстовом поле *Тема* электронного письма должны присутствовать фамилия и имя автора, а также номера группы и лабораторной работы.

5. Создание Web-страниц средствами сайта (4 часа)

Задание:

1) по заданной теме создать Web-страницу или набор Web-страниц на сайте www.narod.yandex.ru средствами Мастерской;

2) адрес созданного сайта выслать преподавателю.

Требование: в текстовом поле *Тема* электронного письма должны присутствовать фамилия и имя автора, а также номера группы и лабораторной работы.

Приложение 2. Кодировка русских букв латинскими

Русские буквы	Латинские буквы	Пример
а, б, в, г, д	a, b, v, g, d	
е	ye или e	Ерофеев — <i>Yerofeyev</i>
ё	yo	Ёлкин — <i>Yolkin</i>
ж	zh	Жаров — <i>Zharov</i>
з, и	z, i	
й	y	Буковский — <i>Bukovskiy</i>
к, л, м, н, о, п, р, с, т, у, ф	k, l, m, n, o, p, r, s, t, u, f	
х	kh или h	Хренов — <i>Khrenov</i>
ц	ts	Ципкин — <i>Tsipkin</i>
ч	ch	Чубайс — <i>Chubays</i>
ш	sh	Тимошин — <i>Timoshin</i>
щ	sch	Ващенко — <i>Vaschenko</i>
ы	y	Быков — <i>Bykov</i>
э	e	Этуш — <i>Etush</i>
ю	yu	Юрский — <i>Yurskiy</i>
я	ya	Ошмян — <i>Oshmyan</i>

Замечание: слова произносятся по правилам английского языка.

Приложение 3. Список полезных сайтов

Адрес	Название
www.elibrary.ru	Научная электронная библиотека
www.translate.ru	Online переводчик
www.skype.com	Сайт аудиовизуального и текстового общения
www.odnoklassniki.ru	Сайт общения Одноклассники.ru

Глава 6 Программирование в среде Delphi

§1. Разработка линейных программ

В этом параграфе рассматриваются следующие вопросы: алгоритм и программа; этапы разработки программы; типы данных языка Object Pascal, переменные и константы, оператор присваивания; некоторые процедуры преобразования, простейшие приемы создания окна в среде Delphi, структура процедуры обработки события, создание и сохранение простейшего проекта, компиляция, запуск на выполнение.

Теория

Алгоритм и программа

Алгоритм — это точная последовательность действий, преобразующих исходные данные в нужный результат. Алгоритм должен обладать тремя свойствами: однозначностью, массовостью и результативностью. Однозначность — это единственность толкования правил и порядка выполнения действий. Массовость — это возможность применения алгоритма для решения класса задач, что предполагает его правильную работу при меняющихся в заданных пределах значениях исходных данных.

Алгоритм решения задачи может быть представлен в виде словесного описания или графически в виде блок-схемы. В блок-схемах используются следующие специальные символы:



начало/конец ввод/вывод преобразование проверка
данных (обработка) условия

Программа — это последовательность команд, описывающих алгоритм на языке программирования.

Этапы разработки программы

1. *Определение требований к программе.* На этом этапе подробно описывается исходная информация и формулируются требования к результату, описывается поведение программы в особых случаях. Разрабатываются диалоговые окна, обеспечивающие взаимодействие пользователя и программы.

2. *Разработка алгоритма.* Определяется последовательность действий, которые надо выполнить для получения результата. Результатом этапа разработки алгоритма является подробное словесное описание алгоритма или его блок-схема.

3. *Кодирование.* Алгоритм записывается на выбранном языке программирования .

4. *Отладка* — это процесс поиска и устранения ошибок. Ошибки в программе разделяются на две группы: синтаксические (ошибки в тексте) и алгоритмические (логические).

5. *Тестирование.* На этом этапе проверяют, как ведет себя программа на как можно большем количестве входных наборов данных, в том числе и на заведомо неверных.

Типы данных языка Delphi

Мы рассмотрим здесь те типы данных, которые необходимы для обучения программированию на начальном уровне.

Замечание. Так как данная глава рассчитана на специалистов, которые в своей работе не будут непосредственно заниматься программированием, то мы рассмотрим небольшую часть типов данных Delphi версии 6.

В *Delphi* простые типы данных разбиты на две группы: порядковые, представляющие данные разных объемов, которыми процессор может легко манипулировать, и действительные, пред-

ставляющие приближенно математические действительные числа. В **порядковых типах** информация представляется в виде отдельных элементов. Связь между отдельными элементами и их представлением в памяти определяет естественные отношения порядка между этими элементами. В *Delphi* определены три группы порядковых типов и два типа, определяемых пользователем. Группы — это целые, символьные и булевы типы. Порядковые типы, задаваемые пользователем, — это перечисления и поддиапазоны.

Все значения любого порядкового типа образуют упорядоченную последовательность, и значение переменной порядкового типа определяется его местом в этой последовательности. За исключением переменных целых типов, значения которых могут быть как положительными, так и отрицательными, первый элемент любого порядкового типа имеет номер 0, второй элемент — номер 1 и т.д.

Целые типы

В переменных целых типов информация представляется в виде целых чисел, т.е. чисел, не имеющих дробной части.

Таблица 1.1. Целые типы

Тип	Диапазон значений	Физический формат
Int64	$-2^{63}..2^{63}-1$	64 бит, со знаком
Byte	0..255	8 бит, без знака
Word	0..65 535	16 бит, без знака
Integer	-32 768..32 767	16 бит, со знаком

Символьный тип

Delphi поддерживает универсальный символьный тип — char. В Delphi char соответствует типу данных AnsiChar.

Булевый тип

В Delphi информация о чем-либо, что можно представить как ИСТИНА (True) или ЛОЖЬ (False), хранится в переменных булевых типов. Всего таких типов четыре, но для Delphi предпочтителен тип Boolean, остальные определены для совместимости с другими языками программирования и операционными системами.

Переменным типа Boolean можно присваивать только значения True (истина) и False (ложь).

Перечислимые типы

Перечислимый тип можно объявить следующим образом:

```
Type enum type = (first value, value2, value3, last value); .
```

Обычно данные перечислимых типов содержат дискретные значения, представляемые не числами, а именами. Тип `Boolean` — простейший перечислимый тип в `Object Pascal`. Булевы переменные могут принимать два значения, выражаемые именами `True` и `False`, а сам тип определен в `Object Pascal` так, как будто он объявлен следующим образом:

```
Type Boolean = (False, True); .
```

С помощью типа `Boolean` в `Object Pascal` выполняются сравнения, большинство же перечислимых типов — это просто списки уникальных имен или идентификаторов, зарезервированных с конкретной целью. Например, можно создать тип `MyColor` (мой цвет) со значениями `myRed`, `myGreen` и `myBlue` (мой красный, мой зеленый, мой синий). Это делается совсем просто:

```
Type MyColor = (myRed, myGreen, myBlue); .
```

В этой строке объявлены четыре новых идентификатора: `MyColor`, `myRed`, `myGreen` и `myBlue`. Идентификатором `MyColor` обозначен порядковый тип, следовательно, в синтаксисе `Object Pascal` можно применять этот идентификатор везде, где разрешены перечислимые типы. Остальные три идентификатора — это значения типа `MyColor`. Подобно символьным и булевым типам перечислимые типы не являются числами, и использовать их наподобие чисел не имеет смысла. Однако перечислимые типы относятся к порядковым, так что значения любого такого типа упорядочены. Идентификаторам в списке присваиваются в качестве порядковых номеров последовательные числа. Первому имени присваивается порядковый номер 0, второму — 1 и т.д.

Действительные типы

В переменных действительных типов содержатся числа, состоящие из целой и дробной частей. В Delphi определено шесть действительных типов. Все типы могут представлять число 0, однако они различаются пороговым (минимальным положительным) и максимальным значениями, которые могут представлять, а также точностью (количеством значащих цифр) и объемом. Действительные типы описываются в табл. 1.

Таблица 6.1. Действительные типы

Тип	Диапазон значений	Количество значащих цифр	Объем (байт)
Real	5.0E-324..1.7E308	15-16	8
Extended	3.6E-4932 ..1.1E4932	19-20	10
Currency	-922337203685477.5808.. ..922337203685477.5807	19-20	8

Строковый тип

В Delphi для обозначения строкового типа допускается, пришедшее из Pascal, название String. Тип String — это статически размещаемые в памяти компьютера строки длиной от 0 до 255 символов. Первый его элемент задает динамическую длину строки, которая может принимать значения от 0 до 255 символов.

Переменные и константы

Переменная — это ячейка памяти компьютера, в которой находятся данные. Чтобы программа могла обратиться к переменной (получить данные для расчета, запомнить результат вычислений), переменная должна иметь имя. Имена переменных состоят из последовательности букв латинского алфавита, цифр и некоторых специальных символов. Первой в имени обязательно должна быть буква. Пробел в имени переменной использовать нельзя. Прописные и строчные буквы в именах переменных компилятором Object Pascal не различаются. Имена переменных желательно задавать так, чтобы они соответствовали логическому смыслу переменной. Например: Summa, x1, skidka. Все переменные должны быть описаны в разделе описания переменных (Var). Например:

```
Var Summa : Integer;
    Skidka : Real;
    Flag : Boolean;
```


Константы подразделяются на обычные и именованные. Обычная константа представляет собой целое или дробное число, строку символов или отдельный символ, логическое значение. Числовые константы могут выглядеть в программе так:

45

0.001

3.14

-347

0

-1.2456000000E+06 // запись в форме с плавающей точкой.

Строковые и символьные константы заключаются в апострофы. Примеры строковых и символьных констант:

'Язык программирования Object Pascal'

'3,14'

'e'

Логическая константа может принимать только одно значение из двух: ложь (False) и истина (True).

Именованная константа описывается в разделе описания констант и в программе используется уже ее имя. Например,

Const Bound =10;

Flag = True;

Pi = 3.1415926;

W = 'WORD'; .

Оператор присваивания

В общем виде оператор присваивания выглядит так:

<Имя переменной >:= <Выражение>; .

В результате выполнения оператора присваивания переменная, стоящая в левой части, получает значение выражения, стоящего справа (часто формула).

Пример:

Skidka := 12.5;

Flag := False;

S:=1/2*a*b; .

Выражение состоит из операндов и операторов. Операндами обычно бывают переменные, константы, функции. Операторы находятся между операндами и обозначают действия:

+ сложение

- вычитание

* умножение
/ деление
DIV деление нацело
MOD остаток от деления нацело
Примеры выражений:
A+B/C+0.1*C
Summa*0.85
Cena MOD 100

Структура программы на языке Delphi

Исходный текст программы состоит из последовательности строк, в которой строка может начинаться с любой позиции. Структурно программа состоит из заголовка и блока. *Заголовок* находится в начале программы и имеет вид:

Program <Имя программы>; .

Блок состоит из двух частей: описательной и исполнительной. В *описательной части* содержится описание элементов программы, в *исполнительной части* указываются действия с различными элементами программы, позволяющие получить требуемый результат.

В общем случае *описательная часть* состоит из следующих разделов:

- подключения модулей;
- объявления меток;
- объявления констант;
- описания типов данных;
- объявления переменных;
- описания процедур и функций.

В конце каждого из перечисленных разделов указывается точка с запятой.

Структуру программы в общем случае можно представить следующим образом:

```
Program <Имя программы>;  
Uses <Список модулей>;  
Label <Список меток>;  
Const <Список констант>;  
Type <Описание типов>;  
Var <Объявление переменных>;  
<Описание процедур>;
```

<Описание функций>;
Begin
<операторы>;
End.

В структуре конкретной программы любой из разделов описания и объявления может отсутствовать. Разделы описаний и объявлений, кроме раздела подключения модулей, который располагается сразу после заголовка программы, могут встречаться в программе произвольное число раз и следовать в произвольном порядке. При этом все описания и объявления элементов программы должны быть сделаны до того, как они будут использованы. Рассмотрим отдельные разделы программы.

Раздел *подключения модулей* состоит из зарезервированного слова `Uses` и списка имен подключаемых стандартных и пользовательских библиотечных модулей. Формат этого раздела:

```
Uses <Имя1>, <Имя2>, ... , <ИмяN>; .
```

Пример. Подключение модулей.

```
Uses Crt, Dos, MyLib;
```

Раздел *объявления меток* начинается зарезервированным словом `Label`, за которым следуют имена меток, разделенные запятыми. Формат этого раздела:

```
Label <Имя1>, <Имя2>, ... , <ИмяN>; .
```

Пример. Объявление меток.

```
Label metka1, metka2, 10, 567; .
```

В разделе *объявления констант* производится присваивание идентификаторам констант их постоянных значений. Раздел начинается ключевым словом `const`, за которым следует ряд конструкций, присваивающих константам значения. Эти конструкции представляют собой имя константы и выражение, значение которого присваивается константе. Имя константы отделено от выражения знаком равенства, в конце конструкции ставится точка с запятой. Формат этого раздела:

```
Const <Идентификатор1> = <Выражение>;  
<ИдентификаторN> = <Выражение>; .
```

Пример. Объявление констант.

```
Const stl = 'WORD'; ch = '5'; n34 = 45.8; .
```

Тип константы распознается компилятором автоматически на основании типа выражения.

В Delphi имеется большое количество констант, которые можно использовать без их предварительного объявления, например False и True.

В разделе *описания типов* описываются типы данных пользователя.

Этот раздел не является обязательным, и типы могут быть описаны неявно в разделе объявления переменных. Раздел описания типов начинается ключевым словом Type, за которым располагаются имена типов и их описания, разделенные знаком равенства. Каждое имя типа и его описание отделяется точкой с запятой. Формат раздела:

```
Type <Имя типа1> = <Описание типа>;  
<Имя типаN> = <Описание типа>; .
```

Пример. Описание типов.

```
Type char2 = ('a' .. 'z') ;  
NumberArray = array[1 .. 100] of real;  
Month = 1 .. 12; .
```

В Delphi имеется много *стандартных* типов, не требующих предварительного описания, например, Real, Integer, Char или Boolean.

Каждая переменная программы должна быть объявлена. Объявление обязательно предшествует использованию переменной. Раздел *объявления переменных* начинается с ключевого слова Var, после которого через запятую перечисляются имена переменных и через двоеточие их тип. Формат раздела:

```
Var <Идентификаторы> : <Тип>;  
<Идентификаторы> : <Тип>; .
```

Пример. Объявление переменных.

```
Var a, bhg, u7: real;  
symbol: char;  
n1, n2: integer; .
```

Подпрограммой называют логически законченную и специальным образом оформленную часть программы, которая по ее имени может вызываться для выполнения из других точек программы неограниченное число раз. Подпрограммы в Delphi разделяются на два вида: *процедуры* и *функции*. Каждая подпрограмма представляет собой блок и должна быть определена в

разделе *описания процедур и функций*. Описание процедур и функций рассматривается ниже.

Раздел *операторов* начинается с ключевого слова `Begin`, после которого следуют операторы языка, разделенные точкой с запятой. Завершает этот раздел ключевое слово `End`, после которого указывается точка. Формат раздела:

`Begin`

`<Оператор1>;`

`<ОператорN>;`

`End.`

Некоторые функции преобразования типов

— `IntToStr (Value: integer): string` — преобразование значения целочисленного выражения `value` в строку;

— `StrToInt (s: string): integer` — преобразование строки `s` в целое число;

— `FloatToStr(Value: Extended): string` — преобразование значения вещественного выражения `value` в строку;

— `StrToFloat (s: string): Extended` — преобразование строки `s` в вещественное число.

Среда программирования Delphi

Главные составные части среды программирования

Ниже перечислены основные составные части Delphi:

- Дизайнер Форм (Form Designer);
- Окно Редактора Кода (Editor Window);
- Палитра Компонентов (Component Palette);
- Инспектор Объектов (Object Inspector);
- Дерево Объектов (Object Tree View);
- Справочник (On-line help).

Есть, конечно, и другие важные составляющие Delphi, вроде линейки инструментов, системного меню и т.д., необходимые для точной настройки программы и среды программирования.

Дизайнер Форм показан на рис. 6.1, окно Редактора - на рис. 6.2.

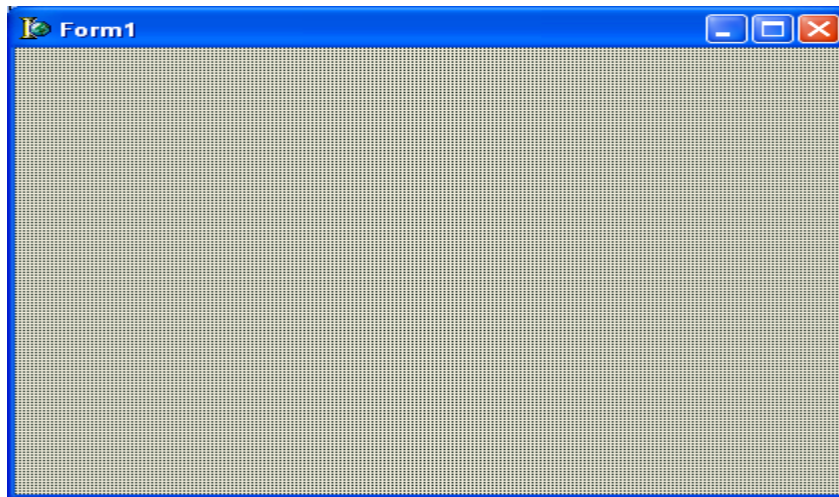


Рис. 6.1. Дизайнер Форм

Дизайнер Форм первоначально состоит из одного пустого окна, которое заполняется всевозможными объектами, выбранными на Палитре Компонентов.

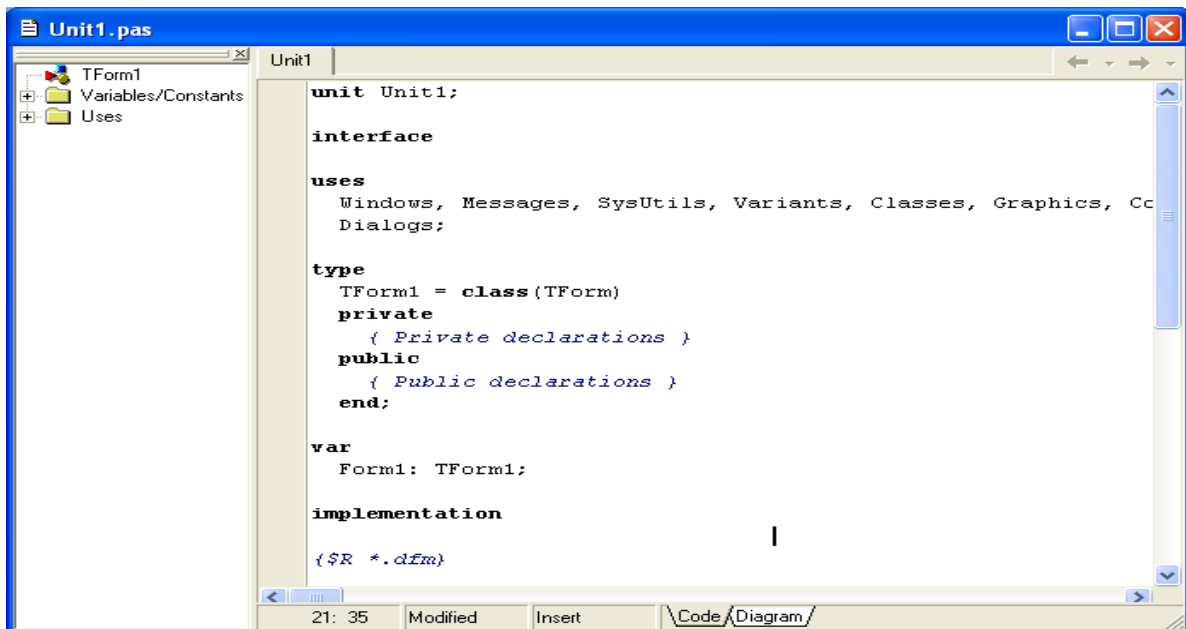


Рис. 6.2. Окно Редактора Кода

Палитра Компонентов (рис. 6.3) позволяет выбрать нужные объекты для размещения их на Дизайнере Форм. Для использования Палитры Компонентов просто первый раз щелкните мышкой на одном из объектов и потом второй раз — на Дизайнере Форм. Выбранный объект появится на проектируемом окне и им можно будет манипулировать с помощью мыши.

Палитра Компонентов использует постраничную группировку объектов. Внизу Палитры находится набор закладок —

Standard, Additional, Dialogs и т.д. Если щелкнуть мышью на одной из закладок, то можно перейти на соответствующую страницу Палитры Компонентов.



Рис. 6.3: Палитра Компонентов — место, где выбираются объекты, которые будут помещены на форму.

Ниже перечислены стандартные компоненты Delphi непосредственно используемые при изучении темы 1 с некоторыми комментариями по их применению.



Курсор - не компонент, просто пиктограмма для быстрой отмены выбора какого-либо объекта.



TLabel служит для отображения текста на экране. Можно изменить шрифт и цвет метки, если дважды щелкнуть на свойство Font в Инспекторе Объектов.



TEdit - стандартный управляющий элемент Windows для ввода. Он может быть использован для отображения короткого фрагмента текста и позволяет пользователю вводить текст во время выполнения программы.



TButton позволяет выполнить какие-либо действия при нажатии кнопки во время выполнения программы. В Delphi все делается очень просто. Поместив TButton на форму, по двойному щелчку можно вызвать заготовку обработчика события нажатия кнопки. Далее нужно заполнить заготовку кодом.

Предположим, нужно поместить компонент TEdit на форму. Можно двигать его с места на место. Можно также использовать границу, прорисованную вокруг объекта для изменения его размеров. Большинство других компонентов можно манипулировать тем же образом.

Слева от Дизайнера Форм можно видеть Инспектор Объектов (рис. 6.4). Информация в Инспекторе Объектов меняется в зависимости от объекта, выбранного на форме. Каждый компонент является настоящим объектом, можно менять его вид и поведение с помощью Инспектора Объектов.

Инспектор Объектов состоит из двух страниц, каждую из которых можно использовать для определения поведения данного компонента. Первая страница — это список свойств, вторая — список событий. Если нужно изменить что-нибудь, связанное с определенным компонентом, то обычно это делается в Инспекторе Объектов. Например, можно изменить имя и размер компонента TLabel, изменяя свойства Caption, Left, Top, Height и Width.

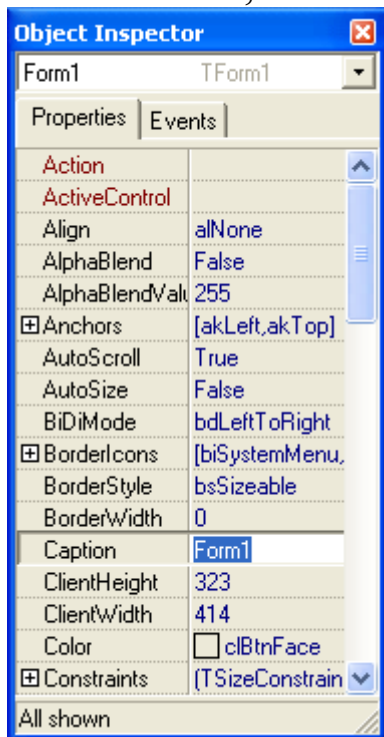


Рис. 6.4. Инспектор объектов

Справочник является контекстнозависимым; при нажатии клавиши F1 на экране появляется подсказка, соответствующая текущей ситуации. Например, находясь в Инспекторе Объектов, выберите какое-нибудь свойство и нажмите F1 — получите справку о назначении данного свойства. Если в любой момент работы в среде Delphi возникает неясность или затруднение, жмите F1 и необходимая информация появится на экране.

Дополнительные элементы

В этом разделе внимание фокусируется на двух инструментах, которые можно воспринимать как вспомогательные для среды программирования:

— Меню (Menu System);

Можно использовать закладки Инспектора Объектов для переключения между страницами свойств и событий. Страница событий связана с Редактором; если дважды щелкнуть мышкой на правую сторону какого-нибудь пункта, то соответствующий данному событию код автоматически запишется в Редактор, сам Редактор немедленно получит фокус, и можно сразу добавить код обработчика данного события.

Последняя важная часть среды Delphi — Справочник (on-line help). Для доступа к этому инструменту нужно просто выбрать в системном меню пункт Help (F1).

— Панель с кнопками для быстрого доступа (SpeedBar).

Меню предоставляет быстрый и гибкий интерфейс к среде Delphi, потому что может управляться по набору «горячих» клавиш. Это удобно еще и потому, что здесь используются слова или короткие фразы, более точные и понятные, нежели иконки или пиктограммы.

SpeedBar (панель кнопок быстрого доступа) находится непосредственно под меню, слева от Палитры Компонентов (рис.6.5). SpeedBar выполняет много из того, что можно сделать через меню. Если задержать мышь над любой из иконок на SpeedBar, то появится подсказка, объясняющая назначение данной иконки.



Рис.6.5: SpeedBar (находится слева от Палитры Компонентов)

Структура проекта

Создаваемое в среде Delphi приложение состоит из нескольких элементов, объединенных в проект. В состав проекта входят следующие элементы (в скобках указаны расширения имен файлов):

- код проекта (DPR);
- описания форм (DFM);
- модули (PAS);
- параметры проекта (DOF);
- описание ресурсов (RES);
- исполняемый файл (EXE).

В зависимости от сложности проекта могут присутствовать еще другие файлы.

Любой проект всегда содержит главный модуль (файл с расширением .dpr). Помимо главного модуля любая программа включает в себя как минимум один модуль формы (файл с расширением .pas). Сложные проекты состоят из нескольких модулей. Главный модуль можно увидеть, выбрав в главном меню команду **Project**→**View Source**. Главный модуль программы “Бак для летнего душа”, которая будет рассмотрена ниже в этой главе, выглядит так:

```
program Project2;
```

```
uses
```

```

Forms,
Unit1 in 'Unit1.pas' {Form1};

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.

```

Текст программы начинается зарезервированным словом **program** и заканчивается словом **end** с точкой за ним.

После слова **uses** указываются модули, которые используются в программе. Модуль `Forms` является уже известным Delphi, а модуль `Unit1` — новым, ранее неизвестным, и система поэтому указывает также имя файла с текстом модуля (`in 'Unit1.pas'`) и имя связанного с модулем файла описания формы `{Form1}`.

Тело программы начинается со слова **begin** (начать) и ограничивается **end** с точкой. В теле записываются операторы. Каждый из них обращается к одному из методов объекта `Application`. *Объект* — это специальным образом оформленный фрагмент программы, состоящий из данных и подпрограмм для их обработки. Данные называются *полями объекта*, а подпрограммы — *методами*. Назначение метода `Application.Initialize` — выполнить подпрограмму, имя которой размещается в системной переменной `InitProc`. Фактически этот метод ничего не делает. Метод `Application.CreateForm(TForm1, Form1)`; создает и показывает на экране окно главной формы. Метод `Application.Run`; получает и обрабатывает поток поступающих от Windows сообщений о действиях пользователя.

Файл проекта формируется самой средой Delphi и в большинстве случаев не подлежит редактированию.

Сохранение проекта

Для каждого проекта рекомендуется создавать отдельную папку.

Для сохранения проекта из меню **File** нужно выбрать команду **Save Project As**. Если проект сохраняется первый раз, то Delphi сначала предложит сохранить содержимое окна Редактора Кода, поэтому на экране появится окно `Save Unit1 As`. В этом окне надо

выбрать папку, предназначенную для файлов проекта, ввести имя модуля. Начинающим программистам советуют оставлять имя модуля сгенерированное компьютером. После нажатия кнопки **Сохранить**, появляется следующее окно, в котором необходимо задать имя файла проекта. Имена файлов модуля (pas-файл) и проекта (dpr – файл) должны быть разными. Начинающим советуют оставить данное системой имя проекта — обычно Project1.

Часто для избежания путаницы программисты пользуются командой **Save All** из меню **File** или кнопкой быстрого доступа



Особенно это важно при сохранении проектов, состоящих из нескольких форм и модулей.

Переписать проект в другое место или под другим именем можно с помощью двух команд — **Save Project As** (сохранить как) и **Save As** (сохранить как).

Структура модуля

Начинается модуль со служебного слова Unit

Модуль состоит из трех разделов: интерфейса, реализации и инициализации.

В разделе интерфейса (начинается словом interface) компилятору сообщается, какая часть является доступной для других модулей программы. В этом разделе подключаются библиотечные модули (после Uses) , используемые данным модулем. В нем находится сформированное Delphi описание формы, которое следует за словом type.

Раздел реализации открывается словом implementation и содержит объявления локальных переменных, процедур и функций, поддерживающих работу формы. Начинается раздел реализации директивой `{R *.dfm}`, указывающей компилятору, что в процессе генерации выполняемого файла надо использовать описание формы. Описание формы находится в файле dfm, имя которого совпадает с именем модуля. Файл описания формы генерируется средой Delphi на основе внешнего вида формы.

За директивой `{R *.dfm}` следуют процедуры обработки событий для формы и ее компонентов. Сюда же программист может помещать другие процедуры и функции.

Третий раздел начинающими программистами обычно не используется.

Компиляция и выполнение проекта

В процессе компиляции проекта создается готовый к использованию файл, которым может быть приложение (EXE). Запуск процесса компиляции выполняется по команде **Project | Compile <Project1>** (**Проект | Компилировать <проект>**) или нажатием комбинации клавиш «Ctrl+F9». В этой команде содержится имя проекта, разработка которого выполняется в настоящий момент, первоначально это Project1.

При сохранении проекта под другим именем соответственно должно измениться имя проекта в команде меню.

Выполнение проекта из среды Delphi осуществляется командой **Run** (Выполнить) или нажатием клавиши «F9». При этом — созданное приложение начинает свою работу. Часто используют быструю кнопку на панели кнопок быстрого доступа (на ней изображена зеленая стрелка). Если в файлы проекта вносились изменения, то предварительно выполняется компиляция проекта.

Практика

Задача

Разработать проект для вычисления объема бака для душа на даче с заданными сторонами: a , b , c (в метрах) и определения количества материала, которое пойдет на изготовление такого бака.

Метод решения:

$V=abc$ — объем бака;

$S=2(ab+bc+ca)$ — площадь поверхности бака (количество материала в квадратных метрах).

Компоненты:

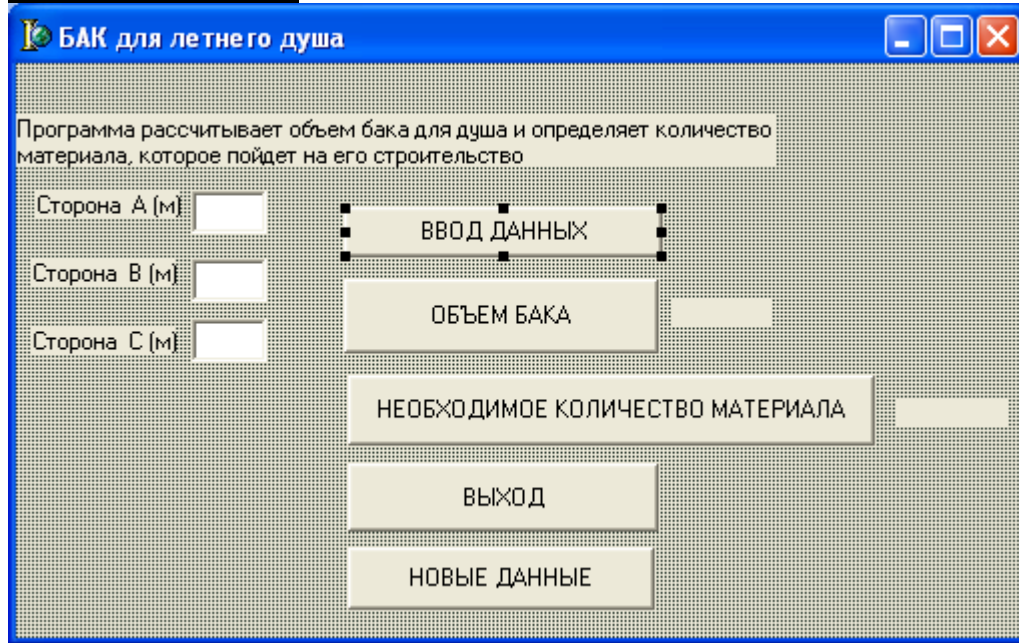
Имя компонента	Свойства компонента	Значение	Назначение
Form1	Caption	БАК для летнего душа	Заголовок формы
Label1	Caption	Программа рассчитывает объем бака для душа и определяет количество материала, которое пойдет на его изготовление	Справочная информация для пользователя программы
Label2	Caption	Сторона А (м)	Подсказка пользователю
Label3	Caption	Сторона В (м)	Подсказка пользователю
Label4	Caption	Сторона С (м)	Подсказка пользователю
Label5	Caption		Поле для вывода

			объема
Label6	Caption		Поле для вывода количества материала
Edit1	Text	Поле для ввода стороны А
Edit2	Text	Поле для ввода стороны В
Edit3	Text	Поле для ввода стороны С
Button1	Caption	ВВОД ДАННЫХ	Кнопка для ввода данных
Button2	Caption	ОБЪЕМ БАКА	Кнопка для вычисления объема бака
Button3	Caption	НЕОБХОДИМОЕ КОЛИЧЕСТВО МАТЕРИАЛА	Кнопка для вычисления необходимого количества материала
Button4	Caption	ВЫХОД	Кнопка закрытия формы и выхода из программы
Button5	Caption	НОВЫЕ ДАННЫЕ	Кнопка для очистки полей ввода для новых данных

Переменные

Обозначение в программе	Содержание	Тип
a	сторона А	Вещественный
b	сторона В	Вещественный
c	сторона С	Вещественный
V	объем бака	Вещественный
S	Площадь поверхности — количество материала	Вещественный

Проект формы



Замечание

В текстах модулей во всем учебном пособии описания и операторы, генерируемые самой системой, выделены курсивом.

В следующих темах в задачах приведены не все инструкции, генерируемые системой. Бездумное удаление инструкций, генерируемых самой системой, приводит к возникновению ошибок в проекте.

Текст модуля

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,  
Forms, Dialogs, StdCtrls;
```

```
type
```

```
TForm1 = class(TForm)
```

```
Label1: TLabel;
```

```
Label2: TLabel;
```

```
Label3: TLabel;
```

```
Label4: TLabel;
```

```
Edit1: TEdit;
```

```
Edit2: TEdit;
```

```
Edit3: TEdit;
```

```
Button1: TButton;
```

```
Button2: TButton;
```

```
Button3: TButton;
```

```
Label5: TLabel;
```

```

Label6: TLabel;
Button4: TButton;
procedure Button1Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
  a,b,c,V,S: real;
  {a, b, c — стороны бака; V — объем бака;
   S — площадь поверхности}
implementation
  {$R *.dfm}
  procedure TForm1.Button1Click(Sender: TObject);
  begin
    // ввод из полей формы значений сторон бака
    a:=StrToFloat(Edit1.Text);
    b:=StrToFloat(Edit2.Text);
    c:=StrToFloat(Edit3.Text);
    end;

  procedure TForm2.Button1Click(Sender: TObject);
  begin
    // вычисление объема
    V:=a*b*c;
    // вывод значения объема на форму
    Label5.Caption:=FloatToStrF(V,Fffixed,8,2)+' куб. м'
    end;

  procedure TForm1.Button3Click(Sender: TObject);
  begin
    // вычисление площади поверхности бака
    S:=2*(a*b+b*c+c*a);
    // вывод значения поверхности бака на форму
    Label6.Caption:=FloatToStrF(S,Fffixed,9,2)+' кв. м'
    end;

  procedure TForm1.Button4Click(Sender: TObject);
  begin
    // закрытие формы

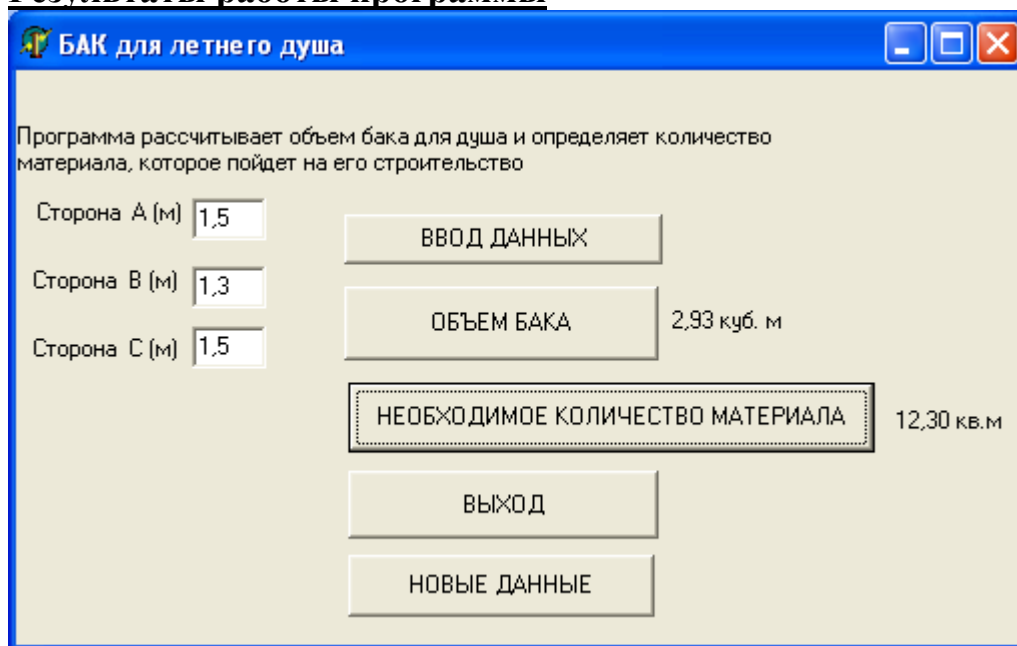
```

```

Form1.Close
end;
procedure TForm1.Button5Click(Sender: TObject);
begin
// очистка полей формы для ввода новых данных
Edit1.Clear;
Edit2.Clear;
Edit3.Clear;
Label5.Caption:=' ';
Label6.Caption:=' ';
end;
end.

```

Результаты работы программы



Задачи для самостоятельного решения

Группа А

№ 1.1. Дан радиус сферы (R) в метрах. Составить программу для вычисления площади полной поверхности

$$S = 4\pi R^2$$

и объема

$$V = 4\pi R^3 / 3.$$

№ 1.2. Известны значения хорды шарового сектора (a в см), радиуса шара (R в см) и стрелы шарового сектора (h в см). Составить программу для вычисления площади полной поверхности шарового сектора

$$S = \pi R(2h + a)$$

и объема шарового сектора

$$V = 2\pi R^2 h / 3.$$

№ 1.3. Дан круговой, прямой цилиндр. Известен радиус цилиндра R (м). Составить программу для вычисления площади боковой поверхности

$$M = 2\pi R h,$$

площади полной поверхности

$$S = 2\pi R(R + h),$$

и объема

$$V = \pi R^2 h.$$

№ 1.4. Дан усеченный круговой цилиндр. Известны: радиус основания цилиндра — R (м); h_1 и h_2 — высоты усеченного цилиндра (м). Написать программу для вычисления площади боковой поверхности

$$M = \pi R(h_1 + h_2),$$

площади полной поверхности

$$S = \pi R \left[h_1 + h_2 + R + \sqrt{R^2 + \left(\frac{h_2 - h_1}{2} \right)^2} \right],$$

и объема

$$V = \pi R \frac{h_1 + h_2}{2}.$$

№ 1.5. Известны: r и R — внутренний и внешний радиусы кругового кольца в метрах; центральный угол φ (в градусах) части кольца. Составить программу для вычисления площади кругового кольца (S_1) и площади части кольца (S_2) с заданным центральным углом по формулам

$$S_1 = \pi(R^2 - r^2), \quad S_2 = \frac{\varphi\pi}{360}(R^2 - r^2).$$

№ 1.6. Составить программу для вычисления объема трехгранной усеченной пирамиды. Если известны стороны оснований пирамиды $a_1, b_1, c_1, a_2, b_2, c_2$, высота пирамиды h , объем вычисляется по формуле

$$V = h[F + f + \sqrt{Ff}] / 3,$$

где F — площадь нижнего основания;

f — площадь верхнего основания.

Площади оснований рассчитать по формуле Герона.

№ 1.7. Написать программу для вычисления расстояния S между двумя населенными пунктами, если автомобиль со скоростью V км/час проезжает его за T часов.

Расстояние S между населенными пунктами равно $S = V \cdot T$.

Предусмотреть запросы при вводе исходных данных:

Скорость автомобиля км/час:

Время проезда в часах:

Форма вывода результата:

Расстояние между населенными пунктами равно <значение S > км.

№ 1.8. Составить программу для вычисления силы тока $I = q/t$, где q — величина электрического заряда (измеряется в кулонах); t — время прохождения заряда через поперечное сечение проводника (в секундах).

№ 1.9. Написать программу для определения объема усеченного конуса высотой H м и параллельными основаниями с радиусами R_1 м и R_2 м. В программе предусмотреть округление объема до $0,001 \text{ м}^3$. Формула для вычисления объема

$$V = H / 3(S_1 + \sqrt{S_1 S_2} + S_2),$$

где

$$S_1 = \pi R_1^2, S_2 = \pi R_2^2.$$

Форма выдачи результата:

ОБЪЕМ КОНУСА= значение V КУБ.М.

№ 1.10. Дан обелиск. Нижнее и верхнее основание являются прямоугольниками, расположенными в параллельных плоскостях; противоположные боковые грани одинаково наклонены к основанию, но не пересекаются в одной точке. Даны a , b и a_1 , b_1 — стороны оснований, h — высота. Найти объем обелиска по формуле

$$V = h[ab + (a + a_1)(b + b_1) + a_1 b_1] / 6.$$

Форма выдачи результата:

ОБЪЕМ ОБЕЛИСКА= значение V КУБ.М.

Группа Б

№ 1.11. Составить программу для исследования величины поправки (ΔS) за редуцирование расстояния (S в м) при переходе с эллипсоида на плоскость в проекции Гаусса. Для этого рассчитать поправку:

$$\Delta S = S \frac{y^2}{2R^2}$$

для случаев :

- 1) линия располагается на осевом меридиане ($y = 0$ км);
- 2) линия находится на краю шестиградусной зоны ($y \approx 330$ км).

R — радиус Земли. $R \approx 6371,1$ км.

y — ордината середины редуцируемого отрезка в км.

Поправку вычислить в метрах с точностью до 0,1 м.

№ 1.12. Составить программу для вычисления поправки ΔP в площадь за переход с поверхности шара на плоскость в проекции Гаусса:

$$\Delta P = P \frac{y^2}{R^2}$$

P — площадь участка на поверхности шара (га).

R — радиус Земли. $R \approx 6371,1$ км.

y — ордината середины редуцируемой площади (0–300 км).

Поправку округлить до 0,01 га.

№ 1.13. Составить программу для расчета цены деления уровня по формуле:

$$\tau'' = \frac{(l_1 - l_2) 206''}{nd}$$

где $l_1 - l_2$ — разность отсчетов по рейке в мм;

n — число делений, на которое сместился пузырек уровня;

d — расстояние от нивелира до рейки в м.

Перевод мм в м в рабочей формуле учтен константой 206''.

Результат вывести с точностью до 0,01 секунды.

№ 1.14. Дана арифметическая прогрессия 1-го порядка. Известны: первый член a_1 арифметической прогрессии, d — разность арифметической прогрессии. Составить программу для вычисления суммы S_n n -членов арифметической прогрессии:

$$S_n = na_1 + (n - 1)nd / 2.$$

№ 1.15. Дана геометрическая прогрессия. Известны первый член геометрической прогрессии b_1 , q ($q \neq 1$) — знаменатель геометри-

ческой прогрессии. Составить программу для вычисления суммы геометрической прогрессии:

$$S_n = a_1 \frac{q^n - 1}{q - 1} .$$

№ 1.16. Написать программу для вычисления поправки за кривизну Земли в длину измеренной линии

$$k = \frac{S^2}{2R},$$

где S — длина линии в метрах, R — радиус Земли. $R \approx 6371,1$ км. Поправку вычислить в см и округлить до 0,1 см.

Перевод в единые единицы измерения предусмотреть в формуле самостоятельно.

№ 1.17. Если известны координаты двух точек X_1Y_1 и X_2Y_2 (рис. 7–1), то координаты X, Y точки, делящей отрезок 1–2 в отношении

$\frac{m}{n}$, можно определить по формулам

$$X = \frac{nX_1 + mX_2}{m + n}, Y = \frac{nY_1 + mY_2}{m + n}.$$

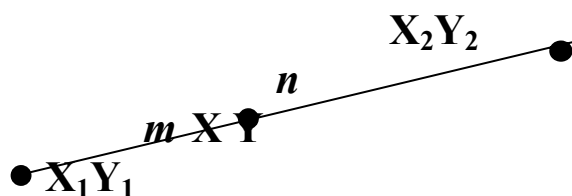


Рис. 6.6

Написать программу для определения X, Y . Значения X_1, Y_1, X_2, Y_2 вводить в метрах. Координаты X, Y выводить в метрах с точностью до 0,1м.

Форма выдачи результата:

КООРДИНАТЫ ДЕЛЯЩЕЙ ТОЧКИ:

$X =$ значение $X, Y =$ значение Y .

№ 1.18. Написать программу для вычисления поправки превышения за кривизну Земли и рефракцию:

$$f = 0,42 \frac{S_i^2}{R},$$

f — поправка измеряется в метрах. Рефракция — преломление светового луча в атмосфере (поправку вычисляют для расстояний более 300 м);

R — радиус Земли ($R \approx 6371,1$ км);

S_i — длина линии в метрах между точками, где измеряется превышение.

Поправку округлять до 0,01 м.

Форма вывода результата:

ПОПРАВКА = значение (м).

№ 1.19. Написать программу для расчета общих годовых издержек на 1 га угодий в зависимости от площади территории P и пространственного расположения земель и хозяйственного центра:

$$Z = \frac{2260}{P} + 1,48K_1K_2\sqrt{P} + 67,4,$$

где K_1 — коэффициент, характеризующий конфигурацию земельной площади и положение хозяйственного центра;

K_2 — коэффициент, показывающий, во сколько раз путь по дорогам к данному участку длиннее, чем путь по прямой.

Значение Z округлить до 0,01.

Форма вывода результата:

ГОДОВЫЕ ИЗДЕРЖКИ = значение Z руб./га.

№ 1.20. Составить программу для вычисления площади сегмента S_1 и площади сектора S_2 , если известны r — радиус круга, α — центральный угол (в градусах) по формулам:

$$S_1 = r^2 [\pi\alpha / 180) - \sin \alpha] / 2;$$

$$S_2 = \pi r^2 \alpha / 360.$$

Форма вывода результатов:

ПЛОЩАДЬ СЕГМЕНТА КРУГА = значение S_1

ПЛОЩАДЬ СЕКТОРА КРУГА = значение S_2 .

Радиус круга задан в м. Площади определить с точностью до 0,1 м².

§2. Стандартные и библиотечные функции в линейных программах

В данном параграфе рассматриваются следующие вопросы: некоторые стандартные библиотечные функции (их библиотеки подключены автоматически), библиотечные функции (некоторые математические функции), функции даты, использование окна ввода, процедура и функции вывода окон сообщений пользователю.

Теория

Некоторые стандартные библиотечные функции

Таблица 6.2 Стандартные математические функции

Функция	Возвращаемое значение
Abs(x)	Абсолютная величина x
ArgTan(x)	Арктангенс x (результат в радианах)
Dec(n)	Аналогична оператору n:=n-1
Cos(x)	Косинус x (x выражается в радианах, а не в градусах)
Exp(x)	Экспоненциальная функция от x (e^x)
Inc(n)	Аналогична оператору n:=n+1
Frac(x)	Дробная часть x
Int(x)	Целая часть x. Несмотря на название, возвращает действительное значение (с плавающей запятой), т.е. просто устанавливает нуль в дробной части
Ln(x)	Натуральный логарифм от x
pi	Константа π
Round(x)	Ближайшее к x целое значение. Возвращает значение целого типа. Условие "ближайшее к x" не работает, если верхнее и нижнее значения оказываются равноудаленными (например, если дробная часть точно равна 0,5). В этих случаях Delphi перекладывает решение на операционную систему. Обычно процессоры Intel решают эту задачу в соответствии с рекомендацией IEEE округлять в сторону ближайшего четного целого числа. Иногда такой подход называют "банкирским округлением"
Sin(x)	Синус x (x выражается в радианах)
Sqr(x)	Квадрат x, т.е. X^2
Sqrt(x)	Квадратный корень от x
Trunc(x)	Целая часть x. В отличие от Int, возвращающей дей-

	ствительное значение, Trunc возвращает целое
--	--

Замечание. В библиотеках Delphi, которые автоматически подключаются к модулю присоздания формы, нет арифметического действия «возведение в степень». Поэтому для возведения числа a в степень x можно воспользоваться следующими функциями

a^x соответствует $exp(x*ln(a))$.

См. также раздел «Некоторые математические функции».

Таблица 6.3. Функции преобразования

Функция	Значение
Chr(n)	Символ с номером n.
IntToStr(n)	Строка, являющаяся изображением значения целого n.
FloatToStr(n)	Строка, являющаяся изображением значения вещественного n. Различие между функциями описано ниже.
FloatToStrF(n,f,l,m)	Строка, являющаяся изображением значения вещественного n с форматом. Различие между функциями описано ниже.
StrToInt(s)	Целое, изображением является строка s.
StrToFloat(s)	Вещественное, изображаемое строкой s.
DateToStr (Date)	Преобразование Значения даты в выражении Date в строку
TimeToStr(Time)	Преобразование значения времени в выражении Time в строку
StrToDate(S)	Преобразование Строки S в дату
StrToTime(S)	Преобразование строки S во время
Pred(n)	Возвращает предыдущее значение n

Функция ***FloatToStrF(n,f,l,m)*** обеспечивает возможность форматного вывода вещественных чисел. Её аргументы имеют следующий смысл:

- n*** — преобразуемое значение;
- f*** — формат (способ изображения);
- l*** — точность (общее количество цифр);
- m*** — количество цифр после десятичной точки.

Поле «формат» может принимать следующие значения

ffGeneral — общий числовой формат. Значение преобразовывается в изображение десятичного числа либо с фиксированной точкой, либо записанного в научном формате. Лишние нули удаляются из результирующей строки, и в случае необходимости

вставляется десятичная точка. Запись числа с фиксированной точкой используется, если количество цифр в числе не превышает общее количество цифр, заданное пользователем, а само число не меньше 0.00001. В противном случае используется научный формат.

ffExponent — научный формат. Число преобразовывается в строку вида *"-d.ddd...E+dddd"*. Если число отрицательное, строка начинается знаком минус. Количество цифр в показателе экспоненты от 0 до 4.

ffFixed — формат с фиксированной точкой. Число преобразуется в строку вида: *"-ddd.ddd..."*. Если число отрицательное, строка начинается знаком минус. Количество цифр после десятичной точки не более 18. Если количество цифр превышает величину, указанную пользователем, изображение числа преобразуется в научный формат.

ffNumber — числовой формат. Возвращает строку в виде *"-d,ddd,ddd.ddd..."*. Он отличается от формата с фиксированной точкой только разделителями групп разрядов.

ffCurrency — денежный формат. Преобразует результат и представляет его в виде суммы денег.

Некоторые строковые функции

Таблица 6.4.

Функция	Описание
Concat(s1, s2, s3)	Возвращает последовательное соединение строк. Эквивалентна оператору s1+s2+s3
Copy(s, pos, len)	Возвращает подстроку длиной максимум len символов, начинающуюся в позиции pos строки s
Delete(s, pos, len)	Удаляет максимум len символов из строки s, начиная с позиции pos
Insert(source, target, pos)	Вставляет строку source в строковую переменную target, начиная с позиции pos
Length(s)	Возвращает динамическую длину строки.
Pos(substring, s)	Возвращает место первого вхождения подстроки substring в строку s.
Str(x, s)	Преобразует численное значение x в строковую переменную s
Val(s, v, code)	Преобразует строку s в соответствующее численное представление v. Если преобразование успешно code=0.

Некоторые библиотечные математические функции

Библиотечные математические функции находятся в библиотеке **Math**. Библиотеку **Math** нужно обязательно подключить в разделе Uses:

Uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, **Math**;

Таблица 6.5.

Функция	Возвращаемое значение
ArcTan2(y/x)	Арктангенс y/x (результат в радианах). Возвращает значение угла в несущем квадранте от -π до π. Значение x не может быть нулем
ArcCos(x)	Арккосинус x (результат в радианах)
ArcSin(x)	Арксинус x (результат в радианах)
Log10(x)	Десятичный логарифм от x
Tan(x)	Тангенс x (x в радианах)
Power(X,Y)	Возведение X в вещественную степень Y
RadToDeg	Перевод радиан в градусы, аналогично преобразованию: degrees = radians(180/pi)
DegToRad	Преобразует углы, заданные в градусах, в радианы, аналогично преобразованию: radians = degrees(pi/180)

Некоторые функции и процедуры даты

Функции даты находятся в библиотеке SysUtils, которая подключается автоматически самой системой.

Таблица 6.6

Функция	Возвращаемое значение
Date	Возвращает текущую дату, установленную на компьютере. Функция не имеет аргументов
DecodeDate(Dat, Y,M,D)	Процедура возвращает дату в виде трех величин Y(год), M(месяц), D(день)
EncodeDate(Y,M,D)	Функция преобразует три величины Y(год), M(месяц), D(день) в дату
YearsBetween (An,At)	Функция возвращает количество прошедших лет между двумя датами An и At
DaySpan(An,At)	Функция возвращает количество дней, прошедших между двумя датами An и At.

Если в программе необходимо использовать переменные, содержащие дату, то их нужно описать типом **TDateTime** в разделе

описания переменных. **TDateTime** — это стандартный объект Delphi.

Ввод из окна ввода

Функция **InputBox** выводит на экран стандартное диалоговое окно — окно ввода. Значение функции **InputBox** — строка, которую ввел пользователь. Формат функции **InputBox** следующий:

Переменная:= **InputBox**(Заголовок, Подсказка, Значение);

где:

Переменная — переменная строкового типа, значение которой должно быть получено от пользователя;

Заголовок — текст, выводимый в заголовок окна;

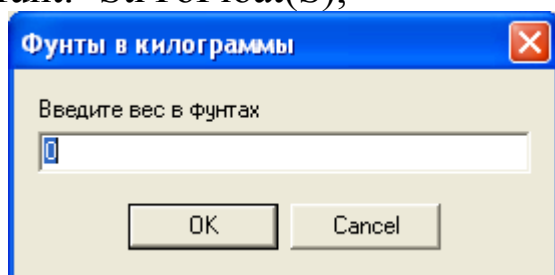
Подсказка — текст поясняющего сообщения;

Значение — текст, который будет находиться в поле ввода, когда окно ввода появится на экране.

Например, следующая инструкция вызывает появление на экране окна ввода, показанного ниже.

S:=**InputBox**('Фунты в килограммы','Введите вес в фунтах','0');

funt:=**StrToFloat**(S);



Если во время работы программы пользователь введет строку и щелкнет на кнопке **OK**, то значением функции **InputBox** будет введенная строка, если — на кнопке **Cancel**, то значением функции будет строка, переданная функции в качестве параметра «Значение» (в примере 0).

Вывод в окно сообщения

Вывести на экран окно с сообщением можно при помощи процедуры **ShowMessage** или функции **MessageDlg**.

Процедура **ShowMessage** выводит на экран окно с текстом и командной кнопкой **OK**.

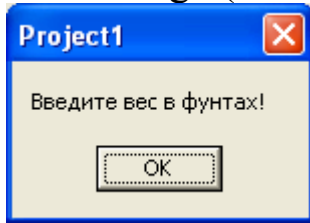
Формат процедуры **ShowMessage** :

ShowMessage (Сообщение);

где **Сообщение** — текст, который будет выведен в окне.

Например,

ShowMessage ('Введите вес в фунтах');



Функция MessageDlg позволяет поместить в окно с сообщением один из стандартных значков, задать количество и тип командных кнопок и определить, какую из кнопок нажал пользователь. Формат функции

MessageDlg следующий:

Выбор:=MessageDlg(Сообщение, Тип, Кнопки, КонтекстСправки);

Где Сообщение — текст сообщения; Тип — тип сообщения.

Тип сообщения задается именованной константой:

Константа	Тип сообщения
MtWarning	Внимание
MtError	Ошибка
MtInformation	Информация
MtConfirmation	Подтверждение
MtCustom	Обычное

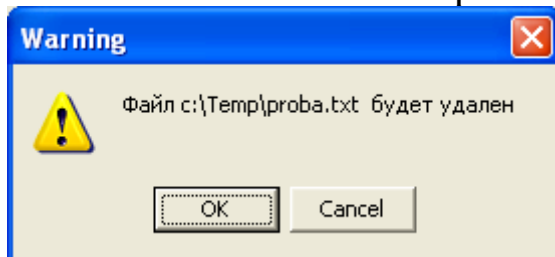
Кнопки — список кнопок, отображаемых в окне сообщения. Список может состоять из нескольких разделенных запятыми именованных констант. Весь список заключается в квадратные скобки.

Константа	Кнопка	Константа	Кнопка
mbYes	Yes	mbAbort	Abort
mbNo	No	mbRetry	Retry
mbOk	Ok	mbIgnore	Ignore
mbCancel	Cancel	mbAll	All
mbHelp	Help		

Например, оператор

r:=MessageDlg('Файл'+ Fname + ' будет удален',
mtWarning, [mbOk,mbCancel], 0);

вызовет появление на экране окна:



Замечание

Переменная r должна быть описана типом Word.

Практика

Задача

Дано: гипотенуза и прилежащий к гипотенузе угол прямо-
угольного треугольника. Составить программу для вычисления
катета, противолежащего заданному углу. Формула из тригоно-
метрии: $a=c*\sin A$.

Компоненты

Имя компонента	Свойства компонента	Значение	Назначение
Form1	Caption	Катет прямоугольного тре- угольника	Заголовок формы
Label1	Caption WordWrap	Программа вычисляет ка- тет прямоугольного тре- угольника по гипотенузе и противолежащему углу True	Справочная ин- формация для пользователя программы Перенос не уме- стившихся слов на новую строку
Label2	Caption	Гипотенуза=	Подсказка поль- зователю
Edit1	Text		Поле для ввода значения гипоте- нузы
Label3	Caption	Противолежащий угол (гр., мин., сек.)=	Подсказка поль- зователю
Edit2	Text		Поле для ввода градусов
Edit3	Text		Поле для ввода минут
Edit4	Text		Поле для ввода секунд
Label4	Caption		Поле для вывода результата
Button1	Caption	ВЫЧИСЛИТЬ	Кнопка вычисле- ния катета
Button2	Caption	ОЧИСТИТЬ	Кнопка для очи- щения полей для ввода новых дан- ных
Button3	Caption	ВЫХОД	Кнопка прекра- щения выполне- ния программы и закрытия формы

Переменные

Обозначение в программе	Содержание	Тип
a	Определяемый катет прямоугольного треугольника	Вещественный
c	Гипотенуза	Вещественный
G	Градусы противоположащего определяемому катету угла	Целый
M	Минуты противоположащего определяемому катету угла	Целый
S	Секунды противоположащего определяемому катету угла	Вещественный

Проект формы

Текст модуля

unit Unit1;

.....

var

Form1: TForm1;

a:Real; // катет прямоугольного треугольника

c:Real; // гипотенуза прямоугольного треугольника

G,M:Integer;

S:Real;

Rad:Real;

implementation

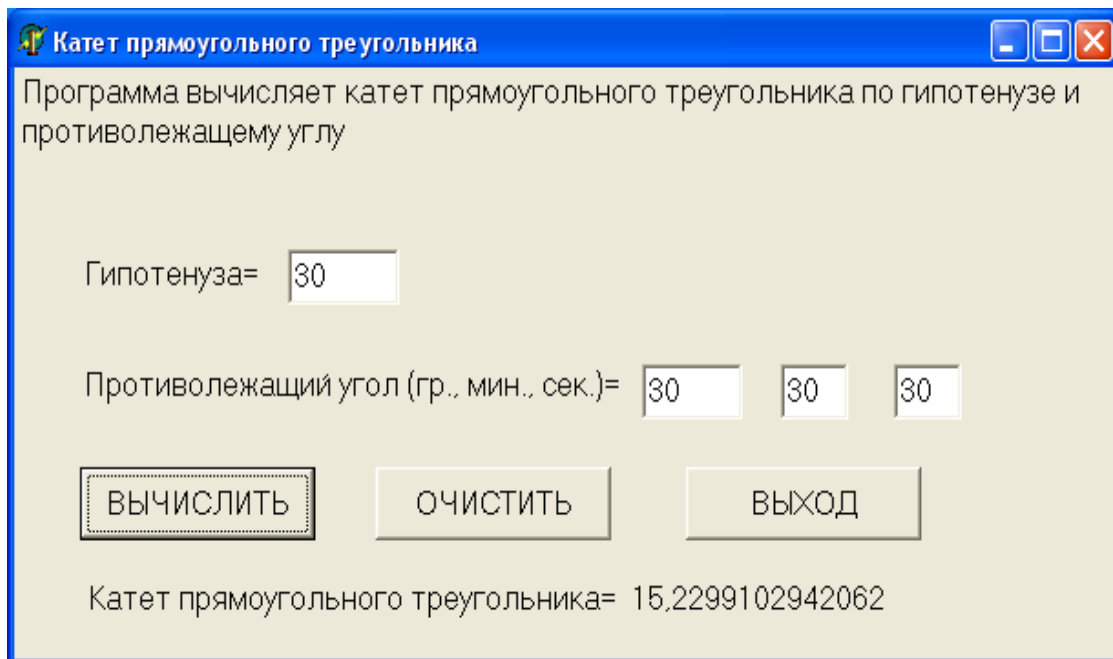
*{ \$R *.dfm }*

```

procedure TForm1.Button1Click(Sender: TObject);
begin
c:=StrToFloat(Edit1.Text);
G:=StrToInt(Edit2.Text);
M:=StrToInt(Edit3.Text);
S:=StrToFloat(Edit4.Text);
Rad:=(G+(M+S/60)/60)*PI/180;
a:=c*sin(Rad);
Label4.Caption:='Катет прямоугольного треугольника= '+FloatToStr(a);
end;
//процедура для очистки полей формы для новых данных
procedure TForm1.Button2Click(Sender: TObject);
begin
Edit1.Clear;
Edit2.Clear;
Edit3.Clear;
Edit4.Clear;
Label4.Caption:= ''
end;
procedure TForm1.Button3Click(Sender: TObject);
begin
Form1.Close
end;
end.

```

Результаты работы программы



Задача

Составить программу, определяющую сколько дней прожил пользователь программы.

Компоненты

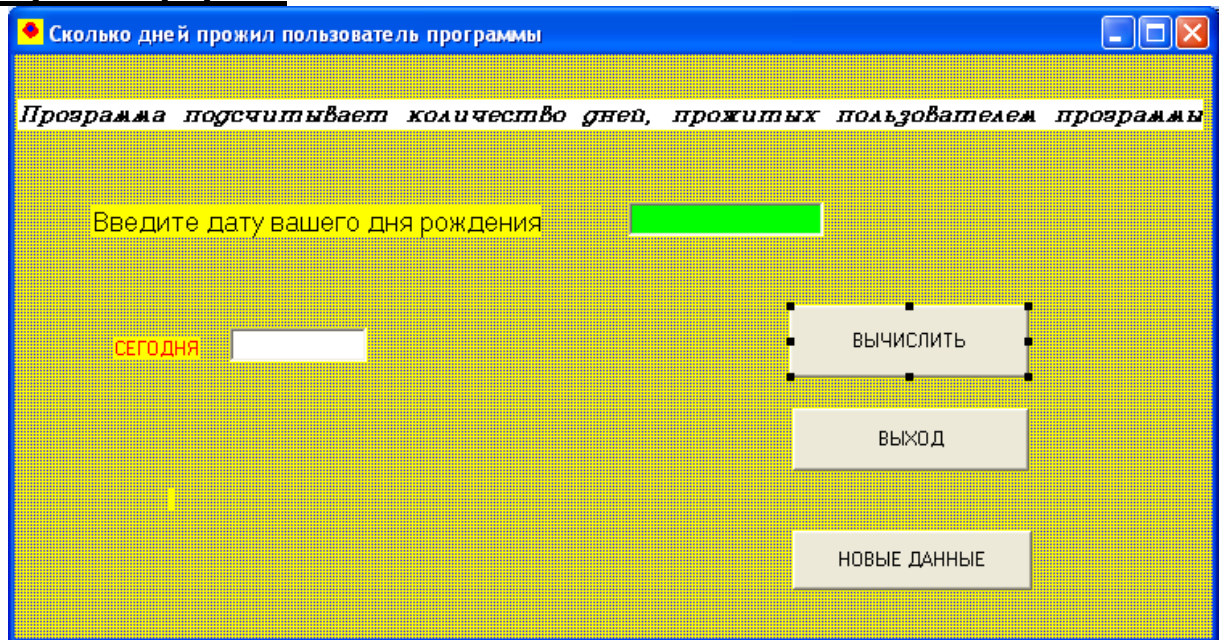
Имя компонента	Свойства компонента	Значение	Назначение
Form1	Caption	Сколько дней прожил пользователь программы	Заголовок формы
Label1	Caption WordWrap	Программа подсчитывает, сколько дней прожил пользователь программы True	Справочная информация для пользователя программы Перенос не уместившихся слов на новую строку
Label2	Caption	Введите дату вашего рождения	Подсказка пользователю
Edit1	Text		Поле для ввода даты рождения
Label3	Caption	Сегодня	Подсказка пользователю
Edit2	Text		Поле, в которое выводится дата, установленная на компьютере
Label4	Caption		Поле для вывода результата

Button1	Caption	ВЫЧИСЛИТЬ	Кнопка вычисления количества дней
Button2	Caption	ВЫХОД	Кнопка прекращения выполнения программы и закрытия формы
Button3	Caption	ОЧИСТИТЬ	Кнопка для очищения полей для ввода новых данных

Переменные

Обозначение в программе	Содержание	Тип
USER_DAT	Дата рождения пользователя программы	TDateTime
USER_DAY	Количество дней, прожитых пользователем программы	Длинный целый

Проект формы



Текст модуля

```
unit Unit1;
```

```
.....  
var
```

```
Form1: TForm1;
```

```
implementation
```

```
{ $R *.dfm }
```

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
Var USER_DAT:TDateTime;
```

```
    USER_DAY:INT64;
```

```
begin
```

```
Edit2.Text:=DateToStr(Date());
```

```
Label4.Caption:='ВЫ ПРОЖИЛИ '+
```

```
FloatToStr(DaySpan(Date(),StrToDate(Edit1.Text)))+ ' дней'
```

```
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
```

```
begin
```

```
Form1.Close
```

```
end;
```

```
procedure TForm1.Button3Click(Sender: TObject);
```

```
begin
```

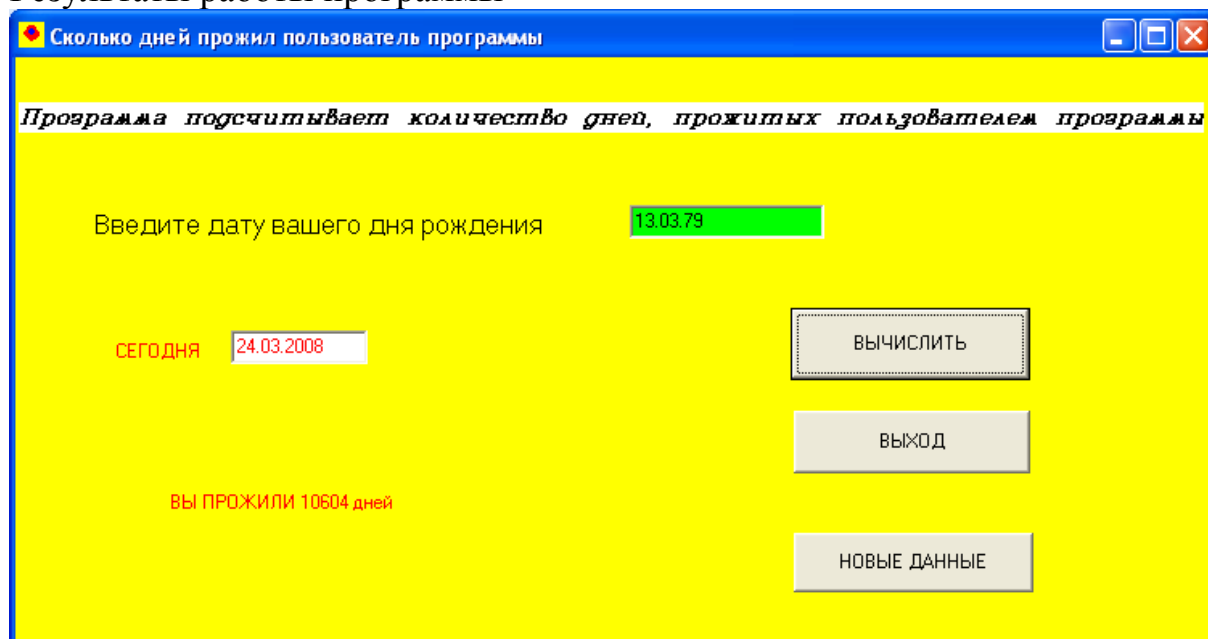
```
Edit1.Clear;
```

```
Label4.Caption:=' ';
```

```
end;
```

end.

Результаты работы программы



Задача

Составить программу для решения прямой и обратной геодезических задач.

Прямая геодезическая задача состоит в том, что по координатам одного конца $A (X_A, Y_A)$ линии AB , по дирекционному углу этой линии α_{AB} и ее горизонтальному проложению S_{AB} вычисляют координаты другого конца B этой линии (X_B, Y_B) .

$$X_B = X_A + S_{AB} \cos \alpha_{AB},$$

$$Y_B = Y_A + S_{AB} \sin \alpha_{AB}.$$

Обратная геодезическая задача состоит в том, что по координатам концов линии AB вычисляют дирекционный угол и горизонтальное проложение этой линии. То есть известны X_A, Y_A, X_B, Y_B . Надо вычислить α_{AB} и S_{AB} .

$$\operatorname{tg} \alpha_{AB} = \frac{Y_B - Y_A}{X_B - X_A},$$

$$S_{AB} = \sqrt{(X_B - X_A)^2 + (Y_B - Y_A)^2}.$$

Компоненты

Имя компонента	Свойства компонента	Значение	Назначение
Form1	Caption	Прямая и обратная геодезические задачи	Заголовок формы
Label1	Caption	Обратная геодезическая	Справочная ин-

	Visible	задача — вычисление по координатам двух точек горизонтального проложения линии между ними и дирекционного угла	формация для пользователя программы Невидимый
Label2	Caption	XA	Подсказка пользователю
	Visible	False	Невидимый
Label3	Caption	YA	Подсказка пользователю
	Visible	False	Невидимый
Label4	Caption	XB	Подсказка пользователю
	Visible	False	Невидимый
Label5	Caption	YB	Подсказка пользователю
	Visible	False	Невидимый
Label6	Caption	Горизонтальное проложение линии АВ в метрах	Подсказка пользователю
	Visible	False	Невидимый
Label7	Caption		Поле для вывода горизонтального проложения
	Visible	False	Невидимый
Label8	Caption	Дирекционный угол (гр. мин. сек.)	Подсказка пользователю
	Visible	False	Невидимый
Label9	Caption		Поле для вывода дирекционного угла (градусы)
	Visible	False	Невидимый
Label10	Caption		Поле для вывода дирекционного угла (минуты)
	Visible	False	Невидимый
Label11	Caption		Поле для вывода дирекционного угла (секунды)
	Visible	False	Невидимый
Edit1	Text	Поле для ввода координаты XA
	Visible	False	Невидимый

Edit2	Text	Поле для ввода координаты YA Невидимый
	Visible	False	
Edit3	Text	Поле для ввода координаты XB Невидимый
	Visible	False	
Edit4	Text	Поле для ввода координаты YB Невидимый
	Visible	False	
Button1	Caption	ПРЯМАЯ ГЕОДЕЗИЧЕСКАЯ ЗАДАЧА	Кнопка запускает на выполнение прямую геодезическую задачу
Button2	Caption	ОБРАТНАЯ ГЕОДЕЗИЧЕСКАЯ ЗАДАЧА	Кнопка делает видимыми поля для решения обратной геодезической задачи
Button3	Caption	ВЫХОД	Кнопка закрытия формы и выхода из программы
Button4	Caption	Вычислить	Кнопка вычисления обратной геодезической задачи
	Visible	False	Невидимый

Переменные

Обозначение в программе	Содержание	Тип
XA	Координата X точки A	Вещественный
YA	Координата Y точки A	Вещественный
XB	Координата X точки B	Вещественный
YB	Координата Y точки B	Вещественный
G	Часть дирекционного угла линии AB в градусах	Целый
M	Часть дирекционного угла линии AB в минутах	Целый
S	Часть дирекционного угла линии AB в секундах	Вещественный
L	Горизонтальное положение линии AB	Вещественный
DX	Приращение координат по оси X	Вещественный
DY	Приращение координат по оси Y	вещественный
DU	Дирекционный угол в градусах	Вещественный

Замечание

При проектировании компонентов для решения обратной геодезической задачи на той же форме свойство Visible компонентов целесообразно установить в положение False.

Текст модуля

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, StdCtrls, Math;
{ библиотечный модуль Math нужно добавить в стандартный набор
  модулей для подключения математической библиотеки }
.....
var
  Form1: TForm1;
  XA, YA: Extended; // координаты точки А
  XB, YB: Extended; // координаты точки В
  G: Integer; // дирекционный угол в градусах
  M: Integer; // часть дирекционного угла в минутах
  S: Real; // часть дирекционного угла в секундах
  L: Extended; // горизонтальное проложение линии АВ
  DX, DY: Extended; // приращения координат по осям X, Y
  DU: Extended; // дирекционный угол в радианах
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
begin
  ShowMessage('Прямая геодезическая задача - определение
координаты'+
' точки по координатам исходной точки, горизонтальному проложению '
+ ' и дирекционному углу.'+#13
+'Дирекционный угол задается в градусах, минутах и секундах. Коор-ты в метрах. ');
  // ввод координат исходной точки, дирекционного угла
  // и горизонтального проложения линии АВ из окон ввода
  XA:=StrToFloat(InputBox('Прямая геодезическая задача',
'Введите координату X точки А','0'));
  YA:=StrToFloat(InputBox('Прямая геодезическая задача',
'Введите координату Y точки А','0'));
  G:=StrToInt(InputBox('Прямая геодезическая задача',
'Введите часть дирекционного угла линии АВ в градусах' + #13
+'Значение < 360','0'));
  M:=StrToInt(InputBox('Прямая геодезическая задача',
```

```

'Введите часть дирекционного угла линии АВ в минутах'+
#13+'Значение <60','0'));
    S:=StrToFloat(InputBox('Прямая геодезическая задача',
    'Введите часть дирекционного угла линии АВ в секундах'
+#13+'Значение <60','0'));
    L:=StrToFloat(InputBox('Прямая геодезическая задача',
    'Введите горизонтальное проложение линии АВ в метрах','0'));
    // вычисление координат точки В
    // в ниже приведенном преобразовании можно использовать
    // функцию преобразования DegTORad
    XB:=XA+L*cos((g+m/60+s/3600)/180*pi);
    YB:=YA+L*sin((g+m/60+s/3600)/180*pi);
    {XB:=XA+L*cos(DegToRad(g+m/60+s/3600));
    YB:=YA+L*sin(DegToRad (g+m/60+s/3600));}
    //округление вычисленных координат
    // до двух знаков после запятой
    XB:=Trunc(XB*100+0.5)/100;
    YB:=Trunc(YB*100+0.5)/100;
    // вывод координат точки В окно сообщения
    ShowMessage('Прямая геодезическая задача ' + #13 + 'XB='
+FloatToStrF(XB,Ffixed,10,2)+ #13
    + 'YB='+FloatToStrF(YB,Ffixed,10,2))
    end;
    procedure TForm1.Button3Click(Sender: TObject);
    begin
    Form1.Close
    end;
    procedure TForm1.Button2Click(Sender: TObject);
    begin
    // сделать видимыми компоненты, относящиеся к обратной
    // геодезической задаче
    Label1.Visible:=True;
    Label2.Visible:=True;
    Label3.Visible:=True;
    Label4.Visible:=True;
    Label5.Visible:=True;
    Label6.Visible:=True;
    Label7.Visible:=True;
    Label8.Visible:=True;
    Label9.Visible:=True;
    Label10.Visible:=True;
    Label11.Visible:=True;
    Edit1.Visible:=True;

```

```

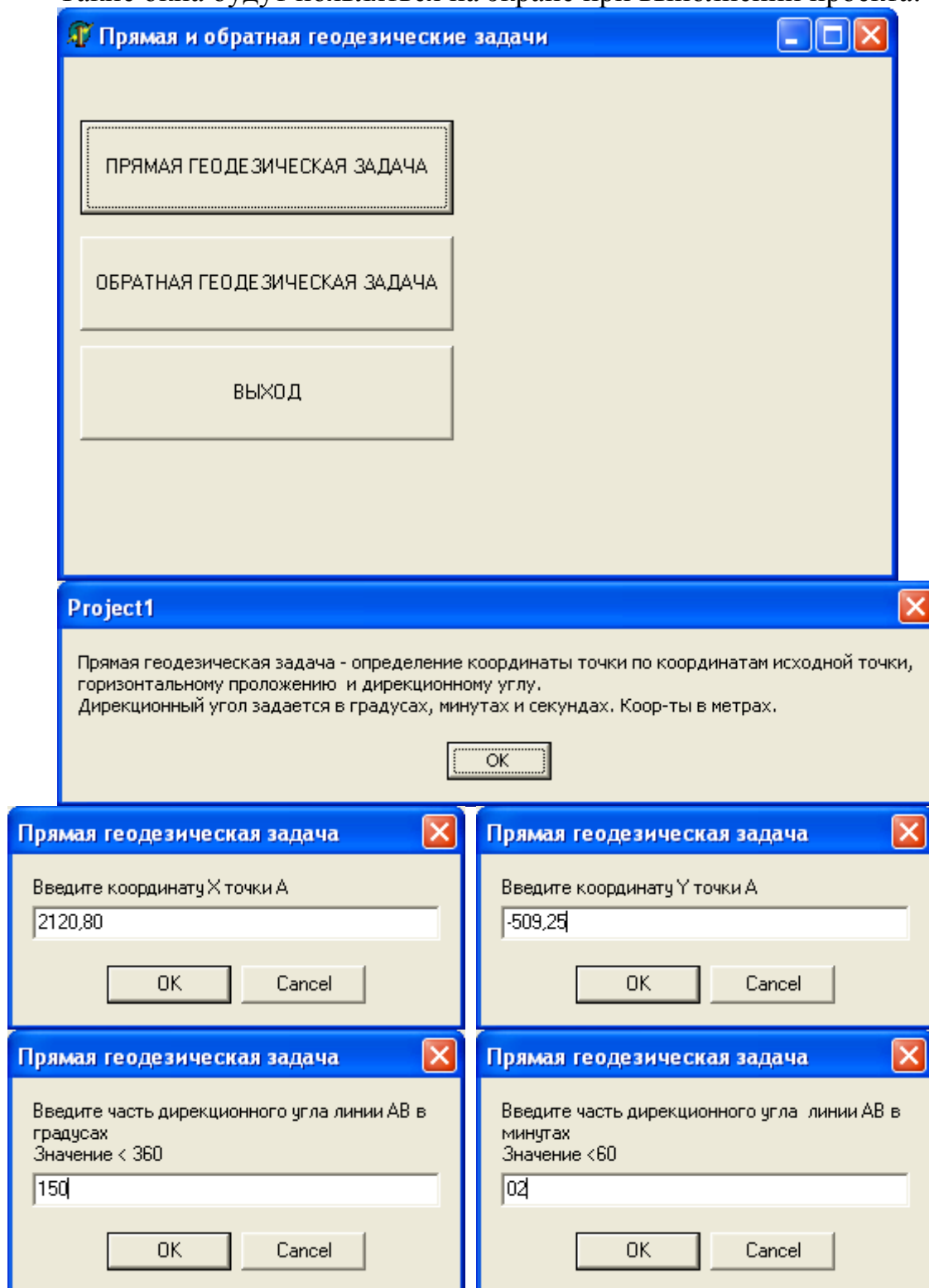
Edit2.Visible:=True;
Edit3.Visible:=True;
Edit4.Visible:=True;
// перенести фокус в первое поле ввода на форме
Edit1.SetFocus;
Button4.Visible:=True;
end;
procedure TForm1.Button4Click(Sender: TObject);
    Var
begin
// ввод координат точек А и В из полей ввода на форме
XA:=StrToFloat(Edit1.Text);
YA:=StrToFloat(Edit2.Text);
XB:=StrToFloat(Edit3.Text);
YB:=StrToFloat(Edit4.Text);
DX:=XB-XA;
DY:=YB-YA;
// вычисление горизонтального проложения линии АВ
L:=Sqrt(Sqr(DX)+Sqr(DY));
// округление горизонтального проложения
// до двух знаков после запятой
L:=Trunc(L*100+0.5)/100;
// вычисление дирекционного угла и перевод
// его значения из радиан в градусы
DU:=ArcTan2(DY,DX)/Pi*180;
{в данном преобразовании можно было использовать функцию
RadToDeg: DU:= RadToDeg (ArcTan2(DY,DX));}
// ArcTan2(Y/X) - библиотечная функция находит угол X/Y
// в несущем квадранте по знакам X и Y от -Pi до Pi в радианах.
// если угол со знаком минус к нему прибавляется 360 градусов
if DU<0 Then DU:=360+DU;
//выделение градусной части дирекционного угла
G:=Trunc(DU);
// выделение минутной части дирекционного угла
M:=Trunc((DU-G)*60);
// выделение секундной части дирекционного угла
S:=((DU-G)*60-M)*60;
{вывод горизонтального проложения и дирекционного угла в поля вы-
вода}
Label7.Caption:=FloatToStrF(L,Fffixed,10,2);
Label9.Caption:=IntToStr(G);
Label10.Caption:=IntToStr(M);
Label11.Caption:=FloatToStrF(S,Fffixed,5,1);

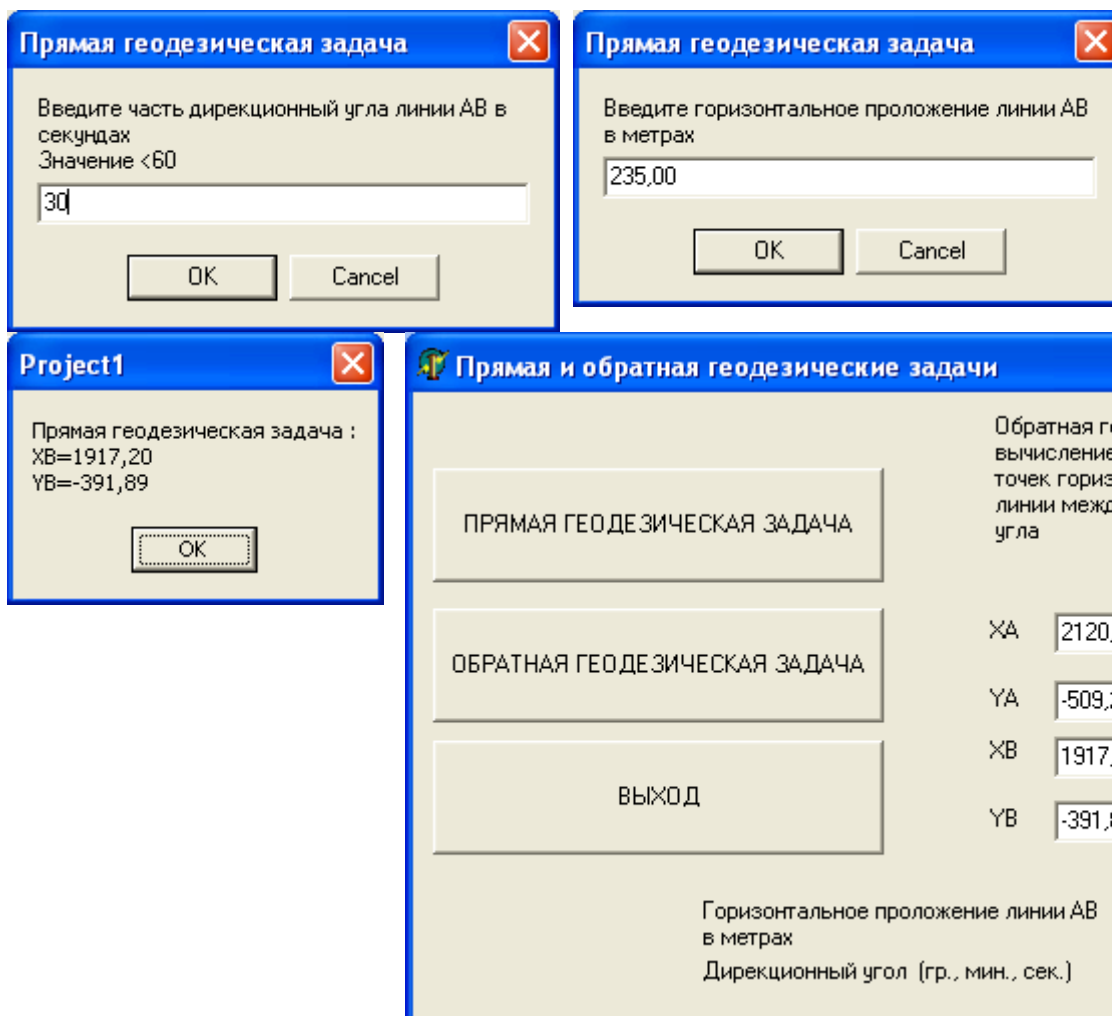
```

end;

end.

Такие окна будут появляться на экране при выполнении проекта:





Задачи для самостоятельного решения

Группа А

№ 2.1. Даны x, y, z . Составить программу для вычисления a, b по формулам:

$$a = \frac{\sqrt{|x-1|} - \sqrt[3]{|y|}}{1 + \frac{x^2}{2} + \frac{y^2}{4}}, b = x(\arctg z + e^{-(x+3)}).$$

№ 2.2. Даны x, y, z . Составить программу для вычисления a, b по формулам:

$$a = \frac{3 + e^{y-1}}{1 + x^2 |y - \operatorname{tg} z|}, b = 1 + |y - x| + \frac{(y-x)^2}{2} + \frac{|y-x|^3}{3}.$$

№ 2.3. Даны x, y, z . Составить программу для вычисления a, b по формулам:

$$a = (1 + y) \frac{x + y/(x^2 + 4)}{e^{-x-2} + 1/(x^2 + 4)}, b = \frac{1 + \cos(y - 2)}{x^4 / 2 + \sin^2 z}.$$

№ 2.4. Даны x, y, z . Составить программу для вычисления a, b по формулам:

$$a = y + \frac{x}{y^2 + \left| \frac{x^2}{y + x^3/3} \right|}, b = (1 + \operatorname{tg}^2 \frac{z}{2}).$$

№ 2.5. Даны x, y, z . Составить программу для вычисления a, b по формулам:

$$a = \frac{2 \cos(x - \pi/6)}{1/2 + \sin^2 y}, b = 1 + \frac{z^2}{3 + z^2/5}.$$

№ 2.6. Даны x, y, z . Составить программу для вычисления a, b по формулам:

$$a = \frac{1 + \sin^2(x + y)}{2 + |x - 2x/(1 + x^2 y^2)|} + x, b = \cos^2(\operatorname{arctg} \frac{1}{z}).$$

№ 2.7. Составить программу, подсчитывающую сколько дней осталось до каникул.

№ 2.8. Даны действительные числа δ и σ . Найти $p(\delta)$ по формуле:

$$p(\delta) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{\delta^2}{2\sigma^2}}.$$

№ 2.9. Составить программу, подсчитывающую количество символов в полном имени пользователя программы, включая пробелы.

№ 2.10. Составить программу, определяющую, сколько лет прожил пользователь программы.

Группа Б

№ 2.11. Написать программу для решения прямой геодезической задачи: по заданным координатам точки $A(X_A, Y_A)$, горизонтальному проложению (L в м) от точки A до точки B и дирекционному углу (α) этого проложения найти координаты точки B :

$$X_B = X_A + L \cdot \cos \alpha$$

$$Y_B = Y_A + L \cdot \sin \alpha$$

Угол вводить в градусах, минутах и секундах, координаты вводить с точностью до 1 м.

В программе предусмотреть следующие запросы при вводе исходных данных:

XA =, YA =

ПРОЛОЖЕНИЕ АВ =

ДИР. УГОЛ АВ: G, M, S =

Форма вывода результата:

XВ = значение XВ YВ = значение YВ.

Примечание

Горизонтальное проложение — ортогональная проекция линии местности на горизонтальную плоскость.

Дирекционный угол — это горизонтальный угол, отсчитываемый от северного направления линии, параллельной оси абсцисс, по ходу часовой стрелки до направления данной линии.

№2.12. Написать программу, которая по заданному значению вертикального угла ν и расстоянию S (в м) между точками А и В определяет превышение точки В над точкой А

$$R = S \cdot \sin \nu$$

и горизонтальное проложение между А и В

$$D = S \cdot \cos \nu$$

Значение R округлять до 0,01 м, значение D — до 0,1 м.

ВЕРТИКАЛЬНЫЙ УГОЛ: G, M, S =

РАССТОЯНИЕ =

Форма вывода результатов:

ПРЕВЫШЕНИЕ = значение R

ГОР. ПРОЛОЖЕНИЕ = значение D

(См. примечание к задаче 2.11).

№ 2.13. Написать программу для вычисления поправки дирекционного угла в сек. по значениям дирекционного угла α , расстояния S в метрах и поправок координат концов отрезка $\delta'_x, \delta'_y, \delta''_x, \delta''_y$ в метрах:

$$\delta_\alpha = \frac{206000}{S} (\sin \alpha \cdot (\delta'_x - \delta'_y) + \cos \alpha \cdot (\delta''_x - \delta''_y)).$$

Поправку округлить до 0,01 сек.

В программе предусмотреть следующие запросы при вводе исходных данных:

ДИР. УГОЛ: ГР., МИН., СЕК. =
РАССТОЯНИЕ =
ПОПРАВКИ КООРДИНАТ:

DX1 = вводить δ'_x

DY1 = вводить δ'_y

DX2 = вводить δ''_x

DY2 = вводить δ''_y

Форма вывода результата:

ПОПРАВКА ДИР. УГЛА= значение в сек.

(См. примечание к задаче 2.11).

№ 2.14. Написать программу для вычисления площади треугольного участка по длине стороны А и прилежащих к ней углов β и γ . Значения углов вводить в градусах, минутах и секундах, длину стороны А — в метрах. Площадь вывести в гектарах с точностью до 0,1 га.

Формула для вычисления площади:

$$S = \frac{A^2 \cdot \sin \beta \cdot \sin \gamma}{2 \cdot \sin(\beta + \gamma)} .$$

В программе предусмотреть следующие запросы при вводе исходных данных:

УГОЛ В: ГР., МИН., СЕК. =

УГОЛ С: ГР., МИН., СЕК. =

СТОРОНА А =

Форма вывода результата:

ПЛОЩАДЬ УЧАСТКА = значение S га.

№ 2.15. Написать программу для перевода декартовых координат (X, Y) некоторой точки в полярные (ρ, Θ) при условии, что полюс совпадает с началом декартовой системы, а полярная ось совпадает с ось ОХ:

$$\rho = \sqrt{X^2 + Y^2}$$
$$\Theta = \arctg \frac{Y}{X} .$$

Значение ρ округлять до 0,1 м. Значение Θ выводить в градусах, минутах и секундах, округляя секунды до 0,1.

В программе предусмотреть следующие запросы при вводе исходных данных:

ДЕКАРТОВЫ КООРДИНАТЫ (X,Y) =

Форма вывода результатов:

ПОЛЯРНЫЕ КООРДИНАТЫ:

УГОЛ (ГР., МИН., СЕК.) = значение

РАССТОЯНИЕ = значение.

№2.16 Написать программу для перевода координат точки М из полярной системы в декартову, считая, что полюс совпадает с началом декартовых координат, а полярная ось совпадает с осью ОХ:

$$X = \rho \cos \Theta, Y = \rho \sin \Theta.$$

Здесь ρ и Θ — координаты точки М в полярной системе, где Θ выражается в градусах, минутах и секундах.

Значения X,Y при выводе округлить до 0,01 м.

В программе предусмотреть следующие запросы при вводе исходных данных:

ПОЛЯРНЫЕ КООРДИНАТЫ:

УГОЛ (ГР., МИН., СЕК.) =

РАССТОЯНИЕ (МЕТРЫ) =

Форма вывода результатов:

ДЕКАРТОВЫ КООРДИНАТЫ:

X = значение X МЕТРОВ Y=значение Y МЕТРОВ.

№2.17. Написать программу для вычисления коэффициента нагрузки лесной полосы (R) по заданному значению угла α , образуемого осью лесной полосы и направлением ветра:

$$R = \sin(\alpha + 6^\circ)(1 - \operatorname{tg}(45^\circ - \frac{\alpha}{2})).$$

Значение α вводить в градусах и минутах.

В программе предусмотреть следующие запросы при вводе:

УГОЛ (ГР., МИН.) =

Форма вывода результатов:

КОЭФФИЦИЕНТ НАГРУЗКИ = значение R.

№ 2.18. Написать программу для вычисления площади треугольного участка по длине стороны А и прилежащих к ней углов β и γ . Значения углов вводить в градусах, минутах и секундах, длину

стороны A — в метрах. Площадь вывести в гектарах с точностью до 0,1 га.

Формула для вычисления площади:

$$S = \frac{A^2 \sin \beta \sin \gamma}{2 \sin(\beta + \gamma)}.$$

В программе предусмотреть следующие запросы при вводе исходных данных:

УГОЛ B : ГР., МИН., СЕК. =

УГОЛ C : ГР., МИН., СЕК. =

СТОРОНА A =

Форма вывода результатов:

ПЛОЩАДЬ УЧАСТКА = значение S га.

№ 2.19. Написать программу для определения угла (рис. 6.2) по заданным значениям X и S . Угол α определять в градусах, минутах и секундах, округлив секунды до 0,1 сек.

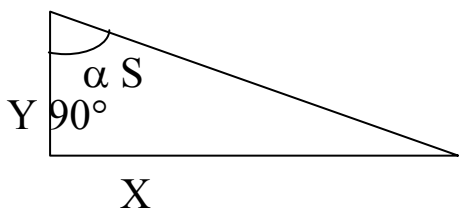


Рис. 6.2

В программе предусмотреть запросы при вводе исходных данных:

ГИПОТЕНУЗА =

ПРОТИВОЛЕЖАЩИЙ КАТЕТ =

Форма вывода результата:

УГОЛ = количество гр., количество минут мин., количество секунд сек.

№ 2.20. Написать программу для определения угла α (рис.) по заданным значениям Y и S . Угол определять в градусах, минутах и секундах, округляя секунды до 0,1.

В программе предусмотреть запросы при вводе исходных данных:

ГИПОТЕНУЗА =

ПРИЛЕЖАЩИЙ КАТЕТ =

Форма вывода результата:

Угол = градусы гр., минуты мин., секунды сек.

§ 3. Программы с разветвлениями

В данном параграфе рассматриваются следующие вопросы: условный оператор, оператор выбора (варианта), оператор перехода, компоненты ListBox, ComboBox, некоторые приемы работы с отладчиком.

Теория

Логические выражения

Логическое выражение в общем виде выглядит так:

ОП1 оператор ОП2,

Где ОП1 и ОП2 — операнды логического выражения, в качестве которых может выступать переменная, константа, функция или выражение; оператор — это оператор сравнения.

Результатом логического выражения является логическое значение True или False. Логические выражения чаще всего используются в условном операторе и в операторах цикла и состоят из:

- логических констант True и False ;
- операндов (переменные, элементы массивов, выражения)
- логических переменных типа boolean;
- операций сравнения (отношения);
- логических операций;
- круглых скобок.

Для установления отношения между двумя значениями, заданными выражениями, переменными или константами, используются следующие операции сравнения:

- = — равно,
- < — меньше,
- > — больше,
- <= — меньше или равно,
- >= — больше или равно,
- <> — не равно,
- in — принадлежность множеству.

Операции сравнения выполняются после вычисления соответствующих выражений. Результатом операции сравнения является значение False, если соответствующее отношение не имеет места, или значение True, если соответствующее отношение имеет место.

Замечание

Приоритет операций сравнения меньше, чем приоритет логических операций. Поэтому если содержащее операцию сравнения логическое выражение является операндом логической операции, то его нужно заключить в круглые скобки.

В сложных логических выражениях используются следующие логические операторы

Операция	Описание	Операнд 1	Операнд 2	Результат
not	Отрицание	False True		True False
and	Логическое и	False False True True	False True False True	False False False True
or	Логическое или	False False True True	False True False True	False True True True
xor	Исключающее или	False False True True	False True False True	False True True False

Пример. Логические выражения

$x < 10$

$x + 17 \geq y$

$(x > a) \text{ and } (x < b)$.

Переменные x , a , b и y могут принадлежать, например, к числовым или строковым типам.

Условный оператор

Условный оператор обеспечивает выполнение или невыполнение некоторых операторов в зависимости от соблюдения определенных условий. Условный оператор в общем случае предназначен для организации разветвления программы на два направления и имеет формат:

```
if <Условие> then <Оператор1> [ else <Оператор2> ];
```


Условие представляет собой выражение логического типа. Оператор работает следующим образом: если условие истинно (имеет значение True), то выполняется Оператор1, в другом случае выполняется Оператор2. Оба оператора могут быть составными. Условный оператор может быть записан в сокращенной форме, когда слово else и оператор после него отсутствуют. В этом случае при невыполнении условия выполняется оператор, следующий за условным.

Для организации разветвлений на три направления и более можно использовать несколько условных операторов, вложенных друг в друга. При этом каждое else соответствует тому then, которое непосредственно ему предшествует. Из-за возможных ошибок следует избегать большой вложенности условных операторов друг в друга.

Пример. Условные операторы:

```
if x > 0 then x := x + 1 else x := 0;
```

```
if q = 0 then a := 1;
```

Оператор выбора

Оператор выбора является обобщением условного оператора и позволяет сделать выбор из произвольного числа имеющихся вариантов, то есть организовать разветвления на произвольное число направлений. Этот оператор состоит из выражения, называемого *селектором*, списка вариантов и необязательной ветви else, имеющей тот же смысл, что и в условном операторе.

Формат оператора выбора:

```
case <Выражение-селектор> of
```

```
<Список1> : <Оператор1>;
```

```
<СписокN> : <ОператорN>
```

```
else <Оператор>;
```

```
end;
```

Выражение–селектор должно быть порядкового типа. Каждый вариант представляет собой список констант, отделенных двоеточием от относящегося к данному варианту оператора, возможно, составного. Список констант выбора состоит из произвольного количества значений и диапазонов, отделенных друг от друга запятыми. Границы диапазона записываются двумя константами через разделитель " . . ". Тип констант должен совпадать с типом выражения–селектора.

Оператор выбора выполняется следующим образом:

- вычисляется значение выражения–селектора;
- производится последовательный просмотр вариантов на предмет совпадения значения селектора с константами и значениями из диапазонов соответствующего списка;
- если для очередного варианта этот поиск успешный, то выполняется оператор этого варианта. После этого выполнение оператора выбора заканчивается;
- если все проверки оказались безуспешными, то выполняется оператор, стоящий после слова `else` (при его наличии).

Пример. Оператор выбора.

```
case DayNumber of
1 .. 5 : strDay := 'Рабочий день';
6, 7 : strDay := 'Выходной день'
else strDay := "";
end;
```

В зависимости от значения целочисленной переменной `DayNumber`, содержащей номер дня недели, присваивается соответствующее значение строковой переменной `strDay`.

Оператор перехода

Оператор безусловного перехода осуществляет переход на оператор, помеченный меткой. Формат оператора перехода следующий:

`Goto` метка, где метка — это идентификатор, находящийся перед оператором, который должен быть выполнен после оператора `Goto`.

Метка должна быть описана в разделе `label`, который располагается перед разделом описания переменных.

```
Например,
Label NNN, 123, G12;
```

```
.....
NNN: A:=D/S;
```

```
....
Goto NNN;
```

Замечание

Современный стиль программирования предполагает составление программ без использования оператора `Goto`.

Компонент ListBox



TListBox (список) нужен для показа прокручиваемого списка. Классический пример ListBox'a в среде Windows — выбор файла из списка в пункте меню File | Open многих приложений. Названия файлов или директорий находятся в ListBox.

Главные свойства компонента TListBox

Свойство	Назначение
Name	Имя компонента. В программе используется для доступа к свойствам компонента
Items	Элементы списка
ItemIndex	Номер выбранного элемента списка. Номер первого элемента списка равен нулю

Список может быть сформирован во время создания формы или во время работы программы.

Для формирования списка во время создания формы надо в окне Инспектора Объектов (Object Inspector) выбрать свойство Items и щелкнуть на кнопке запуска редактора списка строк. В открывшемся окне String List Editor нужно набрать список, каждый элемент списка в отдельной строке. После ввода очередного элемента списка, для перехода к новой строке, необходимо нажать клавишу «ENTER». После ввода последнего элемента списка щелкнуть на кнопке ОК.

Добавить элементы в список непосредственно в программе можно с помощью метода Add, например:

```
ListBox1.Items.Add('Россия');
ListBox1.Items.Add('Австрия');
```

Если ни один из элементов списка не выбран, то значение свойства ItemIndex равно -1.

Компонент TComboBox



TComboBox во многом напоминает TListBox, за исключением того, что позволяет вводить информацию в маленьком поле ввода/редактирования сверху либо путем выбора из списка. Некоторые свойства: Text (текст, находящийся в поле ввода/редактирования); Items (элементы раскрывающегося списка); DropDownCount (количество отображаемых элементов в раскрытом списке). Список может быть сформирован таким же образом, как у компонента ListBox.

Некоторые приемы работы с отладчиком

Отладчик позволяет выполнять программу построчно, по функциям или по блокам. Указав, какие части программы должны выполняться и когда выполнение программы должно приостановиться, можно быстро пропускать те участки программы, которые работают правильно, сосредоточившись на отладке тех частей, которые приводят к неверному выполнению программы. Для управления выполнением программы предназначены следующие команды.

Run	Программа выполняется без остановки на каждой строке. Эта программа эквивалентна выполнению программы вне отладчика
Step Over	Программа выполняется построчно. Подпрограммы выполняются как одна строка
Trace Into	Программа выполняется построчно. Также выполняется каждая строка подпрограммы
Run To Cursor	Программа выполняется до той строки, на которой находится курсор в окне Редактора Кода
Program Pause	Выполнение программы приостанавливается
Program Reset	Выполнение программы завершается
Add Breakpoint	Можно указать точки прерывания при выполнении программы
Add Watch	Можно указать точки просмотра для слежения за изменением значений переменных или полей структур данных

В процессе обучения программированию очень удобно пользоваться пошаговым выполнением программы. Почему? Это позволяет посмотреть, как изменяются значения переменных, участвующих в программе, в процессе пошагового выполнения. Какую последовательность действий предпринять? Установить курсор в тексте программы на вторую исполняемую строку после Begin и запустить команду Run To Cursor (F4). Программа выполнит этот оператор и остановится. В окне Редактора Кода выполненная строка будет помечена зеленой стрелкой слева и выделена синим цветом. Если навести курсор на название какой-нибудь переменной из предшествующей части программы, рядом с курсором появится справочное окошко, где приведено значение

этой переменной. Далее запустить командой Step Over (F8) следующий шаг выполнения. Будет помечена следующая строка при выполнении. С помощью курсора мыши можно посмотреть, как изменились значения переменных при выполнении этого шага программы.

Часто при отладке пользуются окном точек просмотра. Пометить переменные для просмотра можно, наведя на них курсор и выполнив команду Add Watch (Ctrl+F5). Далее следует запустить пошаговое выполнение программы. Часто окно Watch List бывает закрыто окном редактора кода. Его надо отодвинуть.

Отладочные настройки отменяются командой Program Reset.

Практика

Задача

Составить программу, определяющую по росту и весу пользователя программы, похудеть или поправиться ему нужно и на сколько килограммов.

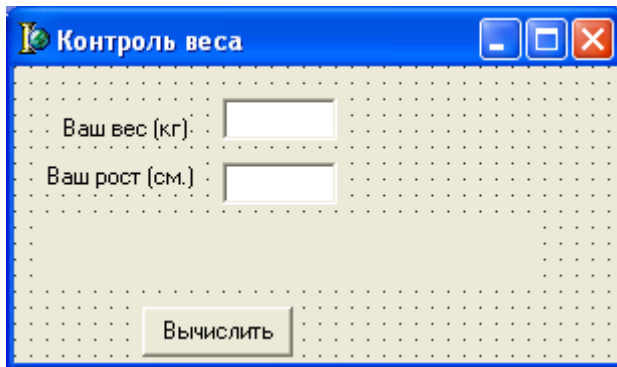
Компоненты

Имя компонента	Свойства компонентов	Значение	Назначение
Form1	Caption	Контроль веса	Заголовок формы
Label1	Caption	Поле для ввода веса	Подсказка пользователю
Label2	Caption	Поле для ввода роста	Подсказка пользователю
Label3	Caption		Поле вывода результатов работы программы
Button1	Caption	Вычислить	Кнопка для вычисления

Переменные

Обозначение в программе	Содержание	Тип
wes	Вес	Вещественный
Rost	Рост	Вещественный
opt	Оптимальный вес	Вещественный
d	Отклонение от оптимального веса	Вещественный

Проект формы



unit wtest;

.....

var

Form1: TForm1;

implementation

*{ \$R *.DFM }*

procedure TForm1.Button1Click(Sender: TObject);

var

wes:real; {вес}

Rost:real; {рост}

opt:real; {ОПТИМАЛЬНЫЙ вес}

d:real; {отклонение от оптимального веса}

begin

wes:=StrToFloat(Edit1.text);

Rost:=StrToFloat(Edit2.Text);

opt:=Rost-100;

if wes=opt

then

Label3.caption:='Ваш вес оптимален!'

else

if wes<opt

then

begin

d:=opt-wes;

Label3.caption:='Вам надо поправиться, на'+ FloatToStr(d)+

'кг.';

end

else

begin

d:=wes-opt;

Label3.caption:='Надо немного похудеть, на '+ FloatToStr(d)+ 'кг.';

end;

end;

end.

Результаты работы программы

Контроль веса

Ваш вес (кг)

Ваш рост (см.)

Вам надо поправиться, на 11 кг.

Задача

Некоторая страховая компания страхует индивидуальные дома по следующей схеме:

Стоимость страховки = Площадь * Цена_кв.м * Коэф.этажности * *Износ/100*Проценты*Курс_у.е..

Цена_кв.м:

Кирпичный коттедж — 200 у.е.

Дом из оцилиндрованного бревна — 250 у.е.

Дом из бруса — 150 у.е.

Дом из бревен — 100 у.е.

Каркасно-щитовой — 50 у.е.

Коэффициент этажности:

1 этаж — 1

2 этажа — 1,2

3 этаж — 1,3

Коэффициент износа:

Возраст дома:

Более 15 лет — 0,8

От 11 до 15 лет — 0,6

от 6 до 10 лет — 0,4

до 5 лет — 1

Проценты страхового сбора:

Количество лет страхования в данной страховой компании

Превышает 7 лет — 1 процент

От 3 до 7 лет — 2 процента

До 3 лет (включительно) — 3 процента

Компоненты

Имя компонента	Свойства компонентов	Значение	Назначение
Form1	Caption	Сумма страховки индиви-	Заголовок формы

		дуального дома	
Label1	Caption	Программа для расчета суммы страхования индивидуального дома в некоторой страховой компании	Справочная информация для пользователя программы
Label2	Caption	Общая площадь (кв.м)	Подсказка пользователю
Label3	Caption	Год постройки	Подсказка пользователю
Label4	Caption	Сколько лет дом страховался в данной страховой компании	Подсказка пользователю
Label5	Caption		Поле для вывода суммы страховки дома
Label6	Caption		Поле для вывода текущей даты, установленной на компьютере
Label7	Caption	Курс у.е.	Подсказка пользователю
Edit1	Text	Поле для ввода общей площади
Edit2	Text	Поле для ввода года постройки дома
Edit3	Text	Поле для ввода количества лет страхования в данной компании
Edit4	Text		Поле для ввода курса у.е.
Button1	Caption	РАСЧЕТ	Кнопка для вычисления суммы страховки
Button2	Caption	ВЫХОД	Кнопка закрытия формы и выхода из программы
ListBox1	Items	Кирпичный коттедж Дом из оцилиндрованного бревна Дом из бруса Дом из бревен Каркасно-щитовой	Список для выбора типа дома
ComboBox1	Text Items	Количество этажей 1 2 3	Поле со списком

Переменные

Обозначение в программе	Содержание	Тип
S	Общая площадь индивидуального дома	Вещественный
Zena_m	Цена кв. метра	Денежный
Summa	Сумма страхового взноса	Денежный
d	Текущий день	Целый беззнаковый
m	Текущий месяц	Целый беззнаковый
Y	Текущий год	Целый беззнаковый
God	Год постройки дома	Целый беззнаковый
KE	Кoeffициент этажности	Вещественный
Iznos	Кoeffициент износа	Вещественный
Prozent	Процент страхового взноса	Вещественный
Kurs	Курс у.е.	Вещественный

Проект формы

Сумма страховки индивидуального дома

Программа для расчета суммы страхования индивидуального дома в некоторой страховой компании

Кирпичный коттедж
Дом из оцилиндрованного бревна
Дом из бруса
Дом из бревен
Каркасно-щитовой

Количество этажей

РАСЧЕТ

ВЫХОД

Общая площадь (кв.м)

Год постройки

Курс у.е.

Сколько лет дом страховался в данной страховой компании

Текст модуля

unit Unit1;

.....

var

Form1: TForm1;

implementation

*{ \$R *.dfm }*

procedure TForm1.Button1Click(Sender: TObject);

Var S:Real; //общая площадь дома

Zena_m, Summa: currency; // цена кв.м, сумма страховки дома

```

d,m,Y,God: Word; // день, месяц, год страхования; год
//постройки дома
KE:Real; // коэффициент, учитывающий этажность
Iznos:Real;// коэффициент износа
Prozent:Integer; // процент страхового взноса
Kurs:Real; // курс у.е.
begin
// ввод площади дома, года постройки и курса у.е.
S:=StrToFloat(Edit1.Text);
God:=StrToInt(Edit2.Text);
Kurs:=StrToFloat(Edit4.Text);
// вывод текущей даты на форму
Label6.Caption:='Сегодня : '+DateToStr(Date);
// по выбранному из списка типу дома определяется цена за
//кв.м
Case ListBox1.ItemIndex of
0:Zena_m:=200;
1:Zena_m:=250;
2:Zena_m:=150;
3:Zena_m:=100;
4:Zena_m:=50;
Else Exit;
end;
// определение коэффициента этажности
// по выбранному из поля со списком числу этажей
Case ComboBox1.ItemIndex of
0:KE:=1;
1:KE:=1.2;
2:KE:=1.3;
Else Exit;
end;
// разделение текущей даты на составляющие
// и помещение их в отдельные переменные: год, месяц, день
Decodedate(Date,Y,m,d);
// если разность между текущим годом и годом постройки дома
// не больше 5 лет, то износ равен 1
If Y-God<=5 Then Iznos:=1
// иначе если разность между текущим годом и годом постройки // дома
// больше 5 лет и не более 10, то износ равен 0,8
Else If ((Y-God)>5) and ((Y-God)<=10)
Then Iznos:=0.8
// иначе если разность между текущим годом и годом
// постройки дома

```

```

// не больше 15 лет, то износ равен 0,6
      Else If Y-God<=15 Then Iznos:=0.6
            Else
                  Iznos:=0.4;// иначе износ 0,4
// если дом застрахован в данной компании не более 3 лет,
// то процент страхового взноса равен 3
If StrToInt(Edit3.Text)<=3 Then Prozent:=3
// иначе дом застрахован в данной компании более 7 лет,
// то процент страхового взноса равен 1
      Else If StrToInt(Edit3.Text)>7 Then Prozent:=1
            // иначе дом страхуется от 3 до 7 лет,
            // процент страхового взноса равен 2
            Else Prozent:=2;
// расчет суммы страхового взноса
Summa:=S*Zena_m*KE*Iznos/100*Prozent*Kurs;
// вывод суммы страховки на форму
Label5.Caption:='Сумма страховки Вашего дома составляет: '
                +FloatToStrF(Summa,Ffixed,10,2)+' руб.'
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
Form1.Close;
end;
end.

```

Результаты работы программы

Сумма страховки индивидуального дома

Программа для расчета суммы страхования индивидуального дома в некоторой страховой компании

Сколько лет дом страховался в данной страховой компании

Сумма страховки Вашего дома составляет: 530,51 руб. Сегодня : 05.04.2007

Задачи для самостоятельного решения

Группа А

№ 3.1. Дано действительное число x . Вычислить $f(x)$, если

$$f(x) = \begin{cases} x^2 & \text{при } -2 \leq x \leq 2, \\ 4 & \text{в противном случае} \end{cases}$$

№ 3.2. Даны три действительных числа. Выбрать из них те, которые принадлежат интервалу $(1;3)$.

№ 3.3. Даны действительные числа x, y ($x \neq y$). Меньшее из этих двух чисел заменить их полусуммой, а большее — их удвоенным произведением.

№ 3.4. Даны три действительные числа. Возвести в квадрат те из них, значения которых неотрицательны.

№ 3.5. Дано действительное число a . Вычислить $f(x)$, если

$$f(x) = \begin{cases} x^2 + 4x + 5 & \text{при } x \leq 2, \\ 1/(x^2 + 4x + 5) & \text{в остальных случаях} \end{cases}$$

№ 3.6. Дано действительное число a . Вычислить $f(x)$, если

$$f(x) = \begin{cases} 0, & \text{при } x \leq 0, \\ x & \text{при } 0 \leq x \leq 1, \\ x^4 & \text{в остальных случаях.} \end{cases}$$

№ 3.7. Дано действительное число a . Вычислить $f(x)$, если

$$f(x) = \begin{cases} 0, & \text{при } x \leq 0 \\ x^2 - x & \text{при } 0 < x \leq 1 \\ x^2 - \sin \pi x^2 & \text{в остальных случаях.} \end{cases},$$

№ 3.8. Даны действительные числа a, b, c . Проверить, выполняются ли неравенства $a < b < c$.

№ 3.9. Даны действительные положительные числа: a, b, c, d . Выяснить, можно ли прямоугольник со сторонами a, b уместить внутри прямоугольника со сторонами c, d так, чтобы каждая из сторон одного прямоугольника была параллельна или перпендикулярна каждой стороне второго прямоугольника.

№ 3.10. Даны действительные положительные числа a, b, c, x, y . Выяснить, пройдет ли кирпич с ребрами a, b, c в прямоугольное отверстие со сторонами x и y . Просовывать кирпич в отверстие разрешается только так, чтобы каждое из его ребер было параллельно или перпендикулярно каждой из сторон отверстия.

Группа Б

№ 3.11. Написать программу для определения площади треугольного участка по заданным значениям длин его сторон A, B и C . Для вычисления площади использовать формулу Герона

$$S = \sqrt{P(P - A)(P - B)(P - C)},$$

где $P = \frac{A + B + C}{2}$ — полупериметр.

В программе предусмотреть проверку существования треугольника со сторонами A, B и C . Треугольник со сторонами A, B и C существует лишь в том случае, если одновременно выполняются неравенства $A + B > C, A + C > B, B + C > A$. В случае, если треугольник с заданными значениями сторон не существует, выдавать сообщение: «ОШИБКА, ПРОВЕРЬТЕ ИСХОДНЫЕ ДАННЫЕ» и передавать управление в начало программы. Длины

сторон вводить в метрах, площадь определять в гектарах с точностью до 1 га.

№ 3.12. Из вершины О трапециевидного участка (рис. 7–3) измерены расстояния А, С и В до трех остальных вершин и угол между сторонами А и В. Написать программу для вычисления площади этого участка по формуле:

$$S = \frac{B + \sqrt{C^2 * \sin^2 \alpha - A * \cos \alpha}}{2} * A * \sin \alpha .$$

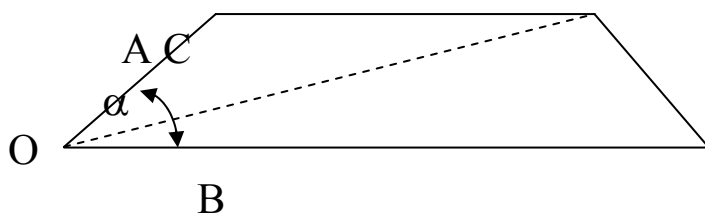


Рис. 6.3.

Расстояния А, В и С измерены в метрах, угол α в градусах, минутах и секундах. Площадь вычислить в гектарах, с точностью до 0,001 га.

Форма вывода результата:

Площадь = значение S в га

В программе предусмотреть следующие запросы при вводе:

БОКОВАЯ СТОРОНА =

ОСНОВАНИЕ =

ДИАГОНАЛЬ =

УГОЛ =

В программе предусмотреть также проверку существования трапеции с введенными параметрами. Случаи, в которых трапецию построить нельзя:

– угол $\alpha \geq 180^\circ$,

$C < A * \sin \alpha$,

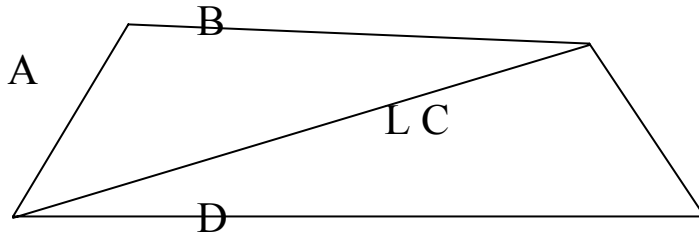
- $0 < \alpha \leq 90^\circ$ и $C < A$.

Во всех указанных ситуациях выводить сообщение «ТРАПЕЦИИ НЕ СУЩЕСТВУЕТ», и передавать управление оператору ввода исходных данных.

№ 3.13. Написать программу для определения площади четырехугольника по значениям его четырех сторон А, В, С, D и диагонали (рис. 7–4) по формуле:

$$S = \sqrt{P_1(P_1 - A)(P_1 - B)(P_1 - L)} + \sqrt{P_2(P_2 - C)(P_2 - D)(P_2 - L)},$$

где $P_1 = \frac{A + B + L}{2}$, $P_2 = \frac{C + D + L}{2}$.



3.30 Рис. 6.4.

Значения А, В, С, D, L вводить в метрах, площадь округлять до сотых гектара.

В программе предусмотреть проверку условия $(A + B > L) \wedge (A + L > B) \wedge (B + L > A) \wedge (C + D > L) \wedge (C + L > D) \wedge (D + L > C)$. В случае невыполнения этого условия выводить сообщение «ОШИБКА В ИСХОДНЫХ ДАННЫХ» и передавать управление оператору ввода исходных данных.

В программе предусмотреть следующие запросы при вводе исходных данных:

СТОРОНА СЛЕВА ОТ ДИАГОНАЛИ =
 СТОРОНА СПРАВА ОТ ДИАГОНАЛИ =
 ДИАГОНАЛЬ =

Форма вывода результата:

ПЛОЩАДЬ = значение S ГА.

№ 3.14. Написать программу, которая определяет величину уклона на участке площадью Р га по заданному сечению горизонталей Н и длине всех горизонталей С:

$$I = \frac{H * C}{175P}.$$

Если вычисленное значение $I \leq 3$, выводить сообщение:

«ИЗМЕРЕНИЕ ПЛОЩАДЕЙ ПО ФОТОСХЕМЕ
 ДОПУСТИМО: $I \leq 3$ ГР.»

В противном случае выводить сообщение:

«ИЗМЕРЕНИЕ ПЛОЩАДЕЙ ПО ФОТОСХЕМЕ
 НЕДОПУСТИМО: $I > 3$ ГР.»

Форма запросов при вводе исходных данных:

ПЛОЩАДЬ =
СЕЧЕНИЕ ГОРИЗОНТАЛЕЙ =
ДЛИНА ГОРИЗОНТАЛЕЙ =

Примечание

Уклон линии — отношение превышения h между точками к горизонтальному проложению S между ними.

Превышение — разность высот точек.

Горизонтальное проложение — ортогональная проекция линии местности на горизонтальную плоскость.

Высоты горизонталей всегда кратны высоте сечения рельефа. *Высота сечения рельефа* — это расстояние, на которое отстоят друг от друга уровенные поверхности, пересекающие земную поверхность.

№ 3.14. Написать программу для вычисления предполагаемого валового сбора зерна по формулам:

$$Y = \begin{cases} 218,36 + 73,14x_1 + 39,56x_2 + 56,84x_3 & \text{для } R < 56,34 \\ 431,45 + 132,6x_1 + 84,51x_2 + 112,59x_3 & \text{для } R \geq 56,34 \end{cases}$$

где x_1, x_2, x_3 — площади почвенных групп (в га),

R — текущая норма осадков (в мм).

Форма запросов при вводе исходных данных и сообщений при выводе результатов.

ПЛОЩАДИ ПОЧВЕННЫХ ГРУПП:

X1=

X2=

X3=

ТЕКУЩАЯ НОРМА ОСАДКОВ=

ПРЕДПОЛАГАЕМЫЙ ВАЛОВЫЙ СБОР ЗЕРНА=значение

Y ц.

№ 3.15. Написать программу для определения положения точки с координатами $X1, Y1$ по отношению к полосе шириной P , ось которой описывается уравнением $AX + BY + C = 0$.

Расстояние от точки с координатами $X1, Y1$ до прямой можно вычислить по формуле

$$R = \left| \frac{AX_1 + BY_1 + C}{\sqrt{A^2 + B^2}} \right|.$$

Условием нахождения точки в указанной полосе является неравенство $R \leq P/2$.

В программе предусмотреть запросы на ввод исходных данных:

A, B, C=

P=

X1, Y1=

Результат выводить в виде одного из сообщений:

ТОЧКА В ПРЕДЕЛАХ ПОЛОСЫ или

ТОЧКА ЗА ПРЕДЕЛАМИ ПОЛОСЫ

№ 3.16. Написать программу для определения положения точки с координатами X1, Y1 по отношению к круглому участку радиуса R с центром в точке с координатами X0, Y0.

Пояснение: признаком нахождения точки внутри или на границе участка является выполнение неравенства

$$\sqrt{\Delta X^2 + \Delta Y^2} \leq R,$$

где $\Delta X = X1 - X0$, $\Delta Y = Y1 - Y0$.

В программе предусмотреть выдачу запросов на ввод исходных данных:

X0, Y0=

R=

X1, Y1=

Результаты выводить в виде одного из сообщений:

ТОЧКА В ПРЕДЕЛАХ УЧАСТКА или

ТОЧКА ЗА ПРЕДЕЛАМИ УЧАСТКА

№ 3.17. Написать программу для определения положения точки с координатами X1, Y1, Z1 по отношению к сфере с радиусом R и координатами центра X0, Y0, Z0. Признаком нахождения точки внутри сферы или на ее границе является выполнение неравенства

$$\sqrt{\Delta X^2 + \Delta Y^2 + \Delta Z^2} \leq R,$$

где

$$\Delta X = X_1 - X_0, \Delta Y = Y_1 - Y_0, \Delta Z = Z_1 - Z_0.$$

В программе предусмотреть выдачу запросов на ввод исходных данных:

X0, Y0, Z0=

R=

X1, Y1, Z1=

Результаты выводить в виде одного из сообщений:

ТОЧКА ЗА ПРЕДЕЛАМИ СФЕРЫ или

ТОЧКА В ПРЕДЕЛАХ СФЕРЫ

№ 3.18. Написать программу для определения положения точки с координатами X1, Y1 по отношению к прямой, описываемой уравнением $AX + BY + C = 0$. Признаком того, что точка лежит на прямой, будем считать выполнение неравенства :

$$|AX_1 + BY_1 + C| < 0,001.$$

Форма запросов при вводе исходных данных:

A, B, C=

X1, Y1=

Форма вывода результата:

ТОЧКА НА ПРЯМОЙ или

ТОЧКА ВНЕ ПРЯМОЙ,

в зависимости от выполнения или невыполнения заданного условия.

№ 3.19. Написать программу, которая определяет положение прямой, заданной уравнением $AX + BY + C = 0$, относительно прямоугольной системы координат. Возможные ситуации:

A=0, B=0, C — любое — ошибка при вводе исходных данных, прямой не существует,

A=0, B≠0, C=0 — прямая совпадает с осью X,

A≠0, B=0, C=0 — прямая совпадает с осью Y,

A=0, B≠0, C≠0 — прямая проходит через точку $Y = -C/B$

параллельно оси X,

A≠0, B=0, C≠0 — прямая проходит через точку $Y = -C/A$

параллельно оси Y,

A≠0, B≠0, C=0 — прямая проходит через начало координат и точку с произвольным значением X (предусмотреть в случае возникновения указанной ситуации ввод значения X) и $Y = -AX/B$.

$A \neq 0, B \neq 0, C \neq 0$ — прямая пересекает ось X в точке $X = -C/A$ и ось Y в точке $Y = -C/B$.

Запросы при вводе исходных данных:

ПАРАМЕТРЫ $A, B, C =$

Форма вывода результатов:

СОВПАДАЕТ С ОСЬЮ X

или СОВПАДАЕТ С ОСЬЮ Y

и т. п.

В случае $A \neq 0, B \neq 0, C = 0$ вначале выводить запрос: «ВВЕДИТЕ ЛЮБОЕ ЗНАЧЕНИЕ X », а затем сообщение:

«ПРОХОДИТ ЧЕРЕЗ НАЧАЛО КООРДИНАТ И ТОЧКУ

$X =$ значение, $Y =$ значение».

№ 3.20. Написать программу, которая определяет, лежит ли точка с координатами X, Y на прямой, проведенной через точки с координатами X_1, Y_1 и X_2, Y_2 . Признаком, того что точка лежит на прямой, будем считать выполнение неравенства.

$$\left| \frac{Y - Y_1}{Y_2 - Y_1} - \frac{X - X_1}{X_2 - X_1} \right| < 0,01.$$

Форма вывода результата:

ТОЧКА НА ПРЯМОЙ, если точка лежит на прямой, и

ТОЧКА ВНЕ ПРЯМОЙ, если точка лежит не на прямой.

Форма запроса при вводе исходных данных:

КООРДИНАТЫ $X_1, Y_1 =$

КООРДИНАТЫ $X_2, Y_2 =$

КООРДИНАТЫ $X, Y =$

§4. Циклы

В данном параграфе рассматриваются следующие вопросы: операторы цикла (For do, While do, Repeat Until), компонент Метод.

Теория

Операторы цикла

Операторы цикла используются для организации циклов (повторов). *Цикл* представляет собой последовательность операторов, которая может выполняться более одного раза. Группу по-

вторяемых операторов называют телом цикла. Всего имеется три вида операторов цикла:

- с параметром;
- с предусловием;
- с постусловием.

Обычно, если количество повторов известно заранее, применяется оператор цикла с параметром, в противном случае — операторы с пост или предусловием (чаще используется оператор с предусловием).

Операторы циклов могут быть вложенными друг в друга.

Оператор цикла с параметром

Оператор цикла с параметром имеет два следующих формата:

for <Параметр> := <Выражение1> to <Выражение2> do <Оператор>;

и

for <Параметр> := <Выражение1> downto <Выражение2> do <Оператор>;

Параметр цикла представляет собой переменную порядкового типа, которая должна быть определена в том же блоке, где находится оператор цикла, выражение1 и выражение2 являются соответственно начальным и конечным значениями параметра цикла и должны иметь тип, совместимый с типом параметра цикла. Оператор цикла обеспечивает выполнение тела цикла, которым является оператор после слова do, до полного перебора всех значений параметра цикла от начального до конечного с соответствующим шагом. Шаг параметра всегда равен 1 для первого формата цикла и -1 — для второго формата. То есть значение параметра последовательно увеличивается (for... to) или уменьшается (for.. .downto) на единицу при каждом повторении цикла. Цикл может не выполниться ни разу, если для цикла for . . . to значение начального выражения больше конечного, а для цикла for.. .downto, наоборот, значение начального выражения меньше конечного.

Пример. Циклы с параметром.

```
var n, k: integer;  
    s:Real;  
    f:Extended;  
begin  
  {цикл вычисления суммы ряда  $1+1/2+1/3+\dots+1/n$ ,  $n=1,\dots, 10$ }  
  s := 0;
```

```

for n := 1 to 10 do
s := s + 1/n;
f :=1;
{ цикл вычисления произведения членов ряда 1*1/2*1/3*...*1/k,
k=1,..., 20}
for k := 1 to 20 do
begin
f :=f*1/k
end;

```

Оператор цикла с предусловием

Оператор цикла с предусловием целесообразно использовать в случаях, когда число повторений тела цикла заранее неизвестно. Тело цикла может не выполняться, если условие цикла является ложью.

Формат оператора цикла с предусловием:

```
while <Условие> do <Оператор>;
```

Оператор тела цикла выполняется до тех пор, пока логическое выражение не примет значение False, то есть в отличие от цикла с постусловием, цикл выполняется при значении логического выражения True.

Пример. Оператор цикла с предусловием.

Рассмотрим расчет суммы натуральных чисел от 1 до n с шагом 3

```

var i,n: integer;
sum: integer;
begin
sum := 0;
i:=1;
while i <= n do
begin
sum := sum + i;
i := i + 3;
end;
end;

```

Если перед первым выполнением цикла условие не выполняется (значение логического выражения равно False), то тело цикла не выполняется ни разу и происходит переход на следующий за оператором цикла оператор.

Оператор цикла с постусловием

Во многом этот оператор аналогичен оператору цикла с предусловием, но проверка условия выполняется в конце оператора. Выход из цикла происходит при равенстве условия константе True (Истина).

Формат оператора цикла с постусловием:

repeat <Оператор>; until <Условие>;

Пример.

Найдем сумму $\sum_{n=1}^{10} \frac{n}{(x-n)}$ с шагом 2, при $x-n \neq 0$.

n:=1;

Sum :=0;

Repeat

Sum := Sum + n/(x-n);

n:=n+2;

Until (n>10) or (x-n=0);

В операторе Repeat Until условие проверяется после выполнения операторов тела цикла. Следовательно, операторы цикла в любом случае выполнятся хотя бы один раз. В этом примере, если в программе заранее не определено, что $x \neq 1$, при попадании в цикл произойдет ошибка выполнения (деление на ноль).

Компонент Мемо



ТМемо — иная форма компонента TEdit. Мемо находится на вкладке Standard. Подразумевает работу с большими текстами и может выполнять основные функции редактора (ввод и редактирование текста). Мемо имеет ограничения на объем текста в 32Кб, это составляет 10-20 страниц.

Главные свойства компонента Мемо

Свойство	Назначение
Name	Имя компонента
Text	Текст, находящийся в поле Мемо. Рассматривается как единое целое
Lines	Текст, находящийся в поле Мемо. Рассматривается как совокупность строк. Доступ к строке осуществляется по номеру

Lines.Count	Количество строк в поле Мемо
-------------	------------------------------

Например, присвоить переменной ST весь текст, находящийся в компоненте Мемо можно так:

```
ST:=Мемо1.Text;
```

Добавить строку:

```
Мемо1.Lines.Add('Еще одна строка');
```

Переменной можно присвоить содержимое одной строки:

```
Stroke:=Мемо1.Lines[i];
```

Практика

Задача

Составить программу для вычисления таблиц значений квадратов, кубов, квадратных и кубических корней, πn , $\pi/(4n^2)$, $1/n$.

Компоненты

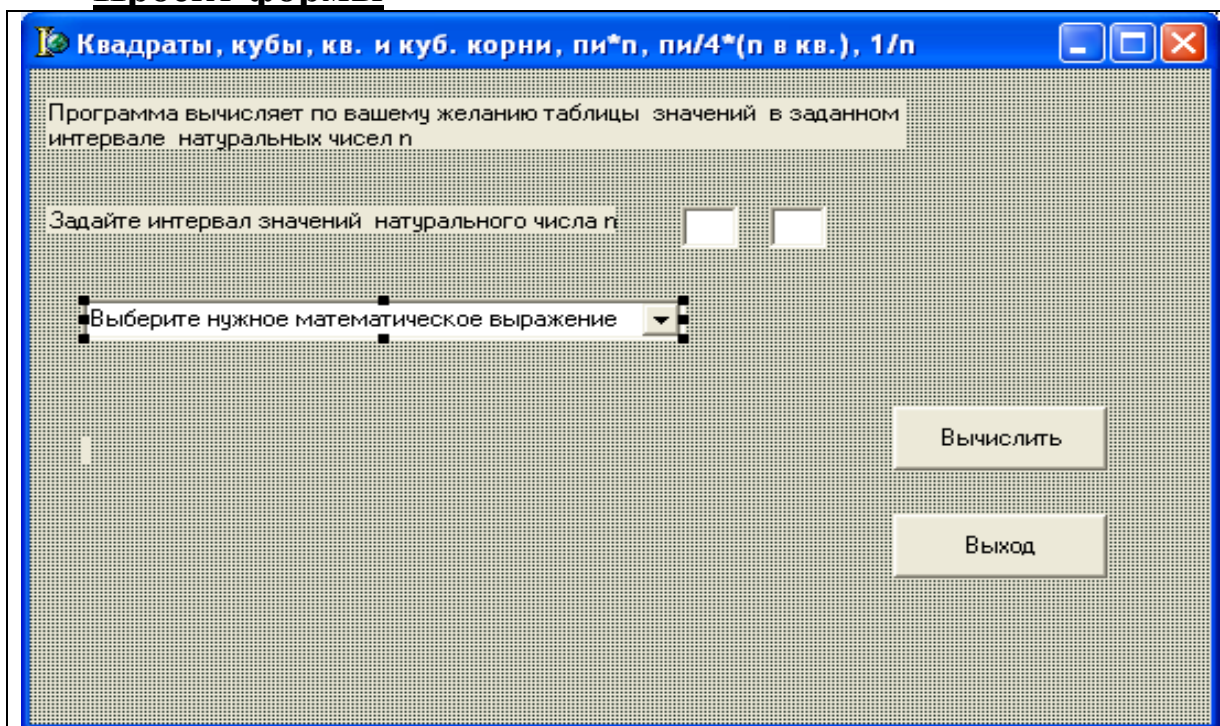
Имя компонента	Свойства компонента	Значение	Назначение
Form1	Caption	Квадраты, кубы, кв. и куб. корни, $\pi*n$, $\pi/4*(n \text{ в кв.})$, $1/n$	Заголовок формы
Label1	Caption	Программа вычисляет по вашему желанию таблицы значений в заданном интервале натуральных чисел n	Справочная информация для пользователя программы
Label2	Caption	Задайте интервал значений натурального числа n	Подсказка пользователю
Label3	Caption		Поле для вывода результатов
Edit1	Text		Поле для ввода начального значения n
Edit2	Text		Поле для ввода конечного значения n
ComboBox1	Text Items	Выберите нужное математическое выражение Квадраты Кубы Квадратные корни Кубические корни $\pi*n$ $\pi/[4*(n \text{ в кв.})]$ $1/n$	Текст в заголовке списка Значения строк в списке
Button1	Caption	Вычислить	Кнопка для вычисления
Button2	Caption	Выход	Кнопка закрытия

		формы и выхода из программы
--	--	-----------------------------

Переменные

Обозначение в программе	Содержание	Тип
n	Текущее значение числа	Целый
n1	Начальное значение интервала натуральных чисел	Целый
n2	Конечное значение интервала натуральных чисел	Целый

Проект формы



Текст модуля

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms, Dialogs, StdCtrls, Math;
// обязательно к стандартному списку подключить библиотеку
// математических функций Math
.....
var
  Form1: TForm1;
  n1,n2:Integer; {начальное и конечное значения интервала
  натуральных чисел}

```



```

    n:integer; // текущее значение числа
implementation
{$R *.dfm}
procedure TForm1.Button2Click(Sender: TObject);
begin
// процедура закрывает форму и осуществляет
// выход из программы
Form1.Close
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
// ввод начального и конечного значений интервала
// натуральных чисел
n1:=StrToInt(Edit1.Text);
n2:=StrToInt(Edit2.Text);
Case ComboBox1.ItemIndex of
0: begin
// очистка поля ввода от предыдущих значений
label3.Caption:=' ';
// если в поле со списком выбрано значение Квадраты
// в цикле от n1 до n2 с шагом 1 вычисляются значения
// квадратов чисел
    For n:=n1 To n2 Do
        Begin
            Label3.Caption:= Label3.Caption+ #13+
                IntToStr(n)+' '+IntToStr(Sqr(n));
        end;
end;

1:begin
// очистка поля ввода от предыдущих значений
label3.Caption:=' ';
// если в поле со списком выбрано значение Кубы
// в цикле от n1 до n2 с шагом 1 вычисляются значения
// кубов чисел
    For n:=n1 To n2 Do
        Begin
            Label3.Caption:= Label3.Caption+ #13+
                IntToStr(n)+ ' ' + IntToStr(n*n*n);
        end;
end;

2:begin label3.Caption:=' ';
// если в поле со списком выбрано значение Квадратные корни

```

```

// в цикле от n1 до n2 с шагом 1 вычисляются значения
// квадратных корней чисел
  For n:=n1 To n2 Do
    Begin
      Label3.Caption:= Label3.Caption+ #13+
      IntToStr(n)+' ' + FloatToStrF(Sqrt(n),Ffixed,10,6);
    end;
end;
3:begin label3.Caption:=' ';
// функция логарифм не определена при нуле,
// поэтому сдвигаем интервал значений на единицу
  If n1=0 Then begin label3.Caption:='0';n1:=n1+1; end;
// если в поле со списком выбрано значение Кубические корни
// в цикле от n1 до n2 с шагом 1 вычисляются значения
// кубических корней чисел
  For n:=n1 To n2 Do
    Begin
      Label3.Caption:= Label3.Caption+ #13+
      IntToStr(n)+' ' + FloatToStrF(exp(1/3*ln(n)),Ffixed,10,6);
    end;
end;
4:begin label3.Caption:=' ';
  For n:=n1 To n2 Do
    Begin
// если в поле со списком выбрано значение пи*n
// в цикле от n1 до n2 с шагом 1 вычисляются значения
// пи*n чисел
      Label3.Caption:= Label3.Caption+ #13+
      IntToStr(n)+' '+ FloatToStrF((pi*n),Ffixed,10,6);
    end;
end;
5:begin label3.Caption:=' ';
// если в поле со списком выбрано значение пи/[4*(n в кв.)]
// в цикле от n1 до n2 с шагом 1 вычисляются значения
// пи/[4*(n в кв.)]
If n1=0 Then begin
      Label3.Caption:='Бесконечность'; n1:=n1+1
    End;
For n:=n1 To n2 Do
  Begin
    Label3.Caption:= Label3.Caption+ #13+
    IntToStr(n)+' '+ FloatToStrF((pi/(4*n*n)),Ffixed,10,6);
  end;
end;

```

```

end;
6:begin label3.Caption:=' ';
      If n1=0 Then begin
          Label3.Caption:=' Бесконечность'; n1:=n1+1
          end;
// если в поле со списком выбрано значение 1/n
// в цикле от n1 до n2 с шагом 1 вычисляются значения 1/n
      For n:=n1 To n2 Do
          Begin
              Label3.Caption:= Label3.Caption+ #13+
              IntToStr(n)+' '+ FloatToStrF(1/n,Fffixed,10,6);
          end;
      end;
Else Exit;
end;
end;
end.

```

Задача

Составить программу для вычисления таблицы уклонов местности. Уклоны местности определяют при проектировании каналов, дорог, направления движения тракторных агрегатов на пашне, изучении эрозии почв. Уклоном i называют отношение превышения h между точками к горизонтальному проложению S между ними. Для вычислений воспользоваться следующими формулами:

$$i = h/s,$$

$$i^{\circ} = \text{Arc tan}(i) / \pi * 180,$$

$$i \text{ в гр.} = \text{целая часть}(i^{\circ}),$$

$$i \text{ в мин.} = \text{округление} ((i^{\circ} - i \text{ в гр.}) * 60).$$

Компоненты

Имя компонента	Свойства компонент	Значение	Назначение
Form1	Caption	Таблица уклонов	Заголовок формы
Label1	Caption	При проектировании каналов, дорог, направления движения тракторных агрегатов на пашне, изучении эрозии почв определяют уклоны местности. Уклоном i называют отношение пре-	Справочная информация для пользователя программы

		вышения h между точками к горизонтальному проложению S между ними.	
Label2	Caption	Интервал изменения превышений	Подсказка пользователю
Edit1	Text		Нижняя граница интервала превышений
Edit2	Text		Верхняя граница интервала превышений
Label3	Caption	Шаг изменения превышений	Подсказка пользователю
Edit3	Text		Поле для ввода шага изменения превышений
Label4	Caption	Интервал изменения горизонтальных проложений	Подсказка пользователю
Edit4	Text		Нижняя граница интервала горизонтальных проложений
Edit5	Text		Верхняя граница интервала горизонтальных проложений
Label5	Caption	Шаг изменения горизонтальных проложений	Подсказка пользователю
Edit6	Text		Поле для ввода шага изменения горизонтальных проложений
Label6	Caption		В это поле программа выводит шапку таблицы уклонов
Memo1	Lines ScrollBars ssBoth	Компонент для вывода значений таблицы уклонов Выводить обе полосы прокрутки (вертикальную и горизонтальную)
Button1	Caption	ВЫЧИСЛИТЬ	Кнопка для выполнения
Button2	Caption	ВЫХОД	Кнопка для завершения работы программы

Переменные

Обозначение в программе	Содержание	Тип
h	Текущее значение превышения	Вещественный расширенный
h1	Нижняя граница интервала превышений	Вещественный расширенный
h2	Верхняя граница интервала превышений	Вещественный расширенный
s	Текущее значение горизонтального проложения	Вещественный расширенный
s1	Нижняя граница интервала горизонтальных проложений	Вещественный расширенный
s2	Верхняя граница интервала горизонтальных проложений	Вещественный расширенный
Dh	Шаг изменения превышений	Вещественный расширенный
Ds	Шаг изменения горизонтальных проложений	Вещественный расширенный
i	Текущее значение уклона в тангенсах	Вещественный расширенный
ig	Градусная часть уклона	Целый
im	Минутная часть уклона	Целый
Ari	Текущее значение уклона в градусах	Вещественный расширенный

Проект формы

Таблица уклонов

При проектировании каналов, дорог, направления движения тракторных агрегатов на пашне, изучении эрозии почв определяют уклоны местности. Уклоном i называют отношение превышения h между точками к горизонтальному проложению S между ними.

Интервал изменения превышений:

Шаг изменения превышений:

Интервал изменения горизонтальных проложений:

Шаг изменения горизонтальных проложений:

ВЫЧИСЛИТЬ

ВЫХОД

Текст модуля

```
unit Unit1;  
interface  
uses  
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, Math;  
// библиотека математических функций Math  
// подключается программистом  
.....  
var  
Form1: TForm1;  
h1,h2: Extended; // границы интервала изменения превышений  
Dh:Extended;// шаг изменения превышений  
s1,s2:Extended;// интервал изменения горизонтальных проложений  
Ds: Extended; // шаг изменения горизонтальных проложений  
implementation  
{ $R *.dfm }  
procedure TForm1.Button2Click(Sender: TObject);  
begin  
Form1.Close
```

```

end;
procedure TForm1.Button1Click(Sender: TObject);
Var i:Extended;// текущее значение уклона в тангенсах
    ig:Integer; // градусная часть уклона
    im:Integer; // минутная часть уклона
    h,s:Extended;// текущее значение превышения и
// горизонтального проложения
    ARi:Extended;// текущее значение уклона в градусах
begin
// ввод исходных данных для программы
h1:=StrToFloat(Edit1.Text);
h2:=StrToFloat(Edit2.Text);
Dh:=StrToFloat(Edit3.Text);
s1:=StrToFloat(Edit4.Text);
s2:=StrToFloat(Edit5.Text);
Ds:=StrToFloat(Edit6.Text);
// вывод шапки таблицы уклонов
Label6.Caption:='Превышение Горизонтальное проложение Уклон в тангенсах'
+ ' Уклон в гр., мин.';
// текущее значение превышения принимает значение нижней границы
h:=h1;
{ цикл выполняется, пока текущее значение превышения
не превысит значение правой верхней границы}
While h <= h2 Do
Begin
// текущее значение горизонтального проложения принимает значение
// нижней границы
s:=s1;
{ цикл выполняется, пока текущее значение горизонтального проложения
не превысит значение правой верхней границы}
While s <= s2 Do
begin
// вычисление текущего значения уклона в тангенсах
i:=h/s;
// перевод уклона в градусную меру
ARi:=ArcTan(i)/pi*180;
// выделение целой части — градусная часть уклона
ig := Trunc(ARi);
// вычисление минутной части значения уклона
im:=Round((ARi-ig)*60);
// вывод строки таблицы уклонов
Memo1.Lines.Add(' '+ FloatToStrF(h,Fffixed,6,1) +
'|'

```

```

+ FloatToStrF(s,Ffixed,15,2) + ' | '
+ FloatToStrF(i,Ffixed,15,5) + ' | ' +
'' + IntToStr(ig) + '' + IntToStr(im));
// увеличение горизонтального проложения на величину шага
s:=s+Ds;
end;
// увеличение превышения на величину шага
h:=h+Dh;
end;
end;
end.

```

Один из возможных результатов работы программы

При проектировании каналов, дорог, направления движения тракторных агрегатов на пашне, изучении эрозии почв определяют уклоны местности. Уклоном i называют отношение превышения h между точками к горизонтальному проложению S между ними.

Интервал изменения превышений:

Шаг изменения превышений:

Интервал изменения горизонтальных проложений:

Шаг изменения горизонтальных проложений:

Превышение | Горизонтальное проложение | Уклон в тангенсах | Уклон в гр., мин.

0,3	170,00	0,00176	0 6
0,3	180,00	0,00167	0 6
0,5	100,00	0,00500	0 17
0,5	110,00	0,00455	0 16
0,5	120,00	0,00417	0 14
0,5	130,00	0,00385	0 13
0,5	140,00	0,00357	0 12
0,5	150,00	0,00333	0 11

ВЫЧИСЛИТЬ

ВЫХОД

Задача

Составить программу для вычисления числа e .

Известно, что сумма ряда $1+1/1!+1/2!+\dots+1/n!$ стремится к числу e . Число $e=2,71828$ играет важную роль в качестве основания натуральных логарифмов.

Компоненты

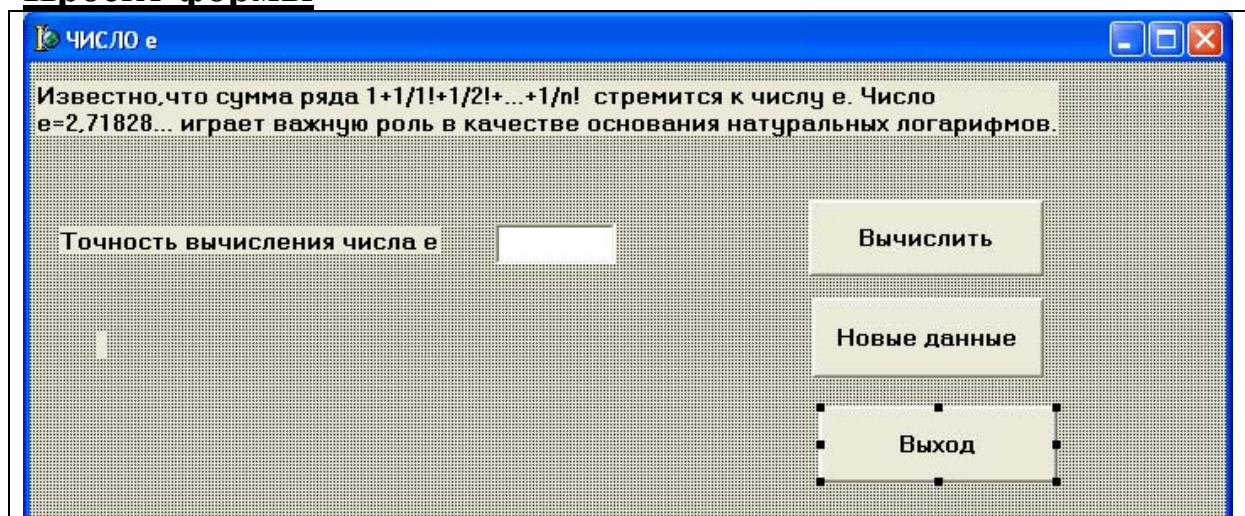
Имя компонента	Свойства компонента	Значение	Назначение
Form1	Caption	Число e	Заголовок формы

Label1	Caption	Известно, что сумма ряда $1+1/1!+1/2!+\dots+1/n!$ стремится к числу e . Число $e=2,71828$ играет важную роль в качестве основания натуральных логарифмов	Справочная информация для пользователя программы
Label2	Caption	Точность вычисления числа e	Подсказка пользователю
Edit1	Text		Поле для ввода точности вычисления числа e
Button1	Caption	Вычислить	Кнопка для вычисления
Button2	Caption	Новые данные	Кнопка для очистки введенного значения точности
Button3	Caption	Выход	Кнопка для завершения работы программы
Label3	Caption		Поле, в которое программа выводит вычисленное значение числа e

Переменные

Обозначение в программе	Содержание	Тип
n	Номер члена ряда -1	Целый
e	Число e	Вещественный, расширенный
z	Знаменатель члена ряда	Длинный целый
element	Текущий элемент ряда	Вещественный, расширенный
T	Точность вычисления числа e	Вещественный, расширенный

Проект формы



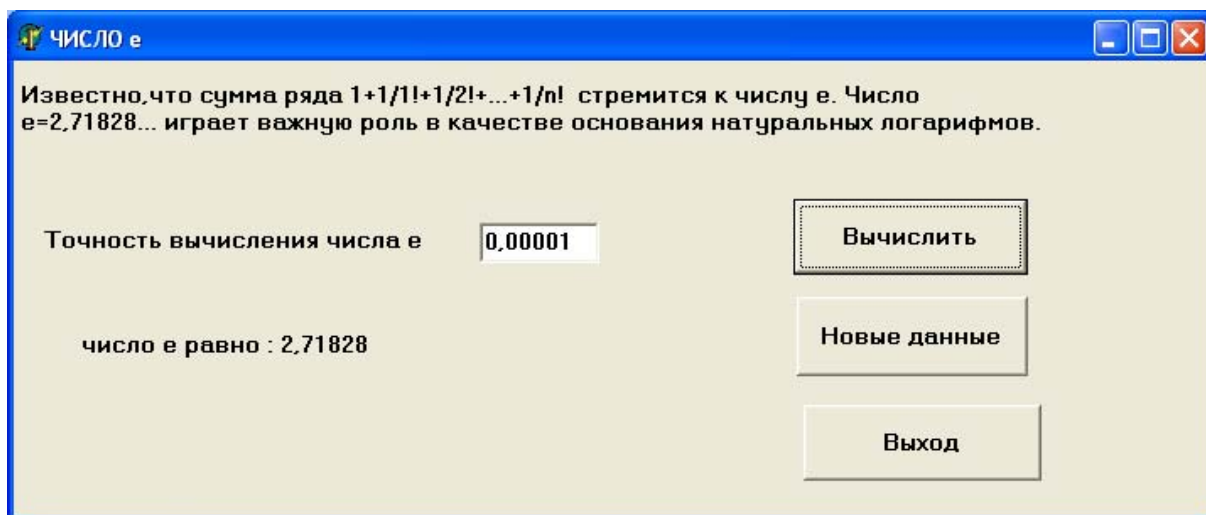
Текст модуля

unit Unit1;

```

.....
var
    Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Button2Click(Sender: TObject);
begin
    Edit1.Clear
end;
procedure TForm1.Button3Click(Sender: TObject);
begin
    Form1.Close
end;
procedure TForm1.Button1Click(Sender: TObject);
Var e: Extended; // число e
    element:Extended; // текущий элемент ряда
    T: Extended;// точность вычисления числа e
    n:Integer; // номер члена ряда -1
    z:LongInt; // знаменатель члена ряда
begin
// инициализируем числом 1 значения переменных e, z, n
e:=1;
z:=1;
n:=1;
// ввод точности вычисления числа e
T:=StrToFloat(Edit1.Text);
//цикл вычисления суммы ряда для числа e
Repeat
z:=Z*n;
element:=1/z ;
e:=e+element;
n:=n+1;
{ выход из цикла происходит, когда очередной член ряда
меньше или равен точности вычисления числа e}
Until T >=element ;
// вывод вычисленного числа e
Label3.Caption:='число e равно: '+FloatToStrF(e,Fffixed,10,8)
end;
end.

```



Задачи для самостоятельного решения

Циклы с условием

№ 4.1. Написать программу для вычисления значений $Y = Z^2$ при изменении Z от $N1$ до $N2$ с шагом K , где $K, N1, N2$ — натуральные числа.

№ 4.2. Написать программу для вычисления суммы нечетных чисел натурального ряда в интервале от $N1$ до $N2$, где $N1, N2$ — натуральные числа.

№ 4.3. Написать программу для вычисления значений $Q = \sqrt{Y}$ при изменении Y от $N1$ до $N2$ с шагом K , где $K, N1, N2$ — натуральные числа.

№ 4.4. Написать программу для вычисления суммы обратных величин чисел от $N1$ до $N2$ с шагом K , где $K, N1, N2$ — натуральные числа.

№ 4.5. Написать программу для вычисления выражения $S = \pi * R * R$ при изменении R в интервале от 1 до N с шагом K где K, N — натуральные числа.

№ 4.6. Составить программу для вычисления членов последовательности:

$$S_n = \frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \dots + \frac{1}{n \cdot (n+1)}, \text{ где } n \text{ — натуральное число.}$$

Члены последовательности вычислять до тех пор, пока текущий член последовательности не станет меньше заданного числа ε ($0 < \varepsilon < 0,0001$).

№ 4.7. Составить программу для вычисления суммы ряда:

$\frac{1}{1^3} + \frac{1}{2^3} + \frac{1}{3^3} + \dots + \frac{1}{n^3}$, где n — натуральное число.

Сумму ряда вычислять до тех пор, пока текущий член последовательности не станет меньше заданного числа ε ($0 < \varepsilon < 0,0001$).

№ 4.8. Составить программу для вычисления суммы ряда:

$$1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \frac{1}{6} + \dots$$

Сумму ряда вычислять до тех пор, пока текущий член последовательности не станет меньше заданного числа ε ($0 < |\varepsilon| < 0,0001$).

№ 4.9. Составить программу для вычисления члена a_n последовательности a_1, a_2, \dots , образованной по закону

$$a_n = \frac{n}{\sqrt{n^2 + 1} - \sqrt{n^2 - 1}},$$

для которого выполняется условие $|a_n - a_{n-1}| < \varepsilon$,

причем n — номер члена последовательности ($n \geq 2$),

ε — заранее заданное положительное число.

№ 4.10. Написать программу для вычисления члена b_n последовательности b_1, b_2, \dots , образованной по закону

$$b_n = \left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{3}\right)\dots\left(1 - \frac{1}{n+1}\right),$$

для которого выполняется условие $|b_n - b_{n-1}| < \varepsilon$,

где n — номер элемента последовательности ($n \geq 2$);

ε — заданное положительное число.

Циклы с параметром

№ 4.11. Написать программу для вычисления суммы $S = S + 1/i$, где $i = 1, 2, \dots, N$.

№ 4.12. Написать программу для вычисления произведения первых N натуральных чисел.

№ 4.13. Написать программу для вычисления $dx = h * h / (2 * S)$, при значениях: $100 \leq S \leq 1000$ м.

№ 4.14. Написать программу для вычисления суммы S из значений $Y = X * X * X$, где X задан в интервале от $N1$ до $N2$, где $N1$ и $N2$ — натуральные числа.

№ 4.15. Написать программу для вычисления выражения $Y=N!$.

§ 5. Использование массивов в программах, компонента Метод для ввода данных в массив

В данном параграфе рассматриваются следующие вопросы: массивы, компонент Метод для ввода и вывода массивов.

Теория

Массивы

Массивом называется упорядоченная индексированная совокупность однотипных элементов, имеющих общее имя. Элементами массива могут быть данные различных типов, включая структурированные. Каждый элемент массива однозначно определяется *именем* массива и *индексом* (номером этого элемента в массиве) или индексами, если массив многомерный. Для обращения к отдельному элементу массива указываются имя этого массива и номер (номера) элемента, заключенный в квадратные скобки, например, `arr1[3, 35]` или `arr3[7]`.

Количество индексных позиций определяет мерность массива (одномерный, двумерный и т. д.), при этом мерность массива не ограничивается. В математике аналогом одномерного массива является вектор, а двумерного массива — матрица. Индексы элементов массива должны принадлежать порядковому типу. Разные индексы одного и того же массива могут быть различных типов. Наиболее часто типом индекса является целочисленный тип.

Различают массивы *статические* и *динамические*. *Статический массив* представляет собой массив, границы индексов и соответственно размеры которого задаются при объявлении — известны до компиляции программы.

Формат описания типа статического массива:

`Array [Тип индексов] of <Тип элементов>;`

Пример. Объявление статических массивов.

`Type tm = Array[1 .. 10, 1 .. 100] of real;`

`Var arr1, arr2: tm;`

`arr3: Array[20 .. 100] of char;`

`arr4: Array['a' .. 'z'] of integer;`

Переменные `arr1` и `arr2` являются двумерными массивами по 1000 элементов — 10 строк и 100 столбцов. Каждый элемент этих массивов представляет собой число типа `real`. Для объявления массивов `arr1` и `arr2` введен специальный тип `tm`. Переменные `arr3` и `arr4` являются одномерными массивами, соответственно на 81 символ и 26 целых чисел.

Доступ к каждому элементу массива осуществляется по индексу. Например, `arr1[2,3]` — элемент массива `arr1`, находящийся во второй строке и третьем столбце; `arr3[34]` — элемент массива `arr3` с индексом 34, но пятнадцатый по порядку; `arr4[i]` — i -ый элемент массива `arr4`.

Компонент Мемо для ввода и вывода массивов

При использовании компонента Мемо для ввода массива значение каждого элемента массива следует вводить в отдельной строке и после ввода каждого элемента массива нажимать клавишу «ENTER».

Получить доступ к находящейся в поле Мемо строке текста можно при помощи свойства `Lines`, указав в квадратных скобках номер нужной строки (строки нумеруются с 0).

Следующий цикл демонстрирует ввод символьного массива из компонента Мемо:

```
For i:=1 to Size do
  a[i]:=Memo1.Lines[i];
```

Вывод массива в компонент Мемо можно осуществить таким образом:

```
For i:=0 to N-1
  Memo2.Lines.Add(FloatToStr(massiv[i+1]));
```

Практика

Задача

Составить программу для сортировки массива действительных чисел: по возрастанию или по убыванию по желанию пользователя. Результаты сортировки вывести в окно сообщения.

Компоненты

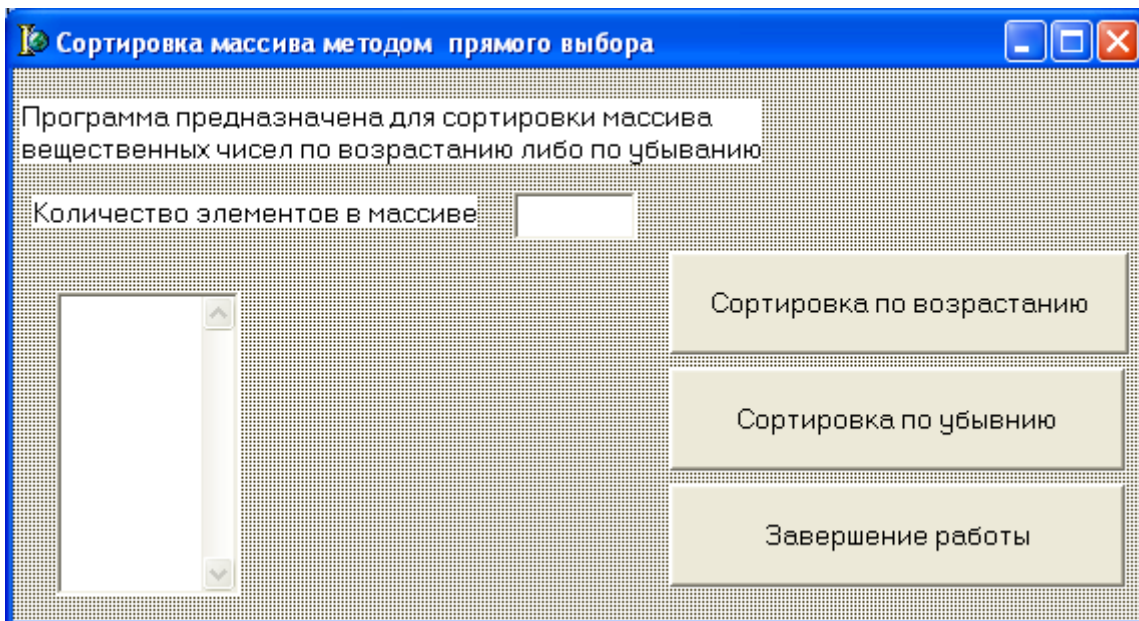
Имя компонента	Свойства компонента	Значение	Назначение
Form1	Caption	Сортировка массива методом прямого выбора	Заголовок формы
Label1	Caption	Программа предназначена для сортировки массива	Справочная информация для

		действительных чисел по возрастанию либо по убыванию	пользователя программы
Label2	Caption	Количество элементов в массиве	Подсказка пользователю
Edit1	Text	Поле для ввода количества элементов в массиве
Memo1	Lines	Компонент для ввода элементов массива в столбик (в одной строке — один элемент)
Button1	Caption	Сортировка по возрастанию	Кнопка для выполнения сортировки по возрастанию
Button2	Caption	Сортировка по убыванию	Кнопка для выполнения сортировки по убыванию
Button3	Caption	Завершение работы	Кнопка для завершения работы программы

Переменные и массивы

Обозначение в программе	Содержание	Тип
N	Количество элементов в массиве действительных чисел	Целый
Massiv	Массив действительных чисел	Вещественный, расширенный
I	Текущий индекс элемента массива	Целый
J	индекс элемента массива, с которого начинается поиск минимального либо максимального элемента	Целый
min	номер минимального элемента в массиве или части массива	Целый
max	номер максимального элемента в массиве или части массива	Целый
buf	буфер для обмена местами элементов массива	Вещественный, расширенный
st	строковое сообщение, содержащее результат	Строковый

Проект формы



Текст модуля

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls;
.....
var
  Form1: TForm1;
  Massiv: array [1..100] of Extended;// массив действительных чисел
  N : Integer;
implementation
{$R *.dfm}
procedure TForm1.Button3Click(Sender: TObject);
begin
  Form1.Close
end;
procedure TForm1.Button1Click(Sender: TObject);
  Var I:Integer;// индекс элемента массива
      min: Integer; // номер минимального элемента в массиве
      //или части массива
      j : Integer; // индекс элемента массива,
      //с которого начинается
      // поиск минимального элемента
      buf: Extended; // буфер для обмена местами
      // элементов массива
      St: String; // строковое сообщение, содержащее результат
begin

```



```

//ввод количества элементов в массиве
N:=StrToInt(Edit1.Text);
//ввод массива действительных чисел
For I:=1 To N do
Massiv[I]:=StrToFloat(Memo1.Lines[I-1]);
// поиск минимального элемента массива методом прямого выбора
{алгоритм метода прямого выбора можно описать следующими слова-
ми. Сначала в качестве минимального элемента берется первый элемент и
с ним последовательно сравниваются все следующие элементы массива,
если попадает элемент меньше принятого за минимальный, то они ме-
няются местами. В результате на первом месте оказывается самый малень-
кий элемент массива. Затем берется второй или следующий элемент мас-
сива и просматривается с перестановкой оставшаяся часть массива. Эта
последовательность шагов повторяется, пока номер минимального элемен-
та не достигнет N-1}
For I:=1 to N-1 do
begin
min:=I;
For j:=I+1 to N do
if Massiv[j] < Massiv[min]
then min:=j;
buf:=Massiv[I];
Massiv[I]:=Massiv[min];
Massiv[min]:=buf;
end;
// вывод результатов сортировки
st:='Отсортированный по возрастанию массив'+#13;
For I:=1 to N do
st:=st+IntToStr(i)+ ' '+FloatToStrF(Massiv[I],Ffixed,10,2)+#13;
ShowMessage(st);
end;
procedure TForm1.Button2Click(Sender: TObject);
Var I:Integer;// индекс элемента массива
max: Integer; // максимальный элемент в массиве
j : Integer; // индекс элемента массива, с которого
// начинается поиск максимального элемента
buf: Extended; // буфер для обмена местами элементов
// массива
St:String;// строковое сообщение, содержащее результат
begin
// ввод количества элементов в массиве
N:=StrToInt(Edit1.Text);
// ввод массива действительных чисел

```

```

For I:=1 To N do
Massiv[I]:=StrToFloat(Memo1.Lines[I-1]);
// поиск максимального элемента массива методом прямого выбора
// алгоритм аналогичен поиску минимального элемента
// отличие состоит в знаке сравнения
For I:=1 to N-1 do
begin
max:=I;
For j:=I+1 to N do
if Massiv[j] > Massiv[max] then max:=j;
buf:=Massiv[I];
Massiv[I]:=Massiv[max];
Massiv[max]:=buf;
end;
// вывод результатов сортировки
st:='Отсортированный по убыванию массив'+#13;
For I:=1 to N do
st:=st+IntToStr(i)+ ' '+FloatToStrF(Massiv[I],Ffixed,10,2)+#13;
ShowMessage(st);
end;
end.

```

Задачи для самостоятельного решения

Группа А

- № 5.1. Дан массив действительных чисел. Найти сумму элементов массива.
- № 5.2. Дан массив действительных чисел. Найти произведение элементов массива.
- № 5.3. Дан массив действительных чисел. Найти сумму квадратов элементов массива.
- № 5.4. Дан массив действительных чисел. Найти модуль суммы и квадрат произведения элементов массива.
- № 5.5. Дан массив действительных чисел. Найти наибольший элемент массива.
- № 5.6. Дан массив действительных чисел. Найти наименьший элемент массива.
- № 5.7. Дан массив действительных чисел. Найти наибольший элемент массива.
- №5.8. Дан массив действительных чисел. Найти наименьший из элементов массива с четными номерами.

№5.9. Написать программу для определения среднего роста студентов группы.

№5.10. Написать программу для вычисления среднего гармонического

$$M_n = n / (1/a_1 + 1/a_2 + \dots + 1/a_n), \text{ где } a_i \neq 0.$$

Группа Б

№ 5.11. Написать программу для обработки результатов N-кратного измерения величины L. Программа должна вычислять Среднее

$$\bar{L} = \frac{1}{N} \sum_{i=1}^N L_i,$$

где L_i — результат i-го измерения.

Среднюю квадратическую погрешность отдельного измерения

$$m = \sqrt{\frac{N(\overline{L^2} - (\bar{L})^2)}{N-1}},$$

где

$$\overline{L^2} = \frac{1}{N} \sum_{i=1}^N L_i^2.$$

Среднюю квадратическую погрешность арифметической середины

$$M = m / \sqrt{N}.$$

Предельную абсолютную погрешность измерений $\Delta L = 3M$.

Количество измерений произвольно в пределах от 2 до 15.

В программе предусмотреть следующую форму запросов при вводе исходных данных:

КОЛИЧЕСТВО ИЗМЕРЕНИЙ=

L1=

L2=

и т. д. для всех измерений.

В программе предусмотреть также проверку условия $N < 2$. Если это условие выполняется, выдавать сообщение «ОДНО ИЗМЕРЕНИЕ НЕДОПУСТИМО» и осуществить переход к оператору ввода N.

Форма выдачи результатов:

УСРЕДНЕННЫЙ РЕЗУЛЬТАТ=значение

СКП ОТДЕЛЬНОГО ИЗМЕРЕНИЯ= значение

СКП СРЕДНЕГО= значение

ПРЕДЕЛЬНАЯ АБСОЛЮТНАЯ ПОГРЕШНОСТЬ= значение

№ 5.12. Написать программу для определения суммарной площади земельных участков

$$S = S_1 + S_2 + \dots + S_N$$

и вычисления средней квадратической погрешности определения этой суммы

$$M = \sqrt{M_1^2 + M_2^2 + \dots + M_N^2}.$$

Здесь S_i — площадь i -го участка; M_i — средняя квадратическая погрешность определения этой площади. Общее количество участков произвольно, но не более 255.

Предусмотреть в программе следующую форму запросов при вводе:

КОЛИЧЕСТВО УЧАСТКОВ=

УЧАСТОК 1:

ПЛОЩАДЬ=

СКП=

УЧАСТОК 2:

ПЛОЩАДЬ=

СКП=

и т.д. для всех участков.

Форма вывода результатов:

СУММАРНАЯ ПЛОЩАДЬ=значение S

СКП=значение M

№ 5.13. Написать программу для вычисления среднего арифметического

$$P = \frac{X_1 + X_2 + \dots + X_N}{N},$$

среднего геометрического

$$G = \sqrt[N]{|X_1 \cdot X_2 \cdot \dots \cdot X_N|},$$

среднего гармонического

$$H = \frac{N}{1/X_1 + 1/X_2 + \dots + 1/X_N}$$

и среднего квадратичного

$$B = \sqrt{X_1^2 + X_2^2 + \dots + X_N^2}$$

для N — чисел. N — произвольное число, но не более 100.

(Указание $\sqrt[N]{A} = A^{1/N}$).

В программе предусмотреть следующие запросы при вводе:

КОЛИЧЕСТВО ЧИСЕЛ=

X1=

X2=

и т.д. для всех чисел.

Форма вывода результатов:

СРЕДНЕЕ АРИФМЕТИЧЕСКОЕ=значение

СРЕДНЕЕ ГЕОМЕТРИЧЕСКОЕ= значение

СРЕДНЕЕ ГАРМОНИЧЕСКОЕ= значение

СРЕДНЕЕ КВАДРАТИЧНОЕ= значение

№ 5.14. Написать программу для вычисления средних арифметических:

$$P_1 = \frac{1}{N} \sum_{i=1}^N X_{1i}, P_2 = \frac{1}{N} \sum_{i=1}^N X_{2i}$$

и средних квадратических отклонений

$$\sigma_1 = \sqrt{X_1^2 - P_1^2}, \sigma_2 = \sqrt{X_2^2 - P_2^2}.$$

Где

$$\overline{X_1^2} = \frac{1}{N} \sum_{i=1}^N X_{1i}^2, \overline{X_2^2} = \frac{1}{N} \sum_{i=1}^N X_{2i}^2$$

для двух массивов чисел, а также коэффициента корреляции между этими массивами:

$$r = \frac{\overline{X_1 X_2} - P_1 P_2}{\sigma_1 \sigma_2},$$

где

$$\overline{X_1 X_2} = \frac{1}{N} \sum_{i=1}^N X_{1i} X_{2i}.$$

№ 5.15. Написать программу для вычисления суммарной площади четырех прямоугольных участков (рис. 6.5) по заданным значениям их диагоналей L_i (в метрах) и углов между диагоналями и основаниями α_i . Формула для определения площади i -го участка:

$$S_i = L_i^2 \cdot \sin \alpha_i \cdot \cos \alpha_i$$

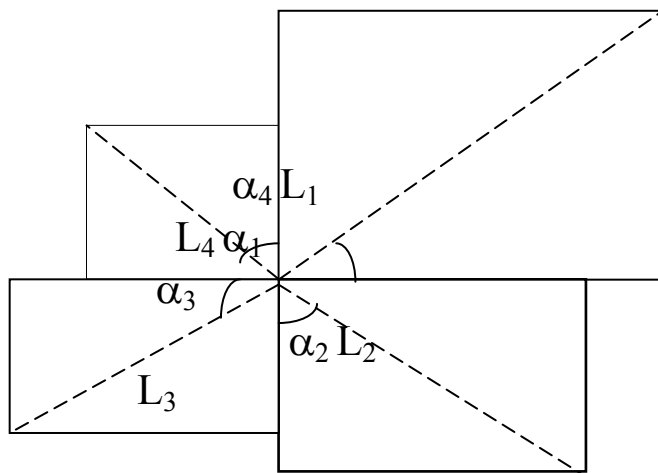


Рис. 6.5

В программе предусмотреть следующую форму запросов при вводе исходных данных:

1-Й УЧАСТОК:

ДИАГОНАЛЬ=

УГОЛ В ГР., МИН., СЕК.=

2-Й УЧАСТОК:

ДИАГОНАЛЬ=

УГОЛ В ГР., МИН., СЕК.=

3-Й УЧАСТОК:

ДИАГОНАЛЬ=

УГОЛ В ГР., МИН., СЕК.=

4-Й УЧАСТОК:

ДИАГОНАЛЬ=

УГОЛ В ГР., МИН., СЕК.=

Форма вывода результата:

ОБЩАЯ ПЛОЩАДЬ= значение суммарной площади (в га).

№ 5.16. Написать программу для определения суммы горизонтальных проложений между точками с указанными координатами:

$$S = \sum_{i=1}^{N-1} \sqrt{(X_{i+1} - X_1)^2 + (Y_{i+1} - Y_1)^2}.$$

Общее количество точек произвольно, но не более 20.

В программе предусмотреть следующую форму запросов при вводе исходных данных:

КОЛИЧЕСТВО ТОЧЕК=

X1=

X2=

и т.д. для всех точек.

Форма вывода результата:

СУММА ГОРИЗОНТАЛЬНЫХ ПРОЛОЖЕНИЙ= значение S.

№ 5.17. Написать программу для определения суммарного расстояния между точками с заданными координатами и значениями вертикального угла, под которым из данной точки видна следующая точка:

$$S = \sum_{i=1}^{N-1} \frac{\sqrt{(X_{i+1} - X_i)^2 + (Y_{i+1} - Y_i)^2}}{\cos \alpha_i}.$$

Общее количество точек произвольно, но не более 25.

В программе предусмотреть следующую форму запросов при вводе исходных данных:

КОЛИЧЕСТВО ТОЧЕК=

КООРДИНАТЫ ТОЧКИ 1: X, Y=

ВЕРТИК. УГОЛ ИЗ ТОЧКИ 1 НА ТОЧКУ 2: ГР., МИН., СЕК.=

КООРДИНАТЫ ТОЧКИ 1: X, Y=

ВЕРТИК. УГОЛ ИЗ ТОЧКИ 1 НА ТОЧКУ 2: ГР., МИН., СЕК.=

и т.д. до предпоследней точки.

КООРДИНАТЫ ТОЧКИ номер последней точки: X, Y=

Форма вывода результата:

СУММАРНОЕ РАССТОЯНИЕ=значение S.

№ 5.18. Дана информация о багаже нескольких пассажиров. Багаж каждого пассажира характеризуется двумя числами: количеством вещей и общим весом вещей. Выяснить, имеется ли пассажир, багаж которого превышает багаж каждого из остальных пассажиров и по числу вещей, и по весу.

№ 5.19. Дана информация о багаже нескольких пассажиров. Багаж каждого пассажира характеризуется двумя числами: количеством вещей и общим весом вещей. Определить, имеются ли два пассажира, багажи которых совпадают по числу вещей и различаются по весу не более чем на 0,5 кг.

№ 5.20. Дана информация о багаже нескольких пассажиров. Багаж каждого пассажира характеризуется двумя числами: количеством вещей и общим весом вещей. Выяснить, имеется ли пассажир, багаж которого состоит из одной вещи весом не менее 30 кг.

§ 6. Разработка многооконных проектов.

Двумерные массивы. Компонент StringGrid


В данном параграфе рассматриваются следующие вопросы: компонент StringGrid, технология разработки многооконных проектов, ввод и вывод двумерных массивов с помощью компонента StringGrid.

Теория

Определение двумерного массива было дано в предыдущем параграфе.

Компонент StringGrid

Находится на вкладке Additional.

 TStringGrid — служит для представления текстовых данных в виде таблицы. Очень удобен для ввода и вывода массивов (одномерных и двухмерных). Доступ к каждому элементу таблицы происходит через свойство Cells. Наиболее важными для нас являются следующие свойства компонента StringGrid.

Свойство	Назначение
Name	Имя компонента
ColCount	Количество колонок таблицы
RowCount	Количество строк таблицы
Cells	Соответствующий таблице двумерный массив. Ячейка таблицы, находящаяся на пересечении столбца номер col и строки номер row определяется элементом cells[col, row]
FixedCols	Количество зафиксированных слева колонок таблицы. Зафиксированные колонки выделяются цветом и при горизонтальной прокрутке остаются на месте
FixedRows	Количество зафиксированных



	сверху строк таблицы. Зафиксированные строки выделяются цветом и при вертикальной прокрутке таблицы остаются на месте
Options.goEditing	Признак допустимости редактирования содержимого ячеек таблицы. True — редактирование разрешено. False — запрещено
Options.goTabs	Разрешает (True) или запрещает (False) использование клавиши <Tab> для перемещения курсора в следующую ячейку таблицы
Options.goAlwaysShowEditor	Признак нахождения компонента в режиме редактирования. Если значение свойства False, то для того, чтобы в ячейке появился курсор, надо начать набирать текст, нажать клавишу <F2> или сделать щелчок мышью

Технология создания многооконных проектов

Решение многих задач требует создания многооконных проектов. Одной из причин, вызывающей необходимость создания многооконных проектов, может быть разделение ввода исходных данных в программу и вывода результатов счета.

После загрузки самой среды Delphi на экране уже появляется заготовка проекта, состоящего из одной формы и одного модуля. Для добавления новой формы в проект можно воспользоваться кнопкой на панели кнопок быстрого доступа или командой из главного меню

	Добавление новой формы в проект. Можно с помощью команды File→New → New Form
---	--

	Вывод списка форм проекта
	Вывод списка модулей проекта

Вместе с формой в проект добавляется связанный с ней модуль. Например, для формы Form2 добавится модуль Unit2.

Основная программа обычно записывается в главном модуле, связанном с главной формой. К главному модулю нужно подключить модули дочерних форм, т. е. в раздел Uses дописать названия дочерних модулей. Например,

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics,  
Controls, Forms, Dialogs, StdCtrls, Grids, Unit2, Unit3;
```

Для вывода на экран дочерних форм используется процедура **Show (показать)**. Например,

```
Form2.Show;
```

В раздел public дочерних модулей целесообразно включить описание переменных и массивов, передающихся из родительского модуля.

Если в родительском модуле используются компоненты из дочерних форм, перед их названием пишется имя формы, где они находятся. Например, Form2.StringGrid1.

Ввод и вывод двумерных массивов с помощью компонента StringGrid

После создания на форме строковой таблицы (компонент StringGrid) и задания с помощью Инспектора Объектов ее свойств, в программе нужно записать группу операторов для подписи строк и столбцов таблицы, чтобы пользователю программы было понятно, как вводить данные. Их можно записать в отдельной процедуре, связанной с событием выход (Exit) из компонента Edit, где вводится размерность массива, или в процедуре, связанной с событием создания формы (FormCreate).

{Процедура подготовки таблицы на форме для ввода квадратной матрицы, которая выполняется при выходе из поля ввода размерности матрицы.}

```
procedure TForm1.Edit1Exit(Sender: TObject);
```

```
Var i,j:Integer;// номера строки столбца квадратной матрицы
```

```
begin
```

```
// порядок матрицы получает свое значение из поля ввода
```

```
N:=StrToInt(Edit1.Text);
```

```
// свойство количество строк компонента StringGrid
```

```

// получает значение: порядок +1
StringGrid1.RowCount:=N+1;
// свойство количество столбцов компонента StringGrid
// получает значение порядок: +1
StringGrid1.ColCount:=N+1;
//подпись заголовка таблицы
For i:=1 to StringGrid1.RowCount do
StringGrid1.Cells[i,0]:= IntToStr(i);
// подпись строк таблицы
For j:=1 to StringGrid1.ColCount do
StringGrid1.Cells[0,j]:= IntToStr(j);
// добавление в свойства таблицы признака
//допустимости редактирования содержимого ячеек таблицы
StringGrid1.Options:= StringGrid1.Options+[goEditing];
end;

```

Замечание

Напоминаем, что для создания процедуры, связанной с каким-то событием, произошедшим с компонентом, нужно перейти на вкладку Events (события), найти нужное событие в строке, например OnCreate, и справа вписать название события (в данном примере Create). В модуле автоматически будет создана заготовка процедуры создания формы:

```

procedure TForm1.FormCreate(Sender: TObject);
begin

end;

```

Если щелкнуть по компоненту Edit, то автоматически будет создана заготовка процедуры обработки события изменение компонента Edit:

```

procedure TForm1.Edit1Change(Sender: TObject);
begin

end;

```

Нужно изменить событие Change (изменить) на событие Exit (выход). Для этого обязательно перейти на вкладку события (Events) в инспекторе объектов. В строке OnExit раскрыть список, выбрать название Edit1Change и заменить Change на Exit.

Для ввода двумерного массива из StrigGrid необходимо записать операторы, подобные этим:

```

For i:=1 to StringGrid1.RowCount-1 do
  For j:=1 to StringGrid1.ColCount-1 do
  begin
    Massiv[i,j]:=StrToFloat(StringGrid1.Cells[j,i]);
  end; {в данном случае begin и end можно опустить}

```

Для вывода:

```

For i:=1 to StringGrid1.ColCount-1 do
  For j:=1 to StringGrid1.RowCount-1 do
  StringGrid1.Cells[j,i]:=FloatToStr(Massiv[i,j]);

```

Понятие о классах в Delphi

В языке объектно-ориентированного программирования Delphi дается возможность определять так называемые *классы*. Класс — это сложная структура, включающая помимо описания данных, описание процедур и функций, которые могут быть выполнены над представителем класса — *объектом*.

Система Delphi всегда сама генерирует свои классы во время проектирования приложения. Описание сформированного класса находится в разделе **type** интерфейсной части модуля. Например, в программе про бак для летнего душа из темы 1 описание класса выглядит так:

```

type
  TForm1 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    Label5: TLabel;
    Label6: TLabel;
    Button4: TButton;
  procedure Button1Click(Sender: TObject);
  procedure Button3Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);

```

```

procedure Button4Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

```

Как мы видим, в класс включены компоненты (данные) и методы их обработки (процедуры). В секции **private** (закрытые) объявляются элементы класса, которые должны быть недоступны в других модулях. В секции **public** (открытые) помещаются элементы, доступные в других модулях.

Практика

Задача

Составить программу для транспонирования квадратной матрицы порядка N и определения следа матрицы.

Транспонированной матрицей к исходной является такая матрица, у которой элементы строк исходной матрицы расставлены в столбцы, а элементы столбцов исходной в строки.

Следом матрицы называется сумма главных диагональных элементов.

Компоненты

Имя компонента	Свойства компонента	Значение	Назначение
Form1	Caption	Матрица	Заголовок главной формы
Label1	Caption	Программа предназначена для преобразования квадратной матрицы (количество строк равно количеству столбцов)	Справочная информация для пользователя программы
Label2	Caption	Размерность матрицы	Подсказка пользователю
Edit1	Text	Поле для ввода размерности матрицы
Form1.StringGrid1	ColCount	5	Количество столбцов по умолчанию
	FixedCol	1	Количество фиксированных слева

	FixedRow	1	столбцов Количество фиксированных сверху строк
	Options: goFixedVertLine	True	Опции: Разметка зафиксированных вертикальных линий
	goFixedHorzLine	True	Разметка зафиксированных горизонтальных линий
	goVertLine	True	Разметка вертикальных линий внутри таблицы
	goHorzLine	True	Разметка горизонтальных линий внутри таблицы
	goTabs	True	Разрешает использование клавиши Tab для перемещения в другую ячейку
	goAlwaysShowEditor	True	Признак нахождения компонента в режиме редактирования
Form1.Button1	Caption	Транспонированная матрица	Кнопка для выполнения транспонирования и вывода в новое окно
Form1.Button2	Caption	След матрицы	Кнопка для вычисления следа матрицы и вывода в новое окно
Form1.Button3	Caption	Выход	Кнопка для завершения работы программы
Form2	Caption	Транспонированная матрица	Заголовок формы
Form2.StringGrid1	ColCount	5	Количество столбцов по умолчанию

	FixedCol	1	Количество фиксированных слева столбцов
	FixedRow	1	Количество фиксированных сверху строк
	Options: goFixedVertLine	True	Опции: Разметка зафиксированных вертикальных линий
	goFixedHorzLine	True	Разметка зафиксированных горизонтальных линий
	goVertLine	True	Разметка вертикальных линий внутри таблицы
	goHorzLine	True	Разметка горизонтальных линий внутри таблицы
	goTabs	True	Разрешает использование клавиши Tab для перемещения в другую ячейку
	goAlwaysShowEditor	True	Признак нахождения компонента в режиме редактирования
Form2.Button1	Caption	Выход	Кнопка для закрытия окна "Транспонированная матрица"
Form3	Caption	След матрицы	Заголовок формы
Label1	Caption	Сумма главных диагональных элементов матрицы называется следом матрицы	Справочная информация для пользователя программы
Label2	Caption		Поле для вывода, рассчитанного программой значения следа матрицы

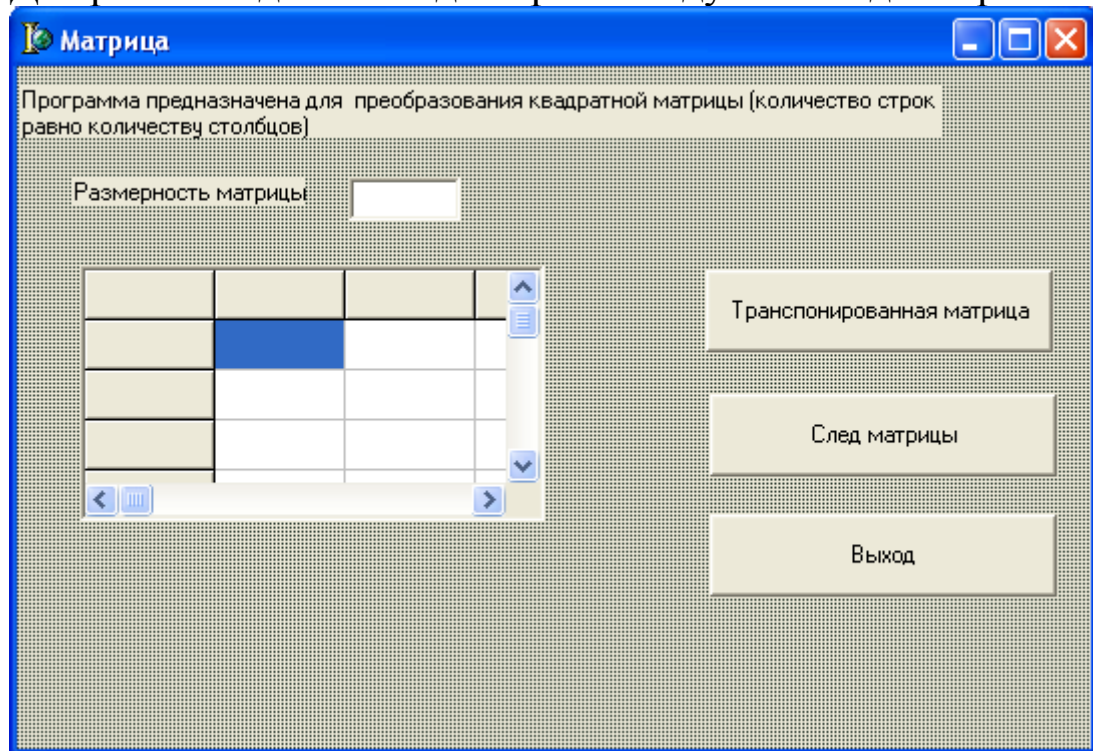
Form3.Button1	Caption	Выход	Кнопка для закрытия окна "След матрицы"
---------------	---------	-------	---

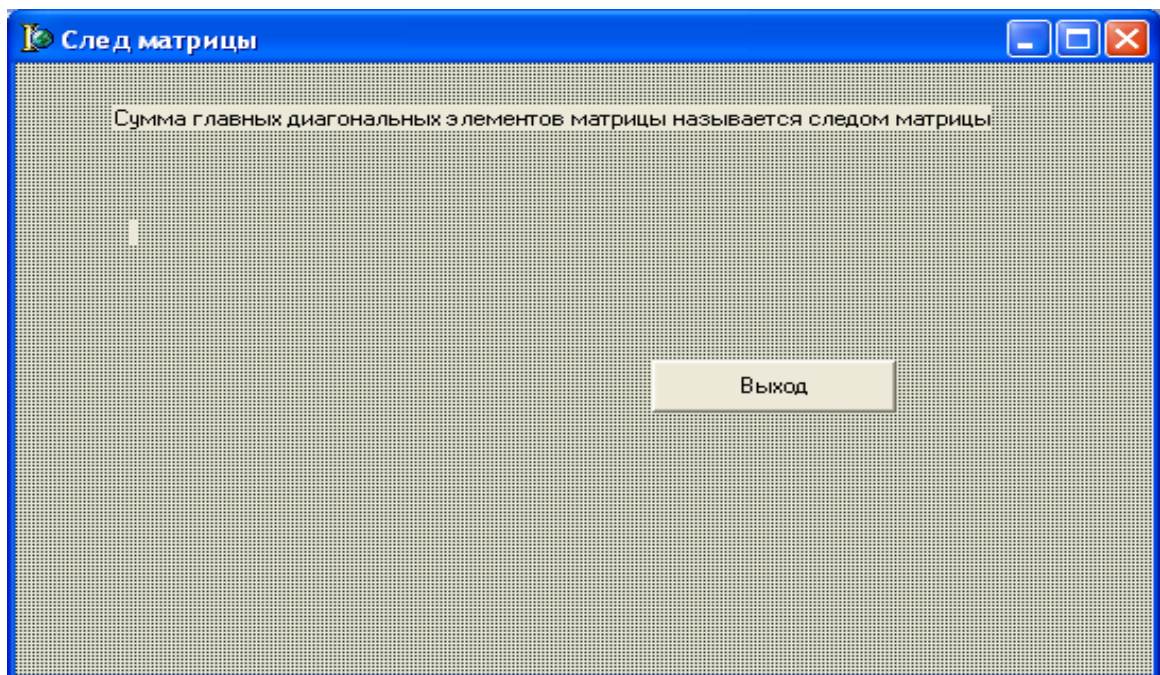
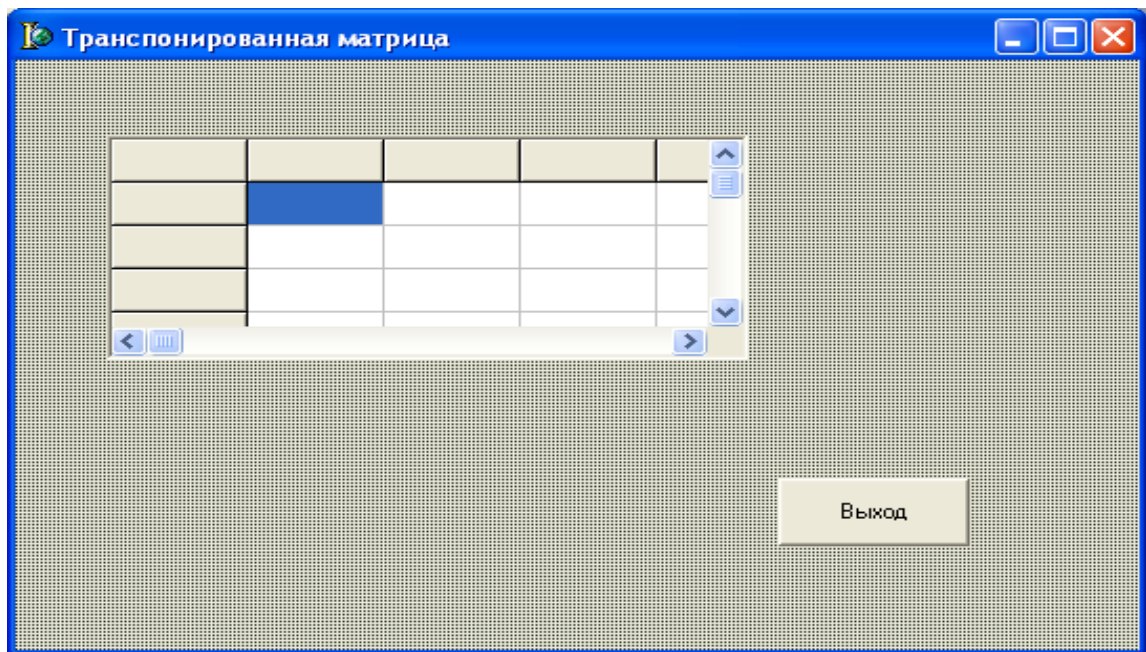
Переменные и массивы

Обозначение в программе	Содержание	Тип
N	Порядок квадратной матрицы	Целый
M	Квадратная матрица действительных чисел	Вещественный, расширенный
I	Текущий номер строки матрицы или столбца в StringGrid	Целый
J	Текущий номер столбца матрицы или строки в StringGrid	Целый
Sled	След матрицы	Вещественный, расширенный

Проекты форм

Для решения данной задачи рекомендуется создать три окна.





Текст модуля

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms, Dialogs, StdCtrls, Grids, Unit2, Unit3;
// подключаются модули, связанные со второй и третьей формами
type
  TForm1 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;

```

```

    Edit1: TEdit;
    StringGrid1: TStringGrid;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Edit1Exit(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
M :array [1..100,1..100] of Extended; // квадратная матрица
N:Integer;// порядок матрицы
Sled: Extended; // след матрицы
end;
var
    Form1: TForm1;
implementation
{$R *.dfm}
// процедура читает из таблицы на главной форме квадратную матрицу
// транспонирует матрицу и выводит в таблицу StringGrid
// на вторую форму
procedure TForm1.Button1Click(Sender: TObject);
Var i:Integer; // номер строки матрицы/номер столбца StringGrid
    j:Integer; // номер столбца матрицы/номер строки StringGrid
begin
// чтение значений квадратной матрицы из компонента StringGrid
// главной формы
For i:=1 to StringGrid1.RowCount-1 do
    For j:=1 to StringGrid1.ColCount-1 do
        begin
            M[i,j]:=StrToFloat(StringGrid1.Cells[j,i]);
        end;
// открывается вторая форма
Form2.Show;
// количеству строк и столбцов компонента StringGrid на второй форме
// присваивается значение порядок: матрицы +1
Form2.StringGrid1.RowCount:=N+1;
Form2.StringGrid1.ColCount:=N+1;
// подписываются заголовок таблицы, и номера строк
For i:=1 to Form2.StringGrid1.RowCount do

```

```

    Form2.StringGrid1.Cells[i,0]:= IntToStr(i);
  For j:=1 to Form2.StringGrid1.ColCount do
    Form2.StringGrid1.Cells[0,j]:= IntToStr(j);
  { вывод в компонент StringGrid на второй форме транспонированной
матрицы }
  { Транспонированная матрица не создается. Транспонирование осуще-
ствляется путем вывода столбца исходной матрицы в строку компонента
StringGrid. В StringGrid номер столбца i, номер строки j. }
  For i:=1 to Form2.StringGrid1.ColCount-1 do
    For j:=1 to Form2.StringGrid1.RowCount-1 do
      Form2.StringGrid1.Cells[i,j]:=FloatToStr(M[i,j]);
    end;
  { Процедура подготовки таблицы на форме для ввода квадратной матрицы,
которая выполняется при выходе из поля ввода размерности матрицы. }
  procedure TForm1.Edit1Exit(Sender: TObject);
  Var i,j:Integer;//номера строки столбца квадратной матрицы
  begin
    // порядок матрицы получает свое значение из поля ввода
    N:=StrToInt(Edit1.Text);
  // свойство количество строк компонента StringGrid
  // получает значение: порядок +1
    StringGrid1.RowCount:=N+1;
  // свойство количество столбцов компонента StringGrid
  // получает значение порядок: +1
    StringGrid1.ColCount:=N+1;
    //подпись заголовка таблицы
    For i:=1 to StringGrid1.RowCount do
      StringGrid1.Cells[i,0]:= IntToStr(i);
    //подпись строк таблицы
    For j:=1 to StringGrid1.ColCount do
      StringGrid1.Cells[0,j]:= IntToStr(j);
    // добавление в свойства таблицы признака
    // допустимости редактирования содержимого ячеек таблицы
    StringGrid1.Options:= StringGrid1.Options +[goEditing];
  end;
  procedure TForm1.Button3Click(Sender: TObject);
  begin
    Form1.Close
  end;
  // процедура вычисляет след матрицы и выводит результат
  // на третью форму
  procedure TForm1.Button2Click(Sender: TObject);
  Var i: Integer;

```

```

begin
Sled:=0;
Form3.Show;
For i:=1 to Form1.StringGrid1.ColCount-1 do
Sled:= Sled + M[i,i];
Form3.Label2.Caption:=' След матрицы равен ' + FloatToStr(Sled)
end;
end.
{ Модуль, связанный со второй формой}
unit Unit2;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms, Dialogs, StdCtrls, Grids;
type
TForm2 = class(TForm)
StringGrid1: TStringGrid;
Button1: TButton;
procedure Button1Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }

end;
var
Form2: TForm2;
implementation
{$R *.dfm}
procedure TForm2.Button1Click(Sender: TObject);
begin
Form2.Close
end;
end.
{ Модуль, связанный с третьей формой. }
unit Unit3;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls;
type
TForm3 = class(TForm)
Label1: TLabel;

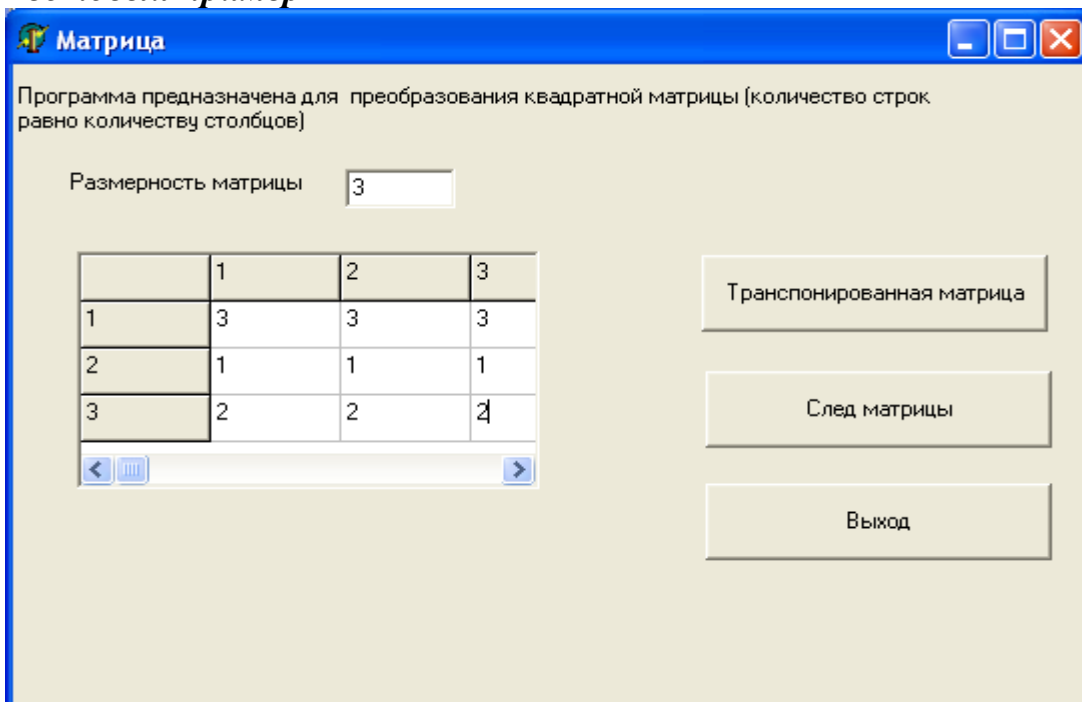
```

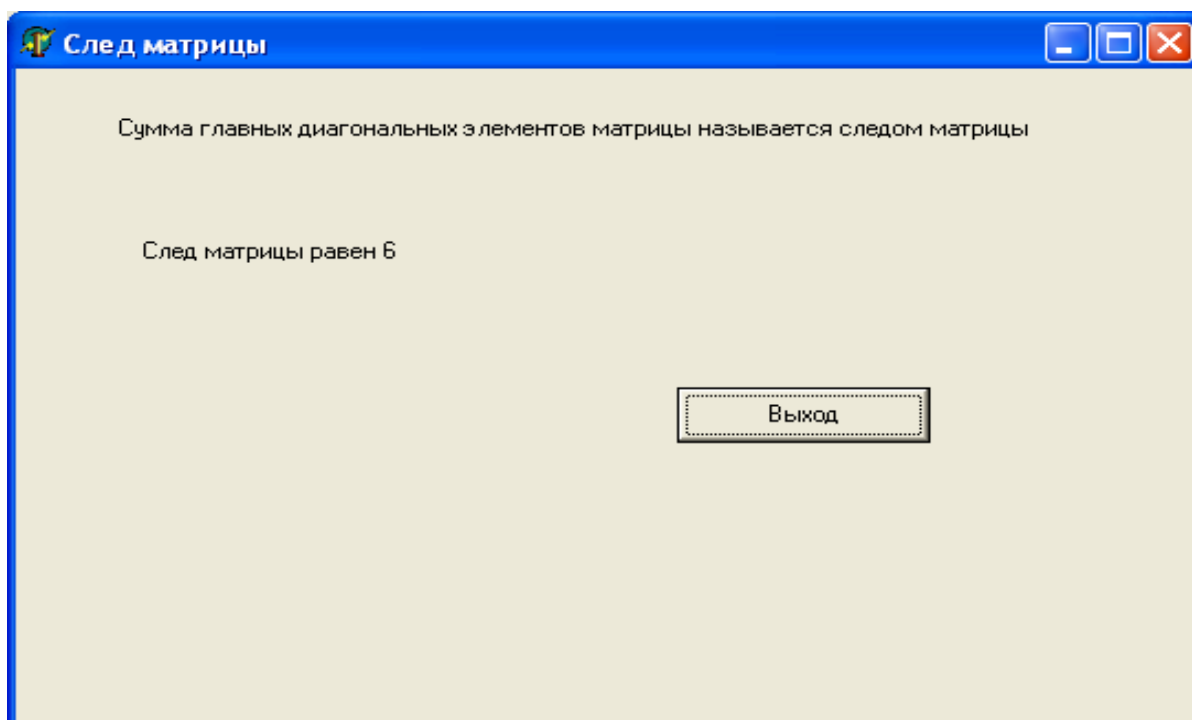
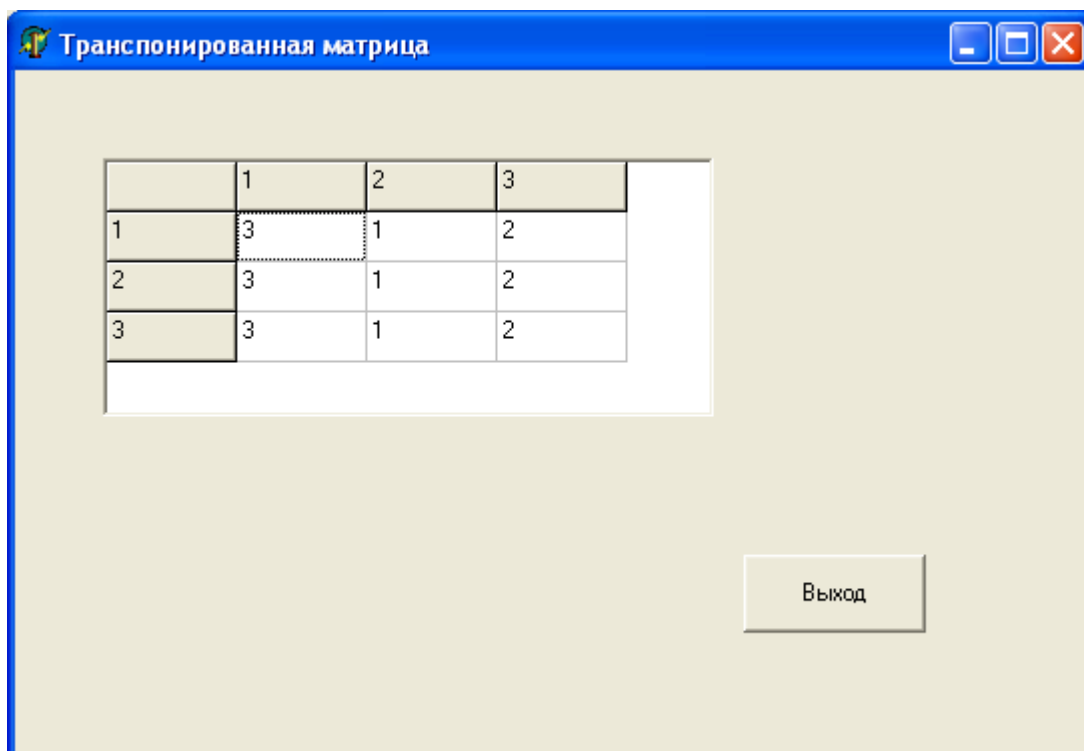
```

Label2: TLabel;
Button1: TButton;
procedure Button1Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
    M :array [1..100,1..100] of Extended;
    N:Integer;
    Sled: Extended;
end;
var
    Form3: TForm3;
implementation
{$R *.dfm}
procedure TForm3.Button1Click(Sender: TObject);
begin
    Form3.Close;
end;
end.

```

Тестовый пример





Задачи для самостоятельного решения

Группа А

№ 6.1. Дана действительная матрица размера $N \times M$ элементов:

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1m} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2m} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nm} \end{vmatrix}$$

Найти сумму положительных элементов матрицы.

№ 6.2. Дана целочисленная квадратная матрица порядка n . Найти номера строк: а) все элементы которых — нули; б) все элементы которых четны.

№ 6.3. Дана действительная матрица размера $N \times M$ элементов, все элементы которой различны. Найти наибольший и наименьший элементы матрицы.

№ 6.4. Дана целочисленная квадратная матрица порядка n . Подсчитать: а) количество нулей, стоящих на главной диагонали; б) количество единиц в матрице.

№ 6.5. Дана действительная матрица размера $N \times M$ элементов, все элементы которой различны. Указать номер строки и столбца, на пересечении которых находится минимальный элемент матрицы. Подсчитать количество отрицательных элементов.

№ 6.6. Дана целочисленная квадратная матрица порядка n . Найти номера строк: а) все элементы которых положительны; б) все элементы которых нечетны.

№ 6.7. Дана целочисленная квадратная матрица порядка n . Найти номера столбцов: а) все элементы которых отрицательны; б) все элементы которых равны нулю.

№ 6.8. Дана действительная матрица размера $N \times M$ элементов:

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1m} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2m} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nm} \end{vmatrix}$$

Найти номера строк матрицы, где все элементы различны.

№ 6.9. Дана целочисленная квадратная матрица порядка n . Подсчитать: а) произведение элементов главной диагонали; б) количество нулей.

№ 6.10. Дана действительная матрица размера $N \times M$ элементов, все элементы которой различны. Вывести элементы матрицы, которые меньше среднего арифметического из ее элементов.

Группа Б

№ 6.11. Дана матрица A размера $N \times M$ элементов, содержащая вещественные числа. Сформировать из нее массив B , состоящий из сумм положительных элементов каждой строки; если таких в строке нет, результат должен быть равен 0.

№ 6.12. Дана матрица A размера $N \times M$ элементов, содержащая вещественные числа. Сформировать из нее массив B , состоящий из произведений положительных элементов каждой строки; если таких в строке нет, результат должен быть равен 0.

№ 6.13. Дана матрица A размера $N \times M$ элементов, содержащая вещественные числа. Сформировать из нее массив B , состоящий из сумм отрицательных элементов каждой строки; если таких в строке нет, результат должен быть равен 0.

№ 6.14. Дана матрица A размера $N \times M$ элементов, содержащая вещественные числа. Сформировать из нее массив B , состоящий из числа отрицательных элементов каждой строки.

№ 6.15. Дана действительная матрица размера $N \times M$ элементов:

$$\begin{array}{|l} | a_{11} a_{12} a_{13} \dots a_{1m} | \\ | a_{21} a_{22} a_{23} \dots a_{2m} | \\ | \dots \dots \dots | \\ | a_{n1} a_{n2} a_{n3} \dots a_{nm} | \end{array}$$

Получить последовательность b_1, \dots, b_n , где b_k — произведение квадратов тех элементов k -й строки, модули которых принадлежат отрезку $[1, 1.5]$.

№ 6.16. Даны: натуральное число n , целочисленная квадратная матрица порядка n . Получить последовательность b_1, \dots, b_n , где b_i — сумма элементов, расположенных за первым отрицательным элементом в i -й строке (если все элементы строки неотрицательны, то принять $b_i = -1$).

№ 6.17. Дана действительная матрица размера $n \times m$, в которой не все элементы равны нулю. Получить новую матрицу путем деления всех элементов данной матрицы на ее наибольший по модулю элемент.

№ 6.18. Даны натуральное число n , действительное число x , действительная матрица размера $n \times 2n$. Получить последовательность b_1, \dots, b_n из нулей и единиц, где $b_i = 1$, если элементы i -й строки матрицы не превосходят x , и $b_i = 0$ в противном случае.

№ 6.19. Дана целочисленная квадратная матрица a порядка m . Строку с номером i матрицы назовем отмеченной, если $a_{ij} > 0$, и неотмеченной в противном случае.

а) Нужно все элементы, расположенные в неотмеченных строках матрицы, преобразовать по правилу: отрицательные элементы заменить на -1 , положительные — на 1 , а нулевые оставить без изменения.

б) Подсчитать число отрицательных элементов матрицы, расположенных в неотмеченных строках.

№ 6.20. Дана действительная квадратная матрица порядка n . Получить $x_1 x_n + x_2 x_{n-1} + \dots + x_n x_1$, где x_k — наибольшее значение элементов k -й строки данной матрицы.

§7. Внешние файлы

В данном параграфе рассматриваются следующие вопросы: процедуры работы с файлами, создание главного меню, использование стандартных диалогов

Теория

Процедуры работы с файлами

Файл — это именованная структура данных, представляющая собой последовательность элементов данного типа, причем количество элементов последовательности практически не ограничено.

Файл должен быть объявлен в разделе описания переменных. В общем виде объявление файла:

Имя: file of ТипЭлементов;

Файл, компонентами которого являются данные символьного типа, называется символьным или текстовым. Описание текстового файла в общем виде:

Имя: TextFile;

где Имя — имя файловой переменной, TextFile — обозначение типа, показывающее, что Имя — это файловая переменная, представляющая текстовый файл.

Примеры описания файлов:

Var Job: file of char; // файл символьного типа

Test: file of real; // файл вещественного типа

Fz: file of integer; // файл целого типа

My_File: TextFile; // файл текстового типа

Для того чтобы программа могла записывать результаты своей работы в файл или считывать исходные данные из файла, необходимо связать файловую переменную с конкретным файлом (задать имя файла).

Процедура **AssignFile** устанавливает физическую связь между файловой переменной *f* и строковой переменной *ИмяФайла*:

```
AssignFile (var f, ИмяФайла: string);
```

Примеры вызова процедуры **AssignFile**:

```
AssignFile(f, 'd:\Мои документы\result.txt');
```

```
AssignFile(Fin, 'test011.txt');
```

```
FileName:= 'otchet1.txt';
```

```
AssignFile(Fot, FileName); .
```

Прежде чем записывать информацию в файл необходимо его открыть. Возможны следующие режимы открытия файла для записи в него данных:

— перезапись (запись нового файла поверх существующего или создание нового файла);

— добавление в существующий файл.

Открыть для чтения файл можно с помощью процедуры **Reset(f)**, где *f* — имя файловой переменной. Процедура **Reset** ищет файл, открывает его и устанавливает указатель на начало файла. Если файл не найден, выдается сообщение об ошибке ввода/вывода и программа завершает свою работу.

Открыть файл для записи позволяет процедура **Rewrite(f)**, где *f* — имя файловой переменной. Процедура **Rewrite(f)** ищет файл, открывает его, стирает в нем старую информацию и устанавливает указатель на начало файла. Если файл не найден, он создается заново.

Для открытия файла с целью добавления в него информации служит процедура **AppendFile(f)**, где *f* — имя файловой переменной. Процедура **AppendFile** открывает файл и устанавливает указатель на конец файла. Если файл не найден, выдается сообщение об ошибке ввода/вывода и программа завершает свою работу.

Для ввода (чтения) информации из файла предназначены процедуры:

Read(f,[список ввода]);

Readln(f,[список ввода]);

где *f* — имя файловой переменной,

список ввода — переменные, элементы массивов.

Процедура **Readln** имеет следующие особенности:

— после ввода данных указатель переносится на начало следующей строки;

— процедура используется только для текстовых файлов.

Для вывода (записи) данных в файл существует процедуры:

Write(f,[список вывода]);

Writeln(f,[список вывода]);

где *f* — имя файловой переменной,

список вывода — переменные, элементы массивов, константы.

Отличительными особенностями процедуры **Writeln** также является применимость ее только в случае текстовых файлов и перенос указателя на новую строчку.

После окончания всех действий с файлом он должен быть обязательно закрыт с помощью процедуры **CloseFile(f)**, где *f* — имя файловой переменной.

Часто в программах при работе с файлами оказываются полезными функции **EOF(f)** и **EOLN(f)**.

EOF(f) — логическая функция, принимает значение **TRUE** (истина), если обнаружен конец файла.

EOLN(f) — логическая функция, принимает значение **TRUE** (истина), если обнаружен конец строки.

Главное меню

Компонент находится на вкладке **Standard**.



TMainMenu позволяет поместить главное меню в программу.

При помещении **MainMenu** на форму это выглядит, как просто иконка. Иконки данного типа называют «невидимыми компонентами», поскольку они невидимы во время выполнения программы. Создание меню включает три шага: (1) помещение **MainMenu** на форму, (2) вызов Дизайнера Меню через свойство **Items** в Инспекторе Объектов, (3) определение пунктов меню в Дизайнере Меню.

Чтобы создать процедуру, обрабатывающую действия, связанные с выбранным пунктом меню, необходимо произвести два щелчка по пункту на форме, и в редакторе кода будет подготов-

```
лена заготовка процедуры, подобная этой:  
procedure TForm1.N1Click(Sender: TObject);  
begin
```

```
end;
```

Использование стандартных диалогов

На странице Dialogs (диалоги) представлены компоненты для вызова стандартных диалогов Windows. Внешний вид диалогов зависит от используемой версии Windows. Объекты, представленные на данной странице невидимы во время выполнения и вызов диалогов происходит программно.



TOpenDialogs находится на странице Dialogs .

Позволяет выбрать файл из списка файлера (подобно тому, как открывается файл в Word с помощью команды **Открыть** .). Одним из главных свойств компонента OpenDialogs (кроме самого имени компонента) является свойство NameFile, которому присваивается имя выбранного из панели диалога файла. В программе для проверки, выбран файл или еще нет, используется функция Execute. Функция логическая. Принимает значение Истина, если файл выбран и Ложь в противном случае. Фрагмент программы для выбора файла из панели диалога может выглядеть так:

```
if not od1.Execute then Exit;  
NameFile := od1.FileName;
```

В этом примере, стандартное имя компонента OpenDialogs1 в Инспекторе Объектов было заменено на od1, для краткости. Программисты очень часто заменяют стандартные имена компонентов на свои (свойство Name).



Компонент TSaveDialog (находится также на странице Dialogs) используется для выбора имени файла, в котором будет сохранена информация из диалоговой панели. Обычно такая диалоговая панель вызывается командами **File→Save (Сохранить)** или **File→Save As (Сохранить как)**. Если пользователь указал имя файла и нажал кнопку Ok, то оно сохраняется как значение свойства FileName. Использование данного компонента аналогично использованию компонента OpenDialogs. Например,

```
if not SaveDialog1.Execute then Exit;
```

```
NameFile:=SaveDialog1.FileName;
```

Организация вывода на принтер

Для вывода на принтер нужно воспользоваться рядом процедур, определенных в модуле Printers. То есть в разделе Uses модуля программы нужно обязательно подключить модуль Printers:

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics,  
Controls, Forms, Grids, DateUtils, Printers;
```

Далее в программном модуле нужно ассоциировать значение тестовой переменной с принтером с помощью процедуры AssignPrn и открыть файл для записи. Например,

```
Var
```

```
TextToPrint: System.Text;
```

```
.....
```

```
AssignPrn(TextToPrint); // ассоциировать с принтером
```

```
Rewrite(TextToPrint); // открыть файл
```

После этого вызов стандартных процедур Write и Writeln приведет к выводу текста на устройство печати. Например, вывод содержимого компонента Memo на принтер можно осуществить с помощью следующих инструкций:

```
For Line:=0 to Memo1.Lines.Count-1 do
```

```
    Writeln(TextToPrint, Memo1.Lines[Line]);
```

После того как вывод информации завершен, необходимо вызвать стандартную процедуру Close:

```
    System.Close(TextToPrint);
```

Структура данных — запись

На практике мы часто сталкиваемся с задачами, где используются данные, состоящие из других данных. Например, информация о студентах может состоять не только из фамилии и инициалов, паспортных данных, номера группы, но и оценок, полученных на экзаменах. Для работы с такой информацией в Delphi используется структура данных — *запись* (record).

Запись — это структура данных, состоящая из полей. Поля — это именованные компоненты разного типа.

Переменную-запись можно в общем виде описать в разделе описания переменных как:

```
Имя-переменной: record
```

```
    Поле_1: Тип_1;
```

```
Поле_2: Тип_2;  
Поле_К: Тип_К;  
End;
```

Например,

```
Student: record  
  f_name: string[20];  
  l_name: string[20];  
  day: integer;  
  month: integer;  
  year: integer;  
  address: string[50];  
end;
```

Если в программе нужно использовать много переменных-записей одной структуры, то целесообразно в разделе Type объявить тип «запись»:

```
Имя = record  
  Поле_1: Тип_1;  
  Поле_2 : Тип_2;  
  Поле_К : Тип_К;  
End;
```

Например,

```
Type  
TDate = record  
  Day: integer;  
  Month: integer;  
  Year: integer;  
End;
```

В разделе описания переменных переменные-записи могут быть объявлены так:

```
Var  
  Birthday: TDate;  
  Dat1, Dat2: TDate;
```

Для работы с записями в языке Delphi предусмотрен оператор With. Он позволяет много раз не писать имя переменной-записи, а работать просто с полями. В общем виде оператор With выглядит так:

```
With Имя do  
  Begin
```

```
{операторы программы}  
End;
```

где:

Имя — имя переменной-записи;

With — зарезервированное слово, означающее, что далее, до слова end, при обращении к полям записи «Имя», имя записи можно не указывать.

Например, если в программе объявлена запись Student (см. выше) и данные о студенте вводятся в поля Edit1, Edit2 и Edit3 диалогового окна, то вместо операторов

```
Student.f_name:=Edit1.Text;  
Student.l_name:=Edit2.Text;  
Student.address:=Edit3.Text;
```

можно записать

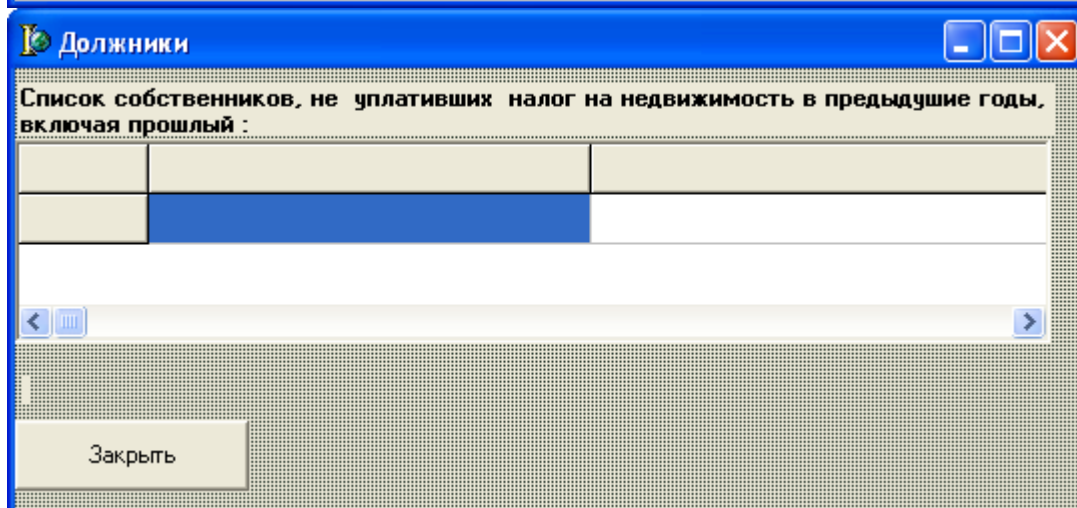
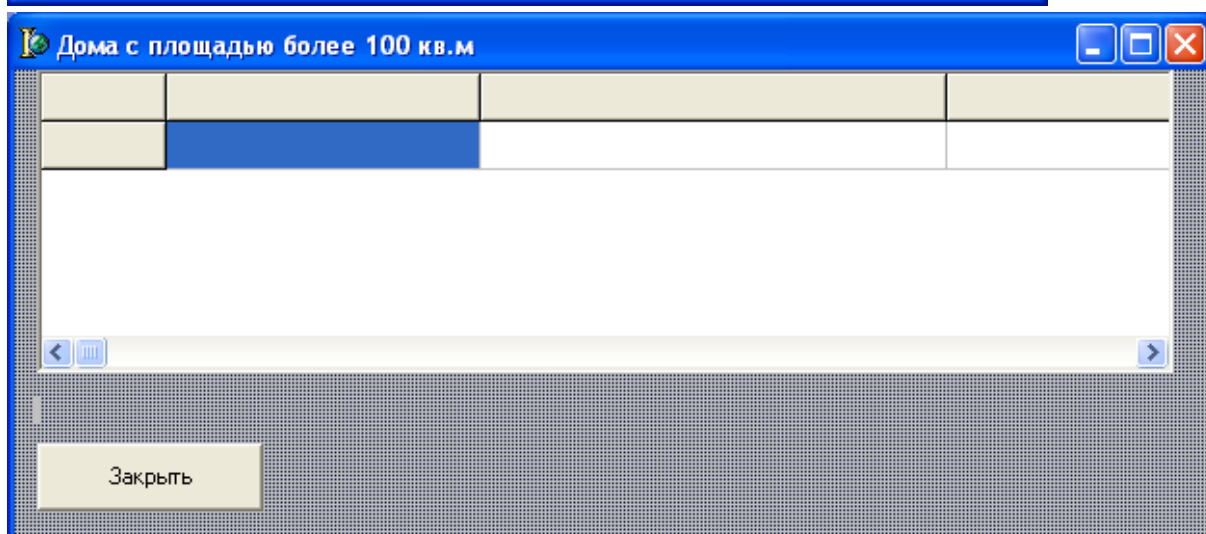
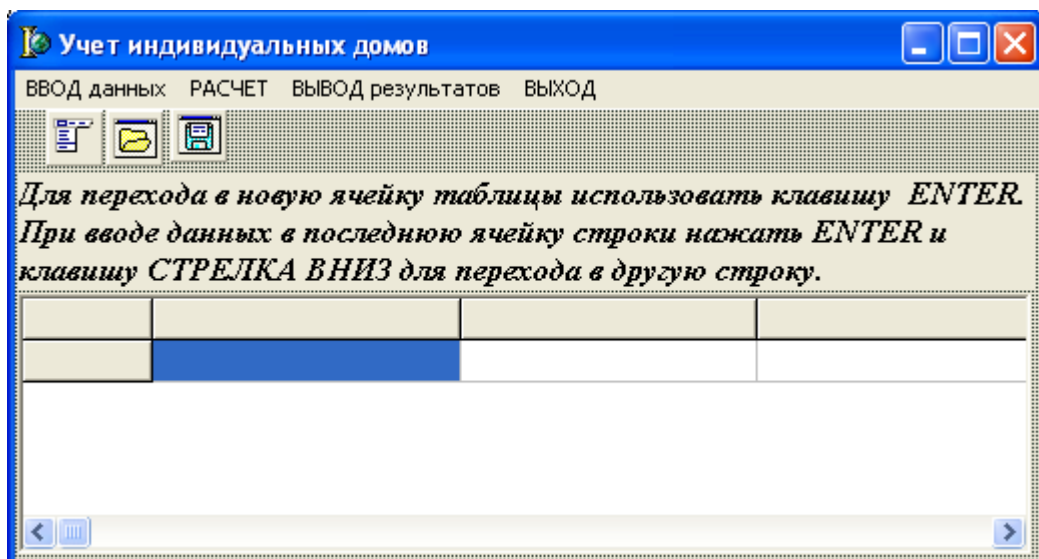
```
with student do  
begin  
f_name:=Edit1.Text;  
l_name:=Edit2.Text;  
address:=Edit3.Text;  
end;
```

Практика

Задача

Дана следующая информация об индивидуальных домах в некотором регионе: номер кадастрового дела, Фамилия И.О. собственника, адрес дома, общая площадь (кв.м), дата последней уплаты налога на недвижимость. Составить программу, которая: 1) находит собственников, не уплативших налог за предыдущий год; 2) выводит информацию о домах площадью более 100 кв.м. Предусмотреть возможности ввода исходных данных с клавиатуры и файла на диске. Вывод предусмотреть в двух вариантах: файл либо принтер.

Для решения задачи предлагается создать три окна



Компоненты

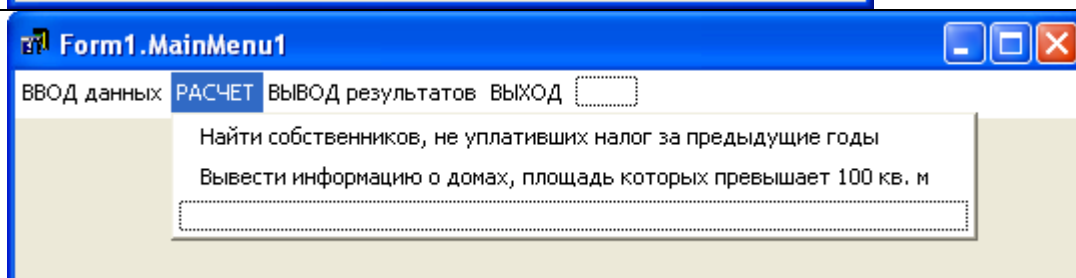
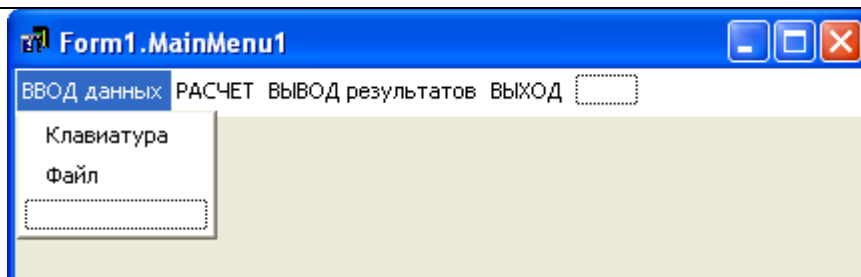
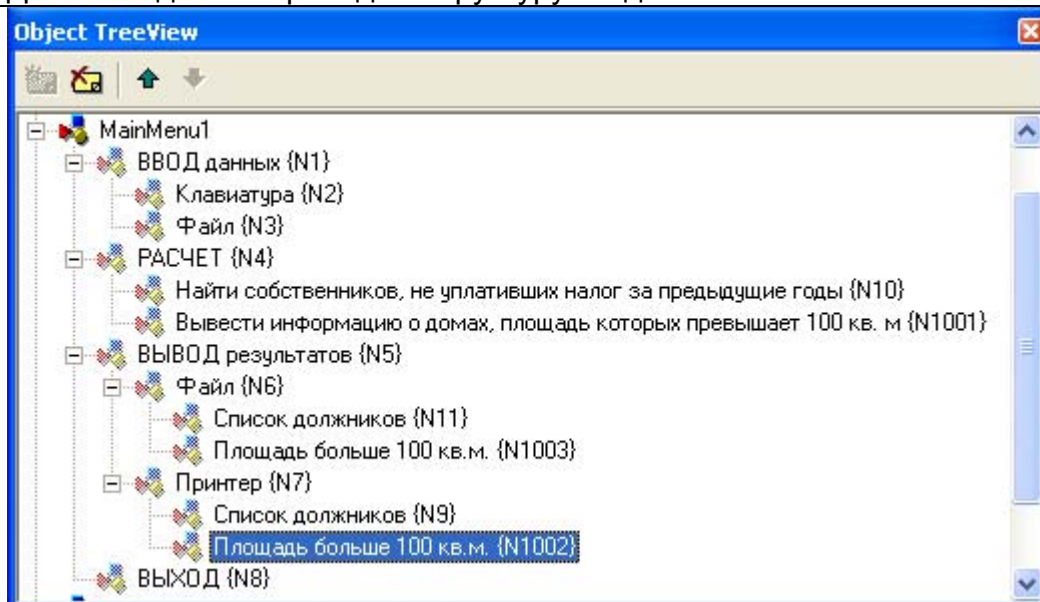
Имя Компонента	Свойства компонента	Значение	Назначение
Form1	Caption	Учет индивиду- альных домов	Заголовок глав- ной формы
Label1	Caption	Для перехода в	Справочная ин-

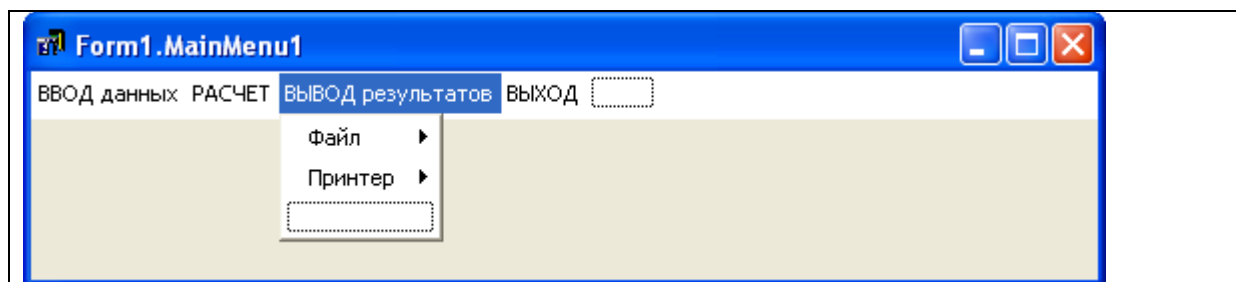
	Visible	новую ячейку таблицы использовать клавишу «ENTER». При вводе данных в последнюю ячейку строки нажать «ENTER» и клавишу «СТРЕЛКА ВНИЗ» для перехода в другую строку False	формация для пользователя программы Строка подсказки невидима
Form1.StringGrid1	ColCount	6	Количество столбцов по умолчанию
	FixedCol	1	Количество фиксированных слева столбцов
	FixedRow	1	Количество фиксированных сверху строк
	Options: goFixedVertLine	True	Опции: Разметка зафиксированных вертикальных линий
	goFixedHorzLine	True	Разметка зафиксированных горизонтальных линий
	goVertLine	True	Разметка вертикальных линий внутри таблицы
	goHorzLine	True	Разметка горизонтальных линий внутри таблицы
	goEditing	True	Признак допустимости редактирования содержимого ячеек таблицы. True — редактирование разрешено. False — запрещено
	goTabs	True	Разрешает использование кла-

	goAlwaysShowEditor	True	виши Tab для перемещения в другую ячейку
	ScrollBars	ssBoth	Признак нахождения компонента в режиме редактирования Выводятся обе полосы прокрутки
Form1.MainMenu	N1 N2 N3 N4 N5 N6 N7 N8 N10 N1001 N11 N1003 N9 N1002	Ввод данных Клавиатура Файл Расчет Вывод результатов Файл Принтер Вывод Найти собственников, не уплативших налог за предыдущие годы Вывести информацию о домах, площадь которых превышает 100 кв. м Список должников Площадь больше 100 кв.м Список должников Площадь больше 100 кв.м	Названия пунктов меню
OpenDialogs1	Name	Od1	Имя компонента
SaveDialog1	Name	SaveDialog1	Имя компонента
Form2	Caption	Дома с площадью более 100 кв.м	Заголовок формы
Form2.StringGrid1	Свойства аналогичны свойствам Form1.StringGrid1 (см. выше)		
Label1	Caption		Поле вывода результата
Form2.Button1	Caption	Закреть	Кнопка для закрытия окна “Дома с площадью более 100

			кв.м ”
Form3	Caption	Должники	Заголовок формы
Form3.StringGrid1	Свойства аналогичны свойствам Form1.StringGrid1 (см. выше)		
Label1	Caption	Список собственников, не уплативших налог на недвижимость в предыдущие годы, включая прошлый	Справочная информация для пользователя программы
Label2	Caption		Поле для вывода результата
Form3.Button1	Caption	Закреть	Кнопка для закрытия окна “Должники”

Для наглядности приведем структуру создаваемого главного меню:





Переменные и массивы

Обозначение в программе	Содержание	Тип
L	количество домов с площадью более 100 кв.м	целый
M	количество должников	целый
Delo_N	массив номеров кадастровых дел	строковый длиной 4
Famil	массив собственников индивидуальных домов	строковый длиной 20
Adres	массив адресов индивидуальных домов	строковый длиной 30
S	массив площадей индивидуальных домов	вещественный
D	массив дат последней уплаты налога на недвижимость	Объект типа дата
n	Счетчик объектов в файле или таблице	целый
F	Файловая переменная	текстовый
NameFile	Имя файла	строковый длиной 20
DA	День уплаты налога	Целый беззнаковый
M	Месяц уплаты налога	Целый беззнаковый
Y	Год уплаты налога	Целый беззнаковый
i	Счетчик строк, управляющая переменная в цикле	Целый
TextToPrint	Файловая переменная для вывода на принтер	Системный текст

Текст модуля

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms, Dialogs, Menus, Grids, DateUtils, StdCtrls, Unit2, Unit3, Printers;
// подключаются модули, связанные с 2-ой и 3-ей формами,
// библиотека функций и процедур работы с принтером
type
  TForm1 = class(TForm)
    MainMenu1: TMainMenu;

```

```

N1: TMenuItem;
N2: TMenuItem;
N3: TMenuItem;
N4: TMenuItem;
N5: TMenuItem;
N6: TMenuItem;
N7: TMenuItem;
N8: TMenuItem;
N10: TMenuItem;
N1001: TMenuItem;
StringGrid1: TStringGrid;
od1: TOpenDialog;
SaveDialog1: TSaveDialog;
N9: TMenuItem;
N1002: TMenuItem;
N11: TMenuItem;
N1003: TMenuItem;
Label1: TLabel;
    procedure N8Click(Sender: TObject);
    procedure N2Click(Sender: TObject);
    procedure N3Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure StringGrid1KeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
    procedure N10Click(Sender: TObject);
    procedure N1001Click(Sender: TObject);
    procedure N11Click(Sender: TObject);
    procedure N1003Click(Sender: TObject);
    procedure N9Click(Sender: TObject);
    procedure N1002Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
L:Integer;// количество домов с площадью более 100 кв.м
M:Integer;// количество должников
Delo_N:Array[1..1000] of String[4];// массив номеров кадастровых дел
Famil :Array[1..1000] of String[20]; ];{ массив собственников
индивидуальных домов}
Adres :Array[1..1000] of String[30];{ массив адресов индивидуальных
домов}
S:Array[1..1000] of Real;// массив площадей индивидуальных домов

```

```

D:Array[1..1000] of TDate; { массив дат последней уплаты налога на не-
движимость}
end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
// процедура закрывает главное окно — ВЫХОД
procedure TForm1.N8Click(Sender: TObject);
begin
Form1.Close;
end;
//активизация ввода с клавиатуры
procedure TForm1.N2Click(Sender: TObject);
begin
N2.Checked := True;// ставится галочка напротив пункта Клавиатура
Label1.Visible:=True;// поле с подсказкой становится видимым
StringGrid1.SetFocus;// фокус переносится в таблицу StringGrid
StringGrid1.Options := StringGrid1.Options + [goEditing];
// в StringGrid добавляется свойство,
// разрешающее редактирование ячеек
end;
// процедура ввода исходных данных из файла на диске
procedure TForm1.N3Click(Sender: TObject);
Var
  n:integer;// счетчик объектов в файле
  F:TextFile;// файловая переменная текстового типа
  NameFile:String[20];// имя файла
  DA,M,Y:Word; { день, месяц, год последней уплаты
налога на недвижимость, в файле эти числа вводятся через пробел }
begin
N3.Checked:=True;// ставится галочка напротив пункта «Файл»
if not od1.Execute then Exit;// если в открывшемся списке
// не выбрано имя файла, то прекращается выполнение процедуры
NameFile := od1.FileName; { переменной имя_файла присваивается
имя, выбранное из списка}
AssignFile(F,NameFile);// процедура связывает файловую переменную
// и конкретный файл
Reset(F);// процедура открывает файл для чтения
// счетчик объектов инициализируется единицей
n:=1;
{ пока не достигнут конец файла, из файла читаются построчно:

```

номер кадастрового дела, фамилия собственника, адрес дома, площадь, дата в виде трех чисел на одной строке. Данные одновременно выводятся в таблицу в главном диалоговом окне}

```
While Not(EoF(F)) Do // пока не достигнут конец файла, делать
  Begin
    Readln(F,Delo_N[n]); // из файла читается номер кадастрового дела
    StringGrid1.Cells[1,n]:=Delo_N[n]; { номер кадастрового дела выво-
    дится в таблицу на 1 форме}
    Readln(F,Famil[n]); // из файла читается фамилия собственника
    StringGrid1.Cells[2,n]:=Famil[n];// фамилия выводится в форму на экран
    Readln(F,Adres[n]); // читается адрес
    StringGrid1.Cells[3,n]:=Adres[n];// адрес выводится в форму на экран
    Readln(F,S[n]);// читается площадь
    StringGrid1.Cells[4,n]:=FloatToStr(S[n]); {площадь выводится в таб-
    лицу на экране}
    Readln(F,DA,M,Y);D[n]:=EncodeDate(Y,M,DA);{преобразование трех
    чисел в объект типа дата – массив дат}
    StringGrid1.Cells[5,n]:=DateToStr(D[n]);// дата выводится в таблицу
    n:=n+1;// счетчик объектов увеличивается на 1
    {если количество строк в таблице на форме меньше n, то ему при-
    сваивается значение счетчика объектов }
    if StringGrid1.RowCount < n then
      StringGrid1.RowCount := n;
    End;
    {после того как достигнут конец файла, файл закрывается}
  CloseFile(F);
  end;
  {процедура выполняется при появлении диалогового окна (формы) на
  экране, связана с событием — создание формы}
  procedure TForm1.FormCreate(Sender: TObject);
  Var
    n:Integer; // счетчик объектов в таблице
  begin
    // заполнение шапки таблицы нулевой строки
    StringGrid1.Cells[0,0]:='Номер п/п';
    StringGrid1.Cells[1,0]:='Номер кадастрового дела';
    StringGrid1.Cells[2,0]:='Фамилия И.О. собственника';
    StringGrid1.Cells[3,0]:='Адрес';
    StringGrid1.Cells[4,0]:='Общая площадь (кв.м)';
    StringGrid1.Cells[5,0]:='Дата последней уплаты налога на недвижимость';
    //цикл нумерации строк таблицы
    For n:=1 to 1000 Do
      StringGrid1.Cells[0,n]:=IntToStr(n);
```

end;

{Процедура, связанная с событием, нажатие клавиши в ячейке таблицы. Проверяется правильность нажатия клавиш при вводе — «ENTER» (для перехода в ячейку справа) и последовательно клавиш «ENTER» и «СТРЕЛКА ВНИЗ» (для перехода на следующую строку). Здесь же происходит считывание данных из ячеек таблицы в массивы. Такая технология ввода позволила организовать ввод для заранее неизвестного количества данных.}

```
procedure TForm1.StringGrid1KeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);
```

```
Begin
```

```
// если нажата клавиша «СТРЕЛКА ВНИЗ»
```

```
if Key = VK_DOWN then begin
```

```
// то если текущий номер строки таблицы меньше предыдущего
```

```
// значения количества строк в таблице, тогда ВЫХОД
```

```
if StringGrid1.Row < pred(StringGrid1.RowCount) then Exit;
```

```
// количество строк в таблице увеличивается на единицу
```

```
StringGrid1.RowCount := StringGrid1.RowCount + 1;
```

```
StringGrid1.Col := 1; // переход в начало строки
```

```
// если в первую ячейку текущей строки ничего не введено,
```

```
// то добавленная строка убирается
```

```
if (Length(StringGrid1.Cells[1,StringGrid1.Row]) = 0) then begin
```

```
StringGrid1.RowCount:=pred(StringGrid1.RowCount);Exit;
```

```
end;
```

```
end;
```

```
// если нажата клавиша «ENTER», то
```

```
if (Key = VK_RETURN) then begin
```

```
case StringGrid1.Col of
```

```
// если номер столбца таблицы равен 1 и если ячейка не пуста,
```

```
1 : if (Length(StringGrid1.Cells[1,StringGrid1.Row]) > 0)
```

```
// тогда содержимое ячейки с индексом равным номеру текущей строки
```

```
// помещается в массив «Номер кадастрового дела»
```

```
then
```

```
Delo_N[StringGrid1.Row] := StringGrid1.Cells[1,StringGrid1.Row]
```

```
// иначе количеству строк в таблице
```

```
// присваивается предыдущее значение и выход, т.е. программа будет
```

```
// ждать заполнения первой колонки в этой строке
```

```
else begin
```

```
StringGrid1.RowCount:=pred(StringGrid1.RowCount);Exit;
```

```
end;
```

```
// если текущий номер колонки таблице равен 2, то
```

```
// в массив собственников под номером равным номеру текущей строки
```

```
// таблицы, помещается содержимое ячейки, стоящей на пересечении
```



```

// второго столбца и текущей строки
    2 : Famil[StringGrid1.Row] := StringGrid1.Cells[2,StringGrid1.Row];
// аналогично предыдущему вводится адрес
    3 : Adres[StringGrid1.Row] := StringGrid1.Cells[3,StringGrid1.Row];
// если четвертая колонка не пуста, то строка символов преобразуется
// в вещественное число и помещается в массив площадей домов
    4 : if Length(StringGrid1.Cells[4,StringGrid1.Row]) > 0 then
S[StringGrid1.Row]:=StrToFloat(StringGrid1.Cells[4,StringGrid1.Row])
// иначе в массив вводится число 0.00
        else
            S[StringGrid1.Row] := 0.00;
// если пятая колонка не пуста, то строка символов преобразуется в дату
// и помещается в массив дат
    5 : if Length(StringGrid1.Cells[5,StringGrid1.Row]) > 0 then
D[StringGrid1.Row] := StrToDate(StringGrid1.Cells[5,StringGrid1.Row])
// иначе в массив дат помещается 0
        else
            D[StringGrid1.Row] := 0;
end;
// если текущий номер колонки меньше 5, то он увеличивается на 1
if StringGrid1.Col < 5 then
    StringGrid1.Col := StringGrid1.Col + 1
end;

end;
{ процедура, связанная с пунктом меню «Найти собственников, не уплативших налог за предыдущие годы»}
procedure TForm1.N10Click(Sender: TObject);
Var
i,n:integer;
begin
// ставится галочка напротив пункта меню
// «Найти собственников, не уплативших налог за предыдущие годы»
N10.Checked:=True;
// подписывается шапка таблицы
Form3.StringGrid1.Cells[0,0]:='Номер по п/п';
Form3.StringGrid1.Cells[1,0]:='Фамилия И.О. собственника';
Form3.StringGrid1.Cells[2,0]:='Дата последней уплаты налога на недвижимость';
i:=0;// — номер строки в таблице
{ В цикле последняя дата уплаты налога собственником дома сравнивается
с текущей датой, установленной на компьютере. Если разница превышает
1 год, собственник является должником. Фамилии должников выводятся в

```

таблицу в третьей форме. Если дата неизвестна, то выводится сообщение: «нет сведений»}

```
For n := 1 to pred(StringGrid1.RowCount) Do
// функция YearsBetween(D1,D2) находит разницу между двумя датами
IF YearsBetween(Date,d[n]) >= 1 then Begin
Inc(i); // i:=i+1;
Form3.StringGrid1.Cells[0,i]:=IntToStr(i);
Form3.StringGrid1.Cells[1,i]:=Famil[n];
if D[n]=0 then Form3.StringGrid1.Cells[2,i]:='нет сведений'
else
Form3.StringGrid1.Cells[2,i]:=DateToStr(D[n]);
Form3.StringGrid1.RowCount:=Form3.StringGrid1.RowCount+1
end;
// вывод количества должников
M:=i;
Form3.Label2.Caption:='Всего: '+ IntToStr(M)+' должников';
//показать форму 3
Form3.Show;
end;
{процедура, связанная с пунктом меню «Вывести информацию о до-
мах, площадь которых превышает 100 кв.м»}
procedure TForm1.N1001Click(Sender: TObject);
Var n,i:Integer;
Begin
// поставить галочку напротив выбранного пункта меню
N1001.Checked:=True;
Form2.StringGrid1.Cells[0,0]:='Номер по п/п';
Form2.StringGrid1.Cells[1,0]:='Номер кадастрового дела';
Form2.StringGrid1.Cells[2,0]:='Фамилия И.О. собственника';
Form2.StringGrid1.Cells[3,0]:='Адрес';
Form2.StringGrid1.Cells[4,0]:='Общая площадь (кв.м)';
Form2.StringGrid1.Cells[5,0]:='Дата последней уплаты налога на недвижимость';
i:=0;
{ В цикле производится сравнение площадей всех домов со 100 кв.м.
Если площадь дома превышает 100 кв.м, то вся информация о нем
выводится в таблицу на форме 2}
For n:=1 to pred(Form1.StringGrid1.RowCount) Do
IF S[n]>100 Then Begin
Inc(i); // i:=i+1;
Form2.StringGrid1.Cells[0,i]:=IntToStr(i);
Form2.StringGrid1.Cells[1,i]:=Delo_N[n];
Form2.StringGrid1.Cells[2,i]:=Famil[n];
Form2.StringGrid1.Cells[3,i]:=Adres[n];
```

```

Form2.StringGrid1.Cells[4,i]:=FloatToStrF(S[n],ffFixed,10,2)+' кв.м';
if D[n]=0 then Form2.StringGrid1.Cells[5,i]:='нет сведений'
else
    Form2.StringGrid1.Cells[5,i]:=DateToStr(D[n]);
Form2.StringGrid1.RowCount:=Form2.StringGrid1.RowCount+1
end;
L:=i;
Form2.Label1.Caption:='Всего домов площадью более 100 кв.м.'+IntToStr(L);
Form2.Show;
end;
{процедура, связанная с пунктом меню «Вывод в файл» — список
должников}
procedure TForm1.N11Click(Sender: TObject);
Var f:TextFile;
    NameFile:String;
    i:Integer;
begin
    N11.Checked:=True;
    {Если из панели диалога не выбрано или не задано имя файла сохранения,
то выход. Т.е. панель диалога будет находится на экране, пока пользова-
тель не задаст имя файла.}
    if not SaveDialog1.Execute then Exit;
// переменной ИмяФайла присваивается имя файла,
// заданное в панели диалога
    NameFile:=SaveDialog1.FileName;
// устанавливается физическая связь между файловой переменной
// и конкретным файлом на внешнем носителе
    AssignFile(f,NameFile);
// открыть файл для записи
    Rewrite(f);
// в следующем фрагменте программы информация о должниках
// из формы 3 записывается в файл
    Writeln (f,'список должников:');
    Writeln (f,'Номер п/п ','Фамилия И.О. собственника',
' Дата последней уплаты налога на недвижимость');
    For i:=1 To M do
Writeln (f,Form3.StringGrid1.Cells[0,i]:4,Form3.StringGrid1.Cells[1,i]:30,' ',
Form3.StringGrid1.Cells[2,i]:30);
// закрытие файла
    CloseFile(f);
end;

```

{процедура, связанная с пунктом меню «Вывод в файл» — список домов с площадью более 100 кв.м}

```
procedure TForm1.N1003Click(Sender: TObject);
  Var f:TextFile;
      NameFile:String;
      i:Integer;
  begin
    N1003.Checked:=True;
  // если не задано имя файла в панели диалога,
  // панель диалога остается на экране
    if not SaveDialog1.Execute then Exit;
  // переменной ИмяФайла присваивается имя файла,
  // заданное в панели диалога
    NameFile:=SaveDialog1.FileName;
  // устанавливается физическая связь между файловой переменной
  // и конкретным файлом на внешнем носителе
    AssignFile(f,NameFile);
  // открыть файл для записи
    Rewrite(f);
  // в следующем фрагменте программы информация о домах с площадью
  // более 100 кв.м из формы 2 записывается в файл
    Writeln (f,'Дома с площадью более 100 кв. м:');
    Writeln (f,'Номер п/п ','Номер кадастрового дела ',
'Фамилия И.О. собственника ', ' Адрес ',
'Общая площадь (кв.м) ', 'Дата последней уплаты налога на недвижимость');
    For i:=1 To L do
      Writeln (f,Form2.StringGrid1.Cells[0,i]:4,Form2.StringGrid1.Cells[1,i]:20,' ',
        Form2.StringGrid1.Cells[2,i]:30,' ',Form2.StringGrid1.Cells[3,i]:30,
        Form2.StringGrid1.Cells[4,i]:16,' ',Form2.StringGrid1.Cells[5,i]:20 );
      CloseFile(f);
  end;
```

{процедура, связанная с пунктом меню «Вывод на принтер» — список должников}

```
procedure TForm1.N9Click(Sender: TObject);
  Var TextToPrint:System.Text;
      i:integer;
  begin
    N9.Checked:=True;
  // устанавливается физическая связь с печатающим устройством
    AssignPrn(TextToPrint);
  // открытие для вывода
    Rewrite(TextToPrint);
```

```

// в следующем фрагменте осуществляется вывод списка должников
// из формы 3 на принтер
  Writeln (TextToPrint,'список должников:');
  Writeln (TextToPrint,'Номер п/п ','Фамилия И.О.',
  ' Дата последней уплаты ');
  Writeln (TextToPrint,','собственника',
  ' налога на недвижимость ');
  For i:=1 To M do
  Writeln (TextToPrint,Form3.StringGrid1.Cells[0,i]:4,
  Form3.StringGrid1.Cells[1,i]:30,' ',Form3.StringGrid1.Cells[2,i]:30);
  System.Close(TextToPrint);
  end;

  {процедура, связанная с пунктом меню «Вывод на принтер» — список
домов с площадью более 100 кв.м}
  procedure TForm1.N1002Click(Sender: TObject);
  Var TextToPrint:System.Text;
      i:integer;
  begin
  N1002.Checked:=True;
  // ассоциация с принтером
  AssignPrn(TextToPrint);
  // открытие для вывода
  Rewrite(TextToPrint);
// в следующем фрагменте осуществляется вывод списка
// домов с площадью более 100 кв.м из формы 2 на принтер
  Writeln (TextToPrint,'Дома с площадью более 100 кв. м:');
  Writeln (TextToPrint,'N п/п','Номер кад. дела ',
  'Фамилия И.О.собств.',' Адрес ',
  'Общ. площ.(кв.м)','Дата посл. Уплаты налога на недвижимость');
  For i:=1 To L do
  begin
  Writeln (TextToPrint,Form2.StringGrid1.Cells[0,i]:4,
  Form2.StringGrid1.Cells[1,i]:20,' ',Form2.StringGrid1.Cells[2,i]:20,' ',
  Form2.StringGrid1.Cells[3,i]:40,Form2.StringGrid1.Cells[4,i]:16,' ',
  Form2.StringGrid1.Cells[5,i]:20 );
  end;
  System.Close(TextToPrint);
  end;

  end.
  {модуль Unit2 связан с формой Form2, куда выводится список домов с
площадью превышающей 100 кв.м}

```

```

unit Unit2;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms, Dialogs, Grids, StdCtrls;
type
  TForm2 = class(TForm)
    StringGrid1: TStringGrid;
    Button1: TButton;
    Label1: TLabel;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form2: TForm2;
implementation
{$R *.dfm}
procedure TForm2.Button1Click(Sender: TObject);
begin
  Form2.Close;
end;
end.
{модуль Unit3 связан с формой Form3, куда выводится список должни-
ков по уплате налога на недвижимость}
unit Unit3;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls, Grids;
type
  TForm3 = class(TForm)
    Label1: TLabel;
    StringGrid1: TStringGrid;
    Button1: TButton;
    Label2: TLabel;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }

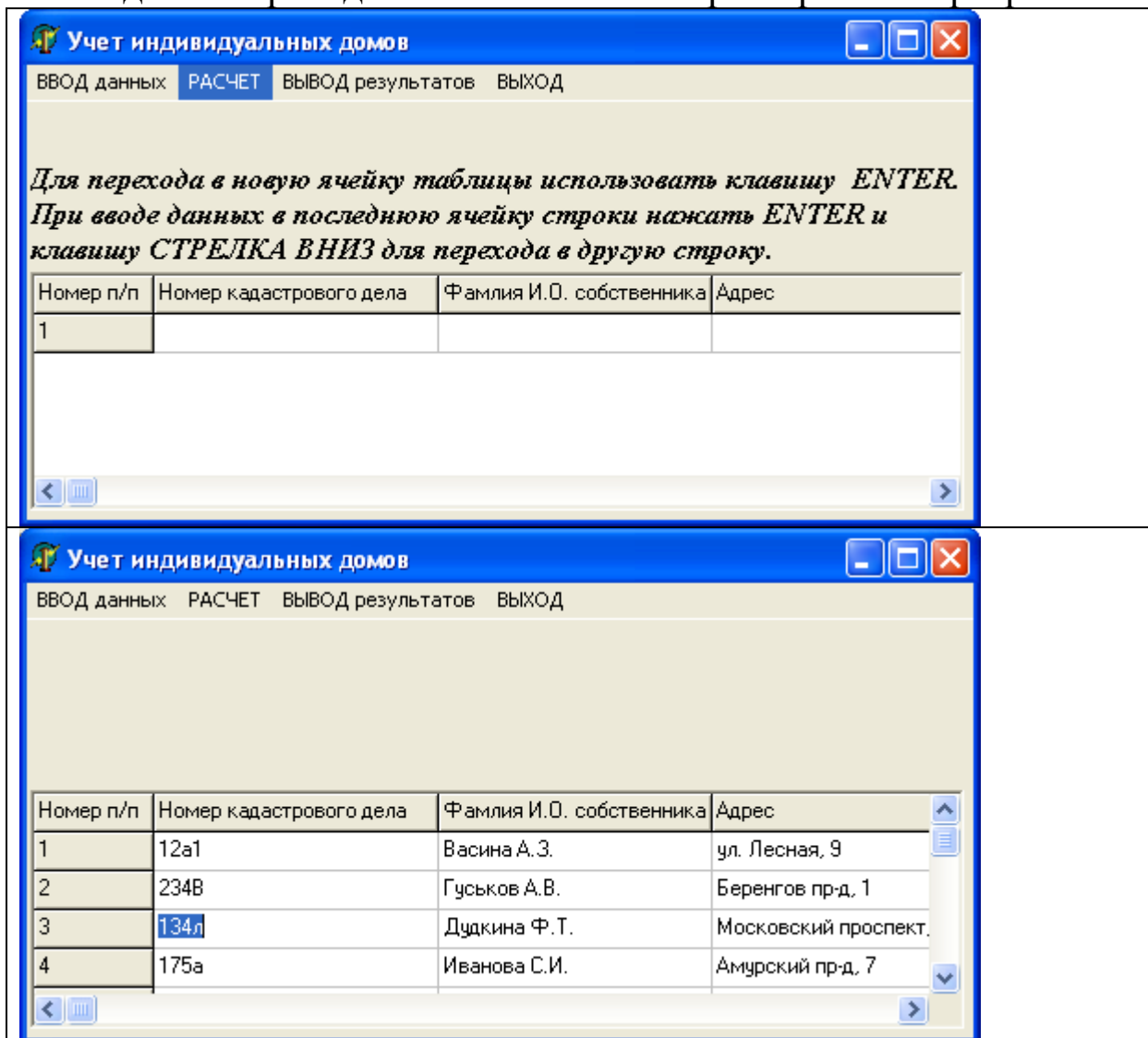
```

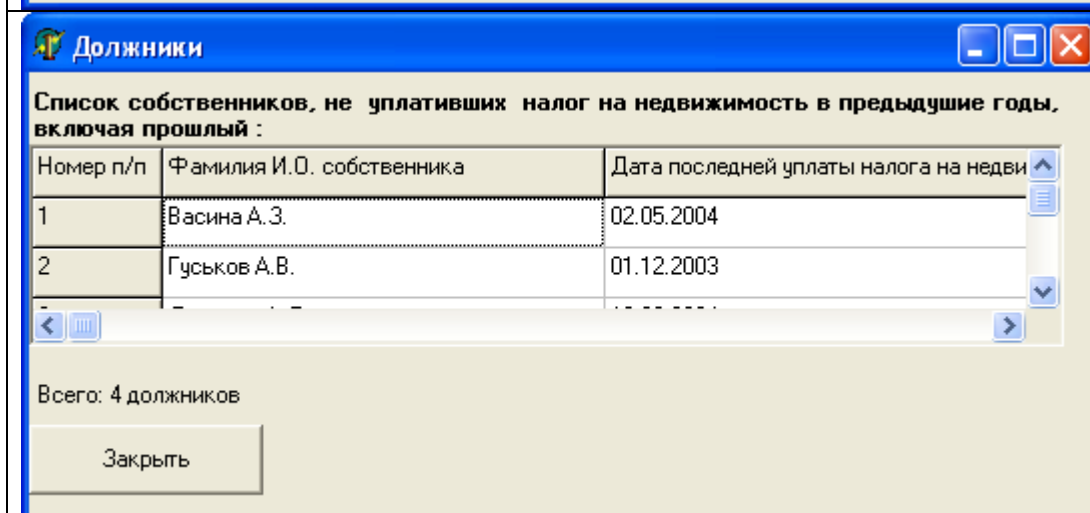
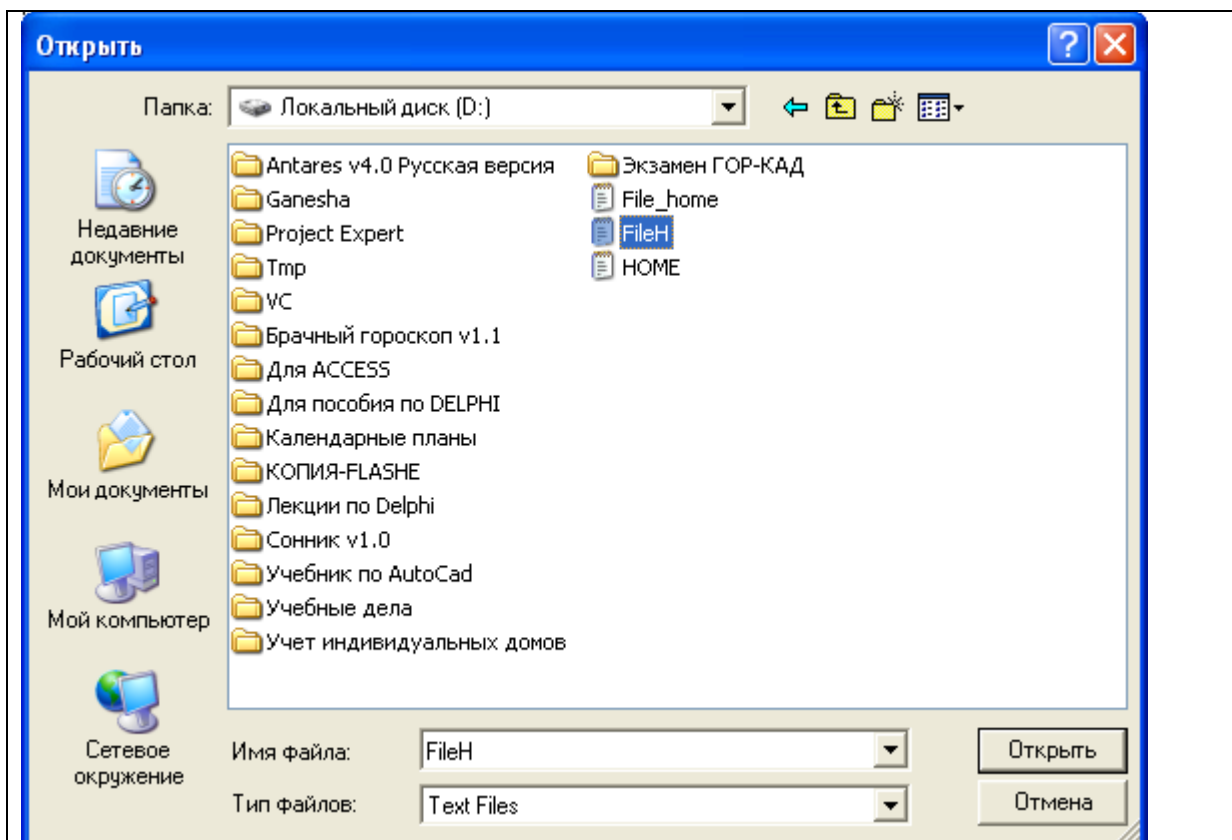
```

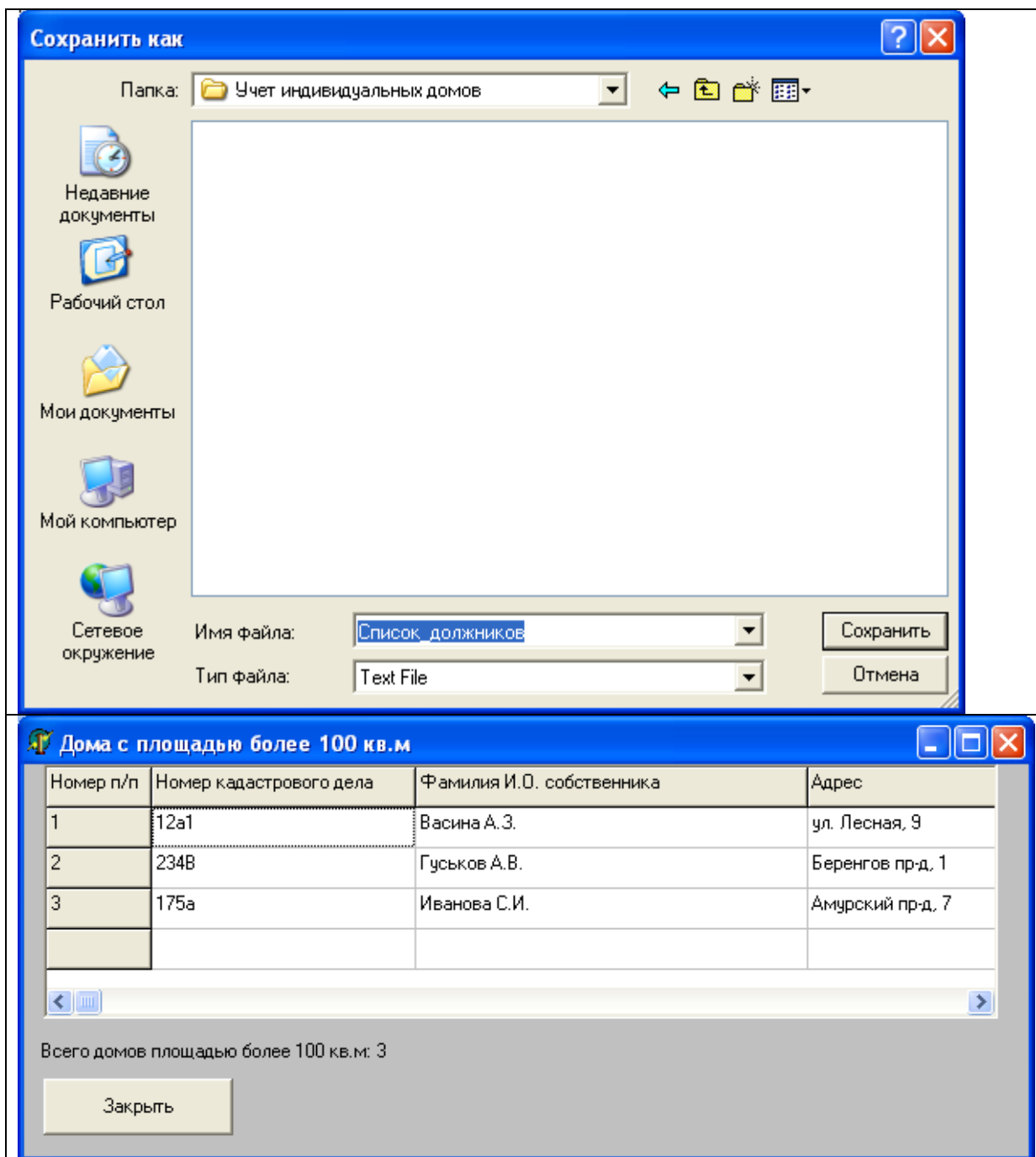
end;
var
    Form3: TForm3;
implementation
{$R *.dfm}
procedure TForm3.Button1Click(Sender: TObject);
begin
    Form3.Close;
end;
end.

```

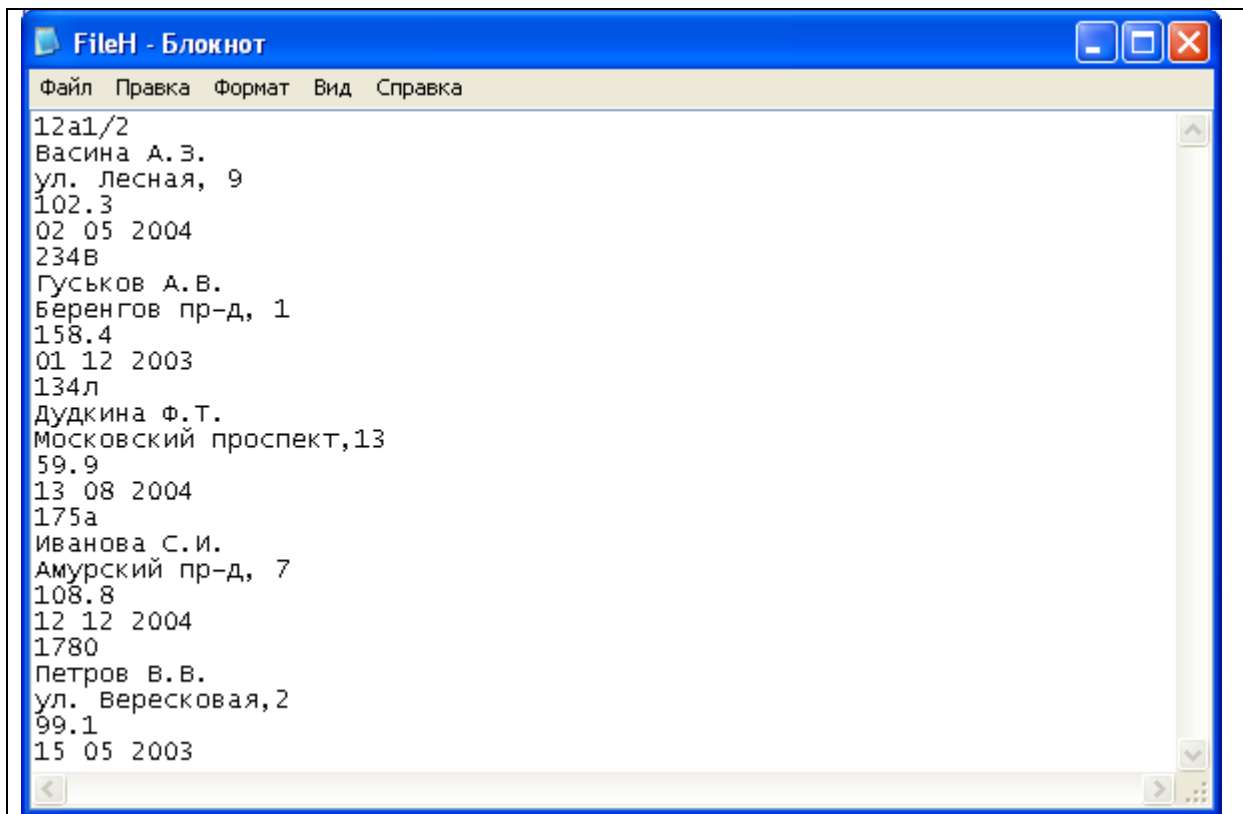
Вид некоторых диалоговых окон во время работы программы





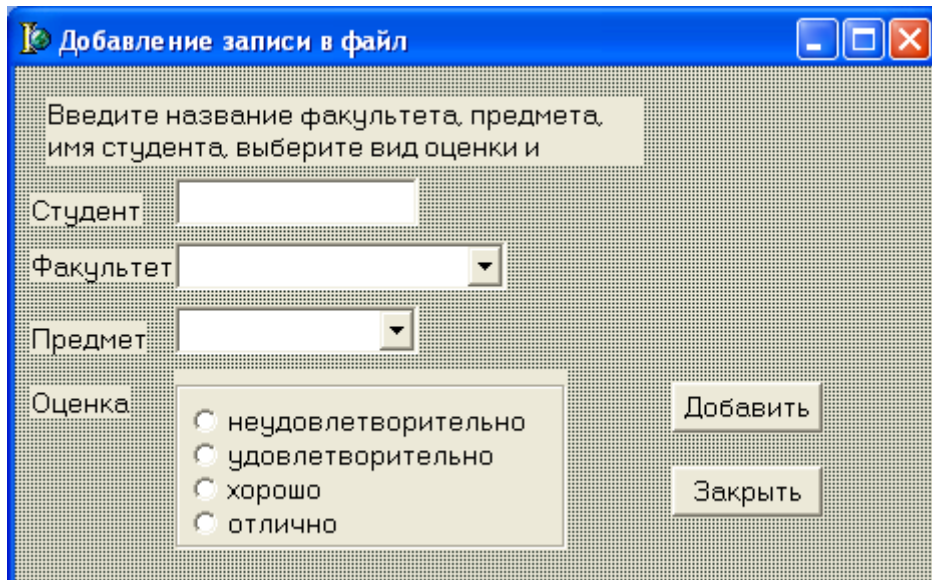


Файл с исходной информацией удобнее подготовить с помощью стандартной программы «Блокнот». Каждое поле записи набирается на отдельной строке. Десятичный разделитель — точка. Дата уплаты налога в файле набирается в виде трех чисел на одной строке через пробел. Для приведенного выше примера файл с исходными данными выглядит так:



Задача

Составить программу, которая записывает в файл введенные пользователем данные о результатах экзаменов, формируя таким образом простую базу данных. Исходные данные вводятся в поля диалогового окна и сохраняются в файле, компонентами которого являются записи типа TOfenka.



unit apprec_;

.....

// тип оценки

```

TKind = (NeUd, Ud, Hor,Otl);

// запись файла
TOzenka=record
Facultet: string[20]; // факультет
Predmet: string[20]; // предмет
person: string[40]; // студент
Oz: TKind; // оценка
end;

var
Form1: TForm1;
f: file of TOzenka; // файл записей – база данных

implementation

{$R *.DFM}

// активизация формы
procedure TForm1.FormActivate(Sender: TObject);
var
  resp : word; // ответ пользователя
begin
  AssignFile(f, 'D:\Ozenki.db');
  {$I-}
  Reset(f); // открыть файл
  Seek( f, FileSize(f)); // указатель записи в конец файла
  {$I+}
  if IOResult = 0
  then button1.enabled:=TRUE // теперь кнопка «Добавить» доступна
  else
  begin
    resp:=MessageDlg('Файл базы данных не найден.'+
' создать новую БД?',mtInformation,[mbYes,mbNo],0);
    if resp = mrYes then
    begin
      {$I-}
      rewrite(f);
    end;
  end;
end;

```

```

{$I+}
if IOResult = 0
then button1.enabled:=TRUE
else ShowMessage('Ошибка создания БД. ');
end;
end;
end;

```

```

// щелчок на кнопке «Добавить»
procedure TForm1.Button1Click(Sender: TObject);

```

```

var
  Ozenka: TOzenka;
  oz :TKind;
begin
  with Ozenka do
  begin
    Facultet := ComboBox1.Text;
    Predmet:= ComboBox2.Text;
    person := Edit1.Text;
    case RadioGroup1.ItemIndex of
    0: oz := NeUd;
    1: oz:= Ud;
    2: oz := Hor;
    3: oz := Otl;
    end;
  end;
  write(f,Ozenka ); // записать содержимое полей записи в файл
end;

```

```

// завершение работы программы
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  CloseFile( f ); // закрыть файл
end;

```

```

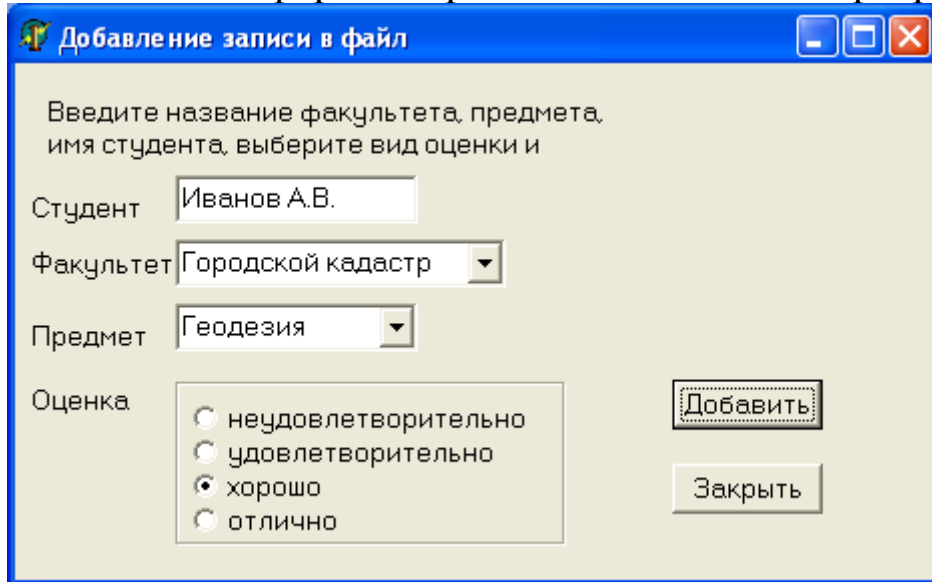
procedure TForm1.Button2Click(Sender: TObject);
begin

```

```
Form1.Close  
end;
```

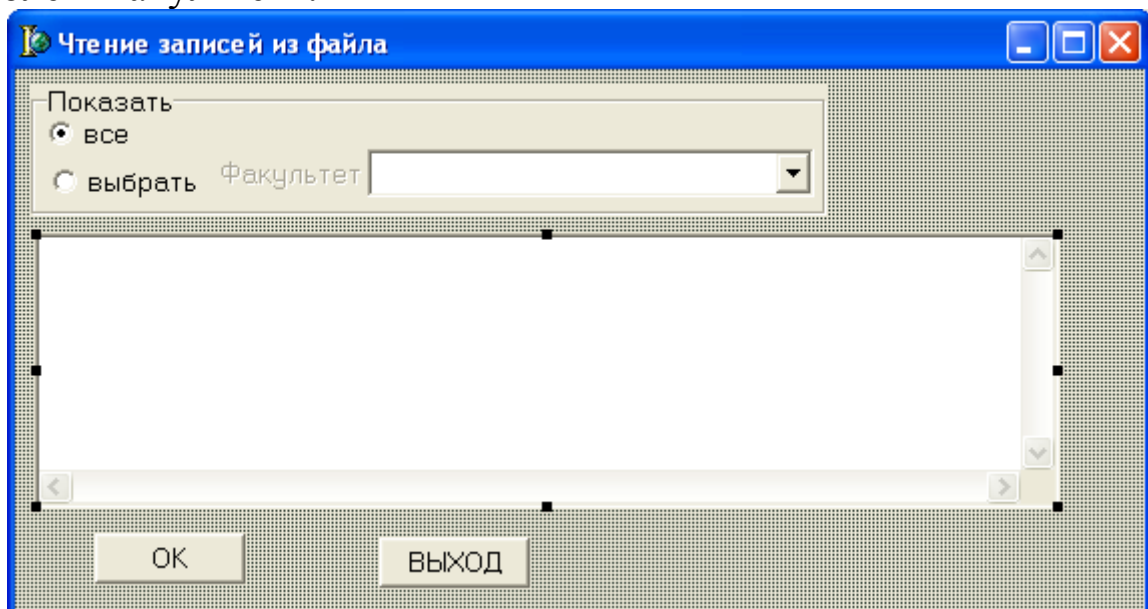
```
end.
```

Так выглядит форма в процессе выполнения программы:



Задача

Составить программу, которая читает и обрабатывает данные, записанные в файл программой *Добавление записи в файл*, и, в зависимости от того, какой из переключателей — *все* или — *выбрать* выбран, выводит список оценок, полученных студентами всех факультетов или факультета, название которого введено в поле «Факультет».



```
unit rdrec_;
```

```

.....
var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.Button1Click(Sender: TObject);
type
  // тип оценки
  TKind = (NeUd,Ud,Hor,Otl);

  // запись файла
  TOzenka = record
    Facultet:string[20];
    Predmet:string[20];
    person:string[40];
    kind:TKind;
  end;

var
  f: file of TOzenka; // файл записей
  rec: TOzenka; // запись, прочитанная из файла
  n: integer; // количество записей, удовлетворяющих запросу
  st: string[80];

begin
  AssignFile(f,'D:\ozenki.db');
  {$I-}
  Reset(f); //открыть файл для чтения
  {$I-}
  if IOResult <> 0 then
  begin
    ShowMessage('Ошибка открытия файла БД. ');
    Exit;
  end;
  Memo1.Clear; // очистить список поля Мемо

```

```

// обработка БД
if RadioButton2.Checked then
Memo1.Lines.Add('*** ' + ComboBox1.Text + '***');

n := 0;
while not EOF(f) do
begin
read(f, rec); // прочитать запись
if RadioButton1.Checked or
(rec.facultet = ComboBox1.Text) then
begin
n := n + 1;
st := rec.person+ ', ' + rec.predmet;
if RadioButton1.Checked then
st := st + ', ' + rec.facultet;
case rec.kind of
NeUd: st := st+ ', неудовлетворительно ';
Ud: st := st+ ', удовлетворительно';
Hor: st := st+ ', хорошо';
Otl: st := st+ ', отлично';
end;
Memo1.Lines.Add(st);
end;
end;
CloseFile(f);
if n = 0 then
ShowMessage('В БД нет запрашиваемой информации.');
```

```

end;

// переключатель «Выбрать»
procedure TForm1.RadioButton2Click(Sender: TObject);
begin
Label1.Enabled := True;
ComboBox1.Enabled := True;
// теперь поля Факультет доступно
ComboBox1.SetFocus; // курсор в поле «Факультет»
end;
```

```

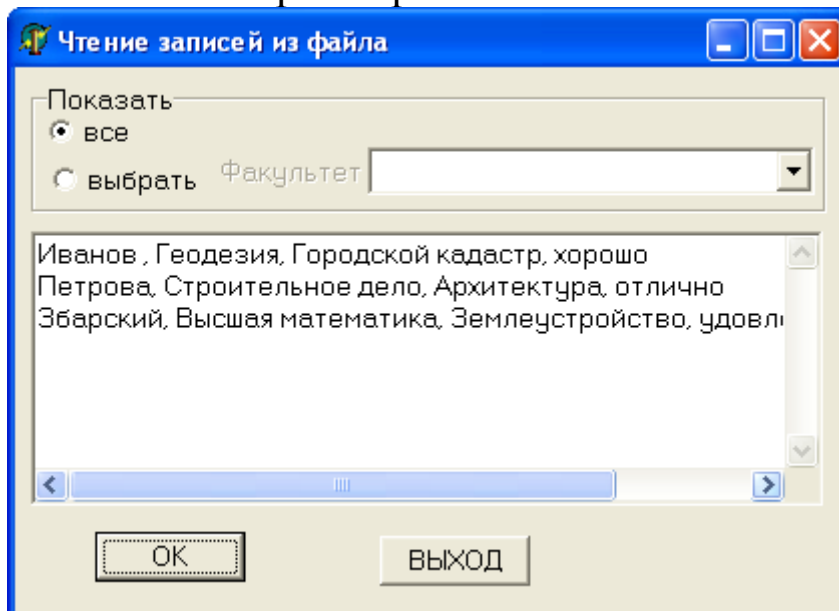
// переключатель «Все»
procedure TForm1.RadioButton1Click(Sender: TObject);
begin
  Label1.Enabled := False;
  ComboBox1.Enabled := False;
  // теперь поле «Факультет» не доступно
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  Form1.Close
end;

end.

```

Возможный вариант решения:



Задачи для самостоятельного решения

Группа А

№ 7.1. Подсчитать значение Q , вычисляемой по формуле

$$\sum_{i=1}^N Y_i$$

$$Q = \frac{\sum_{i=1}^N Y_i}{N!},$$

где Y_i — элементы массива Y с размерностью N . Массив Y ввести из файла на диске. Вывод обеспечить в трех вариантах: 1) на экран; 2) на печатающее устройство; 3) в файл на диске.

№ 7.2. Дан массив X , содержащий вещественные числа. Найти и записать на место $X(1)$ — наибольший элемент массива X , а на место $X(N)$ — наименьший элемент массива (X_1, X_2, \dots, X_N). Ввод исходного массива X осуществить из файла на диске. Вывод обеспечить в трех вариантах: 1) на экран; 2) на печатающее устройство; 3) в файл на диске.

№ 7.3 Дан массив $Q(N)$, состоящий из отрицательных и положительных чисел. Составить из Q два массива: A , содержащий только положительные числа; B , содержащий только отрицательные. Массив Q вводится из файла на диске. Вывести исходный массив и массивы, полученные в процессе преобразования. Вывод обеспечить в трех вариантах: 1) на экран; 2) на печатающее устройство; 3) в файл на диске.

№ 7.4. Даны натуральное число N , действительные числа X_1, X_2, \dots, X_n . Получить $(1+r)/(1+s)$, где r — сумма всех тех членов последовательности X_1, X_2, \dots, X_n , которые не превосходят 1, а s — сумма членов, больших 1. Вывести сначала массив X . Затем r , s . Ввод исходных данных осуществить из файла на диске. Вывод — по желанию пользователя программы либо на экран, либо на печать или в файл на диске.

№ 7.5. Дана последовательность из N различных чисел. Найти сумму чисел этой последовательности, расположенных между максимальным и минимальным числами (в сумму включить и оба этих числа). Последовательность ввести из файла на диске. Вывести исходную последовательность в виде таблицы с указанием порядкового номера числа в последовательности и самого числа и сумму чисел. Вывод предусмотреть в трех вариантах: 1) в файл на диске; 2) на экран; 3) на печатающее устройство.

№ 7.6. Даны: n — целое число ($n \geq 2$), последовательности действительных чисел : a_1, a_2, \dots, a_n . Получить:

$\min(a_1+a_2, a_2+a_3, \dots, a_{n-1}+a_n)$;

$\max(a_1, a_1a_2, a_1a_2a_3, \dots, a_1 \dots a_{n-1}a_n)$.

Ввод исходных данных осуществить из файла на диске. Вывод — по желанию пользователя программы либо на экран, либо на печать, либо в файл на диске.

№ 7.7. Дан файл f_1 , который содержит номера телефонов сотрудников учреждения: указывается фамилия сотрудника, его инициалы и номер телефона. Найти телефон сотрудника по его фамилии и инициалам. Вывод по желанию пользователя либо на экран, либо в файл на диске, либо на печать.

№ 7.8. Дан файл f , содержащий сведения о кубиках: размер каждого кубика (длина ребра в см), его цвет (красный, желтый, зеленый или синий) и материал (деревянный, металлический, картонный). Найти: а) количество кубиков каждого из перечисленных цветов и их суммарный объем;

б) количество деревянных кубиков с ребром 3 см и количество металлических кубиков с ребром, большим 5 см. Вывод по желанию пользователя либо на экран, либо в файл на диске, либо на печать.

№ 7.9. Сведения об автомобиле состоят из его марки, номера и фамилии владельца. Дан файл f , содержащий сведения о нескольких автомобилях. Найти: а) фамилии владельцев и номера автомобилей данной марки; б) количество автомобилей каждой марки. Вывод по желанию пользователя либо на экран, либо в файл на диске, либо на печать.

№ 7.10. Дан файл f , содержащий сведения о книгах. Сведения о каждой из книг — это фамилия автора, название и год издания.

а) Найти названия книг данного автора, изданных до 2002 года.

б) Определить, имеется ли книга с названием «Информатика». Если да, то сообщить фамилию автора и год издания. Если таких книг несколько, то сообщить имеющиеся сведения обо всех этих книгах. Вывод по желанию пользователя либо на экран, либо в файл на диске, либо на печать.

Группа Б

№ 7.11. В память компьютера вводятся по очереди координаты N точек. Определить, сколько из них попадает в кольцо с внутренним радиусом R_1 и внешним R_2 . Обе окружности имеют общий центр с координатами X_0 и Y_0 . Результаты обработки выдать либо на печать, либо на экран. Записать на дискету координаты точек, попавших в кольцо.

№ 7.12. Имеется таблица со среднесуточными температурами за период с 1.06.XX по 31.08.XX с точностью до 0,1 градуса. Необходимо определить и вывести на экран, а по просьбе пользователя и на печать, максимальную и минимальную среднесуточные температуры за указанный период, а также дни, в которые среднесуточная температура отличалась от максимальной или минимальной не более чем на 0,5 градуса. Исходные данные вводить с магнитного диска. Форма представления исходных данных на магнитном диске:

Июнь

1.....температура

2.....—

—.....—

—.....—

30.....—

Июль, август — аналогично.

Форма вывода исходных данных и результатов на экран и печать.

Среднесуточные температуры:

Июнь.

Дата температура Дата температура и т.д.

1.....—.....6.....— всего 6 пар столбцов

2.....—.....7.....—

3.....—.....8.....—

4.....—.....9.....—

5.....—.....10.....—

Июль, август — аналогично.

Максимальная температура XX градусов была:

в июне: число, число, и т.д.

в июле: число, число, и т.д.

в августе: число, число, и т.д.

Минимальная температура XX градусов была:

в июне: число, число, и т.д.

в июле: число, число, и т.д.

в августе: число, число, и т.д.

Ввод исходных данных осуществить из файла на диске. Вывод предусмотреть в трех вариантах: 1) в файл на диске; 2) на экран дисплея; 3) на печать.

№ 7.13. Дан файл f , содержащий сведения о веществах: указывается название вещества, его удельный вес и проводимость (проводник, полупроводник, изолятор).

а) Найти удельные веса и названия всех полупроводников.

б) Выбрать данные о проводниках и упорядочить их по убыванию удельных весов.

Составить программу. Вывод по желанию пользователя либо на экран, либо в файл на диске, либо на печать.

№ 7.14. Даны два файла f_1 и f_2 . Файл f_1 — это инвентарный файл, содержащий сведения о том, сколько изделий разных видов продукции хранится на складе (вид продукции задается его порядковым номером). Файл f_2 — это вспомогательный файл, содержащий сведения о том, на сколько уменьшилось или увеличилось количество изделий по некоторым видам продукции. Вспомогательный файл может содержать несколько сообщений по продукции одного вида или не содержать ни одного такого сообщения. Обновить инвентарный файл на основе вспомогательного, образовав новый файл g .

№ 7.15. Составить программу назначения студентов на стипендию по результатам экзаменационной сессии:

1) если сессия сдана на все «5», то стипендия повышенная;

2) если на «4» и «5», то стипендия обыкновенная;

3) если есть хотя бы одна «3», то стипендия не назначается.

Исходные данные ввести из файла на диске. Вывод обеспечить в трех вариантах: 1) на экран дисплея; 2) на печатающее устройство; 3) в файл на диске.

№ 7.16. Даны координаты n точек на плоскости: $X_1, Y_1, \dots, X_n, Y_n$ ($n=20$). Найти номера двух точек, расстояние между которыми наибольшее (считать, что такая пара точек единственная). Координаты точек вводятся из файла на диске. Вывести координаты точек в виде таблицы, номера двух точек, расстояние между которыми наибольшее и само расстояние между ними. Вывод пре-

дусмотреть в трех вариантах: 1) в файл на диске; 2) на печатающее устройство; 3) на экран дисплея.

№ 7.17. Дан файл f , содержащий сведения об учениках школы. Сведения об ученике состоят из его имени и фамилии и названия класса (года обучения буквы), в котором он учиться и среднего балла по результатам в последней четверти.

а) Выяснить, сколько учеников школы не имеют отметок ниже четырех.

б) Собрать в файле g сведения о лучших учениках школы, т.е. об учениках не имеющих средний балл ниже 4,5.

Составить программу. Вывод по желанию пользователя либо на экран, либо в файл на диске, либо на печать.

№ 7.18. Информация о температуре воздуха за месяц задана в виде массива. Определить:

а) сколько раз температура опускалась ниже 0 градусов по Цельсию;

б) температура скольких дней была ниже среднемесячной.

Результаты выдать на печать. Исходные данные должны быть подготовлены на дискете.

№ 7.19. По результатам экзаменационной сессии подготовить напечатанный протокол, который должен содержать данные о неуспевающих студентах (пофамильно), среднюю оценку каждого студента, средний балл по группе. В заголовке протокола указать название факультета и номер группы. Файл с исходными данными подготовить на дискете.

№ 7.20. Подготовить на диске файл с фамилиями и оценками, полученными студентами группы во время сессии. Определить число неуспевающих студентов, фамилии студентов сдавших экзамены на отлично и средний балл по группе. Протокол обработки данных выдать на печать, либо вывести на экран.

§8. Разработка внутренних процедур и функций

В данном параграфе рассматриваются следующие вопросы: понятие подпрограммы, процедуры, функции.

Теория

Подпрограмма представляет собой группу операторов, логически законченную и специальным образом оформленную. Подпрограмму можно вызывать неограниченное число раз из различных частей программы. Использование подпрограмм позволяет улучшить структурированность программы и сократить ее размер.

По структуре подпрограмма почти полностью аналогична программе и содержит заголовки и раздел операторов (блок), однако в блоке подпрограммы отсутствует раздел подключения модулей. Кроме того, заголовок подпрограммы по своему оформлению отличается от заголовка программы.

Работа с подпрограммой состоит из двух этапов:

- описание подпрограммы;
- вызов подпрограммы.

Любая подпрограмма должна быть предварительно описана, после чего допускается ее вызов. При описании подпрограммы указывается ее имя, список параметров и выполняемые подпрограммой действия. При вызове подпрограммы указываются имя подпрограммы и список аргументов (фактических параметров), передаваемых подпрограмме для работы.

В различных модулях Delphi имеется большое число стандартных подпрограмм, которые можно вызывать без предварительного описания. Кроме того, программист может создавать свои подпрограммы, которые также называют *пользовательскими*. Подпрограммы делятся на *процедуры* и *функции*, которые имеют между собой много общего. Основное различие между ними заключается в том, что функция может возвращать под своим именем в качестве результата значение и соответственно может использоваться в качестве операнда выражения.

С подпрограммой взаимодействие осуществляется по управлению и по данным. Взаимодействие *по управлению* заключается в передаче управления из программы в подпрограмму и органи-

зации возврата в программу. Взаимодействие *по данным* заключается в передаче подпрограмме данных, над которыми она выполняет определенные действия. Этот вид взаимодействия может осуществляться следующими основными способами:

- с использованием файлов;
- с помощью глобальных переменных;
- с помощью параметров.

Наиболее часто используется последний способ. При этом различают параметры и аргументы. *Параметры* (формальные параметры) являются элементами подпрограммы и используются при описании алгоритма, выполняемого подпрограммой. *Аргументы* (фактические параметры) являются элементами вызывающей программы. Они замещают параметры при вызове подпрограммы. При этом осуществляется проверка на соответствие типов и количества параметров и аргументов. Имена параметров и аргументов могут различаться, однако их количество и порядок следования должны совпадать, а типы параметров и соответствующих им аргументов должны быть совместимыми. Для прекращения работы подпрограммы можно использовать процедуру Exit, которая прерывает выполнение операторов подпрограммы и возвращает управление вызывающей программе. Подпрограммы можно вызывать не только из программы, но и из других подпрограмм.

Процедуры

Описание *процедуры* включает в себя заголовок и блок, который, за исключением раздела подключения модулей, не отличается от блока программы.

Заголовок состоит из ключевого слова procedure, имени процедуры и необязательного списка параметров в круглых скобках с указанием типа каждого параметра. Заголовок имеет формат:
Procedure <Имя> [(формальные параметры)];

Для обращения к процедуре используется *оператор вызова* процедуры. Он включает имя процедуры и список фактических параметров, заключенный в круглые скобки. Например, Test(a,b,c), Test — имя вызываемой процедуры, a, b, c — фактические параметры. Процедура Test должна быть описана с помощью подобного заголовка:

```
Procedure Test(a:integer;b:Real; var c:Real);
```

где *a*, *b* — исходные данные, передаваемые процедуре из вызывающей процедуры;
c — результат работы процедуры, возвращаемый в вызывающую процедуру.

Функции

Описание *функции* состоит из заголовка и блока. *Заголовок* включает ключевое слово `Function`, имя функции, необязательный список формальных параметров, заключенный в круглые скобки, и тип возвращаемого функцией значения. Заголовок имеет формат:

```
Function <Имя> [ (Формальные параметр) ] : <Тип результата>;
```

Возвращаемое значение может иметь любой тип, кроме файлового. *Блок* функции представляет собой локальный блок, по структуре аналогичный блоку процедуры. В теле функции должен быть хотя бы один оператор присваивания, в левой части которого стоит имя функции. Именно он и определяет значение, возвращаемое функцией. Если таких операторов несколько, то результатом функции будет значение последнего выполненного оператора присваивания. В этих операторах вместо имени функции можно указывать переменную `Result`, которая создается в качестве синонима для имени функции. В отличие от имени функции, переменную `Result` можно использовать в выражениях блока функции. С помощью переменной `Result` можно в любой момент получить внутри блока доступ к текущему значению функции.

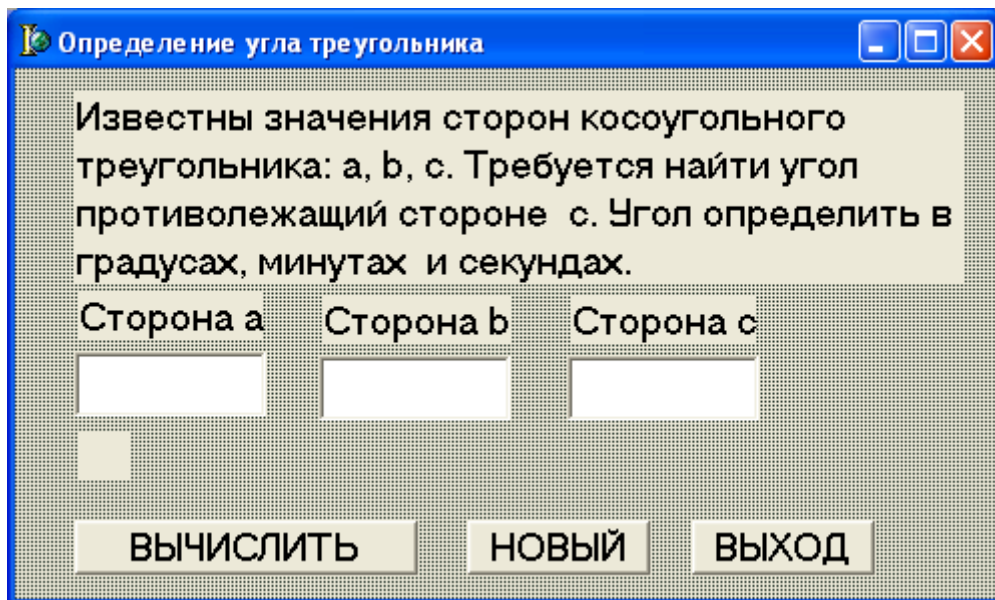
Замечание

Имя функции можно использовать в выражениях блока функции, однако это приводит к рекурсивному вызову функцией самой себя.

Вызов функции осуществляется по ее имени с указанием в круглых скобках списка аргументов, которого может и не быть. При этом аргументы должны попарно соответствовать параметрам, указанным в заголовке функции, и иметь те же типы. В отличие от процедуры, имя функции может входить как операнд в выражения. Например, `T:=Factor(n)/(n-k);` где `Factor(n)` вызов функции для вычисления факториала. Функция `Factor(n)` должна быть описана примерно таким заголовком:

```
Function Factor(i:Integer):Int64;
```

Задача



Текст модуля

```

unit Unit1;
.....
var
  Form1: TForm1;

implementation

{$R *.dfm}
{процедура выделяет из угла в градусах целые градусы, це-
лые минуты и секунды}
procedure GMS(Ugol:real;var G,M:Integer; S:Real);
begin
  G:=Trunc(Ugol+0.1);
  M:=Trunc((Ugol-G)*60);
  S:=((Ugol-G)*60-M)*60;
end;
procedure TForm1.Button1Click(Sender: TObject);
Var a,b,c: Real;
  p : Real;
  G,M : Integer;
  S : Real;
  Tan_Pol_Ugla, Ugol_b_Rad,Ugol_b_Grad:Real;
begin
  a:=StrToFloat(Edit1.Text);

```

```

b:=StrToFloat(Edit2.Text);
c:=StrToFloat(Edit3.Text);
p:=(a+b+c)/2;
If (a+b>c) and (a+c>b) and (b+c>a) Then begin
Tan_Pol_Ugla:=Sqrt((p-a)*(p-b)/(p*(p-c)));
Ugol_b_Rad:=Arctan(Tan_Pol_Ugla)*2;
Ugol_b_Grad:=Ugol_b_Rad/pi*180;
GMS(Ugol_b_Grad,G,M,S);
Label5.Caption:='Угол,      противоположащий      стороне      с,
равен'+#13+IntToStr(G)+'гр. '
      +IntToStr(M)+'мин. '+FloatToStrF(S,ffFixed,3,1)+'сек.'
end
Else Label5.Caption:='ТРЕУГОЛЬНИК НЕ СУЩЕСТВУЕТ';
end;

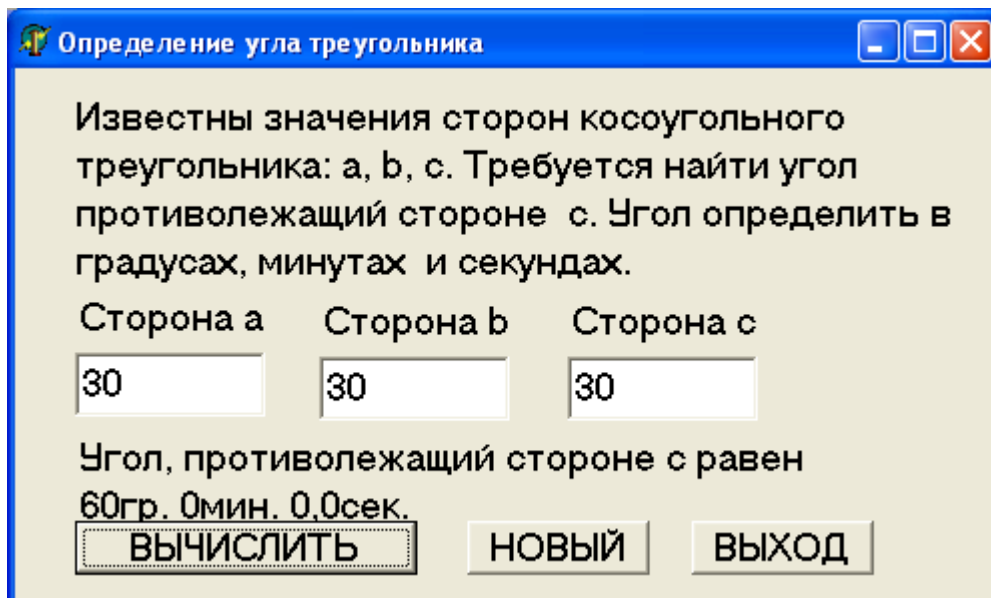
procedure TForm1.Button2Click(Sender: TObject);
begin
Edit1.Text:=' ';
Edit2.Text:=' ';
Edit3.Text:=' ';
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
Form1.Close
end;

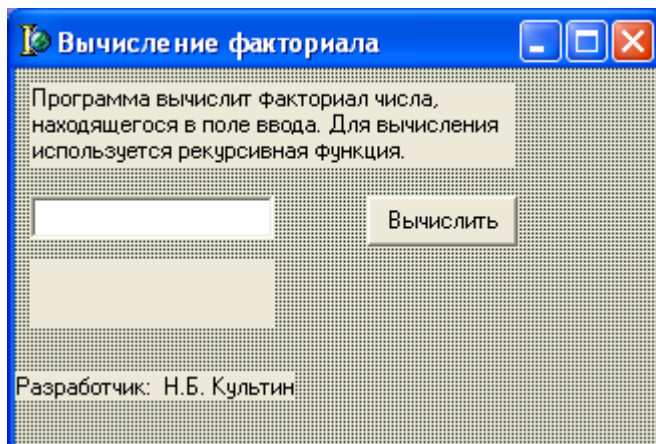
end.

```

Возможный результат решения:



Задача



Текст модуля

```

unit factor_ ;
.....
var
  Form1: TForm1;

implementation

{$R *.DFM}

// рекурсивная функция
function factorial(n: integer): integer;
begin
  if n > 1
  then factorial := n * factorial(n-1) // функция вызывает сама себя
  else factorial:= 1; // факториал 1 равен 1

```

```

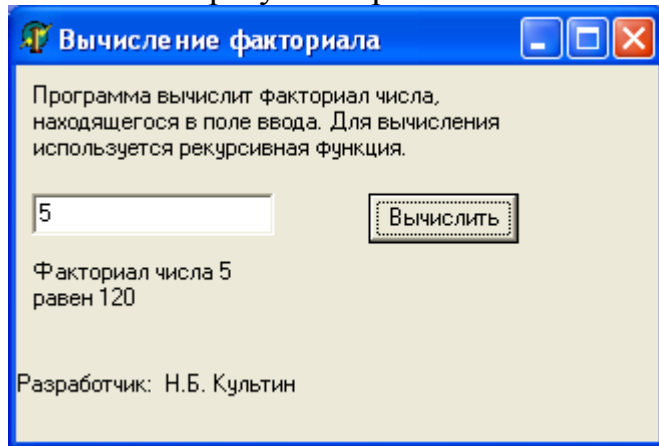
end;

procedure TForm1.Button1Click(Sender: TObject);
var
  k:integer; // число, факториал которого надо вычислить
  f:integer; // значение факториала числа k
begin
  k := StrToInt(Edit1.Text);
  f := factorial(k);
  label2.caption:='Факториал числа '+Edit1.Text
  +' равен '+IntToStr(f);
end;

end.

```

Возможный результат решения:



Задача

Составить программу для вычисления значений выражений $P=(x_1+y_1)/2$ и $Q=(x_2+y_2)/2$, где x_1 и x_2 — корни квадратного уравнения с коэффициентами A, B, C , а y_1 и y_2 — корни квадратного уравнения с коэффициентами D, F, G . Если хотя бы одно из уравнений не имеет действительных корней выдать на экран сообщение об этом и прекратить выполнение программы.

В программе разработать функцию для проверки введенных в поле ввода данных. Вводить в поле ввода можно только числа. Десятичным разделителем является запятая. Для вычисления корней квадратного уравнения разработать процедуру.

Компоненты

Имя компонента	Свойства компонента	Значение	Назначение
Form1	Caption	Выражение	Заголовок формы
Label1	Caption	Программа вычисляет значение выражений:	Справочная информация для

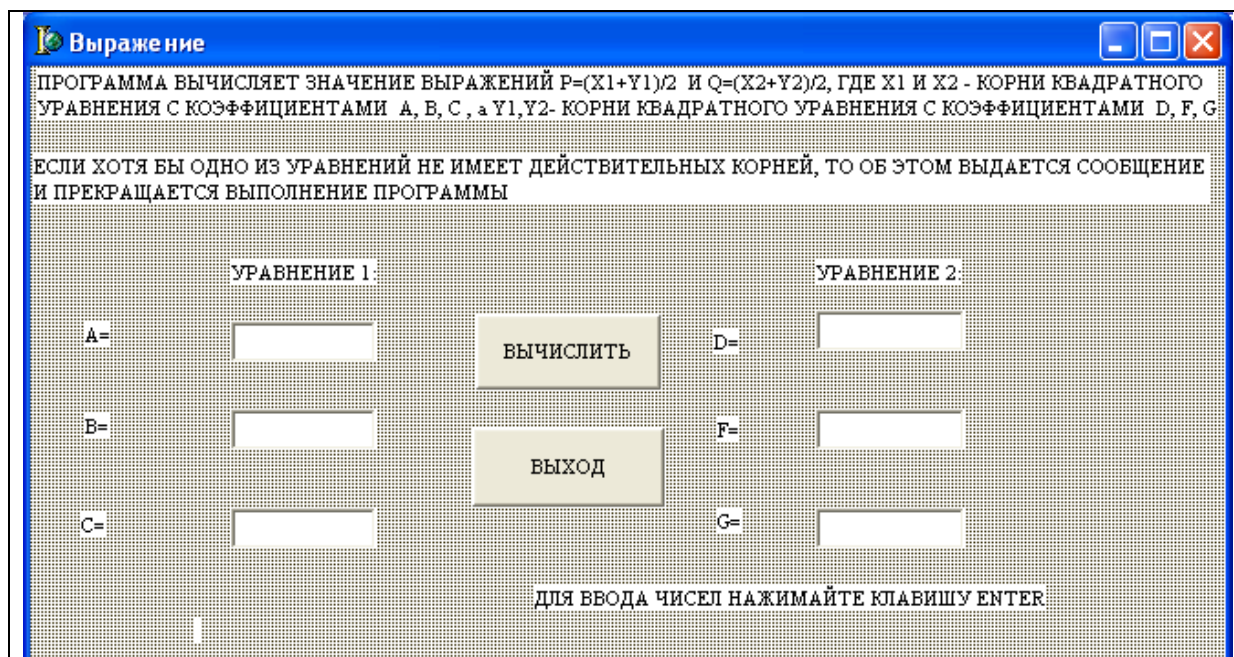
	WordWrap	$P=(X1+Y1)/2$ и $Q=(X2+Y2)/2$, где $X1$ и $X2$ – корни квадратного уравнения с коэффициентами A, B, C , а $Y1, Y2$ – корни квадратного уравнения с коэффициентами D, F, G True	пользователя программы Перенос не уместившихся слов на новую строку
Label2	Caption WordWrap	Если хотя бы одно из уравнений не имеет действительных корней, то об этом выдается сообщение и выполнение программы прекращается True	Подсказка пользователю Перенос не уместившихся слов на новую строку
Label3	Caption	Уравнение 1	Подсказка пользователю
Label4	Caption	Уравнение 2	Подсказка пользователю
Label5	Caption	A=	Подсказка пользователю
Label6	Caption	B=	Подсказка пользователю
Label7	Caption	C=	Подсказка пользователю
Label8	Caption	D=	Подсказка пользователю
Label9	Caption	F=	Подсказка пользователю
Label10	Caption	G=	Подсказка пользователю
Label11	Caption		Поле для вывода результата
Label12	Caption	Для ввода чисел нажимайте клавишу ENTER	Подсказка пользователю
Edit1	Text	Поле для ввода коэффициента A
Edit2	Text	Поле для ввода коэффициента B
Edit3	Text	Поле для ввода коэффициента C
Edit4	Text	Поле для ввода коэффициента D
Edit5	Text	Поле для ввода коэффициента F
Edit6	Text	Поле для ввода

			коэффициента G
Button1	Caption	ВЫЧИСЛИТЬ	Кнопка для вычисления
Button2	Caption	ВЫХОД	Кнопка закрытия формы и выхода из программы

Переменные

Обозначение в программе	Содержание	Тип
A	Коэффициент первого квадратного уравнения	Вещественный
B	Коэффициент первого квадратного уравнения	Вещественный
C	Коэффициент первого квадратного уравнения	Вещественный
D	Коэффициент второго квадратного уравнения	Вещественный
F	Коэффициент второго квадратного уравнения	Вещественный
G	Коэффициент второго квадратного уравнения	Вещественный
S	Строка символов, передаваемая из поля ввода	Строковый
i	Номер по порядку символа в строке символов	Локальная переменная. Целый
Z	Счетчик запятых	Локальная переменная. Целый
DT	Дискриминант	Локальная переменная. Вещественный
R	Признак решения уравнения	Локальная переменная. Логический
R1	Признак решения уравнения 1	Логический
R2	Признак решения уравнения 2	Логический
Key	Ключ — какая клавиша нажата	Слово, целый беззнаковый
X1	1 корень уравнения 1	Вещественный
X2	2 корень уравнения 1	Вещественный
Y1	1 корень уравнения 2	Вещественный
Y2	2 корень уравнения 2	Вещественный
P	$P=(X1+Y1)/2$	Вещественный
Q	$Q=(X2+Y2)/2$	Вещественный

Проект формы



Текст модуля

```
unit Unit1;
```

```
.....
```

```
var
```

```
Form1: TForm1;
```

```
A,B,C:Real;// коэффициенты первого квадратного уравнения
```

```
D,F,G:Real;// коэффициенты второго квадратного уравнения
```

```
implementation
```

```
{$R *.dfm}
```

{Функция проверяет, является ли введенное значение числом. Допустимыми значениями в числе являются цифры от 0 до 9, минус в качестве первого символа, одна десятичная запятая. Функция принимает значение Истина, если в строке встретился запрещенный символ.}

```
function CheckEdit(s:string):boolean;
```

```
// s строка символов, передаваемая из поля ввода
```

```
Var i:integer;// номер по порядку символа в строке символов
```

```
Z:integer;// счетчик запятых
```

```
begin
```

```
CheckEdit:=False;// значению функции присваивается ЛОЖЬ
```

```
Z:=0;// обнуление счетчика десятичных запятых
```

```
// цикл выполняется столько раз, сколько символов
```

```
// во введенной строке
```

```
for i:=1 to Length(S) do
```

```
Begin
```

```
// если в строке встречается запятая, счетчик увеличивается на 1
```

```
if s[i]=',' Then Z:=Z+1;
```

```
// если символ не относится к числу допустимых, значение
```

```
// функции принимает значение ИСТИНА
```

```

if not(s[i] in ['0'..'9','-','(',')])Then CheckEdit:=True
//иначе, если не первый символ равен минусу или счетчик
// запятых больше или равен 2,
// значение функции принимает значение ИСТИНА
else if ((i<>1) and (S[i]='-')) or (Z>=2) Then CheckEdit:=True
end;
end;
// процедура вычисления корней квадратного уравнения
procedure Root(A,B,C:Real;Var R:Boolean;Var X1,X2:Real);
Var DT:Real;//дискриминант
begin
// признак действительных решений у уравнения
// принимает значение ИСТИНА
R:=True;
DT:=sqr(B)-4*A*C;
if DT>0 Then begin
        X1:=(-B-sqrt(DT))/(2*A);
        X2:=(-B+sqrt(DT))/(2*A);
        end
else
// признак действительных решений у уравнения
// принимает значение ЛОЖЬ
        R:=False;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
Form1.Close
end;
// процедура обработки события — переход вниз из поля ввода
procedure TForm1.Edit1KeyDown(Sender: TObject; var Key: Word; Shift:
TShiftState);
begin
// если была нажата клавиша «ENTER», то если значение вызванной
// функции проверки допустимости символов в строке, которой
// в качестве фактического параметра передается содержимое поля
//ввода, равно истине (символ недопустим), то поле ввода
// очищается и фокус остается в нем
if Key=VK_RETURN
then if CheckEdit(Edit1.Text) Then begin
                Edit1.Clear;
                Edit1.SetFocus
                end
// иначе фокус переносится в следующее поле ввода

```



```

Else
Edit2.SetFocus;

end;
// процедура аналогична процедуре
// procedure TForm1.Edit1KeyDown
procedure TForm1.Edit2KeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);
begin
if Key=VK_RETURN
then if CheckEdit(Edit2.Text) Then begin
Edit2.Clear;
Edit2.SetFocus
end
Else
Edit3.SetFocus;
end;

// процедура аналогична procedure TForm1.Edit1KeyDown
procedure TForm1.Edit3KeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);
begin
If Key=VK_RETURN
then if CheckEdit(Edit3.Text) Then begin
Edit3.Clear;
Edit3.SetFocus
end
Else
Edit4.SetFocus;
end;

// процедура аналогична procedure TForm1.Edit1KeyDown
procedure TForm1.Edit4KeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);
begin
If Key=VK_RETURN
then if CheckEdit(Edit4.Text) Then begin
Edit4.Clear;
Edit4.SetFocus
end
Else
Edit5.SetFocus;
end;

// процедура аналогична procedure TForm1.Edit1KeyDown
procedure TForm1.Edit5KeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);
begin
If Key=VK_RETURN
then if CheckEdit(Edit5.Text) Then begin
Edit5.Clear;

```

```

Edit5.SetFocus
end
Else
Edit6.SetFocus;
end;

// процедура аналогична procedure TForm1.Edit1KeyDown
procedure TForm1.Edit6KeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);
begin
If Key=VK_RETURN
then if CheckEdit(Edit6.Text) Then begin
Edit6.Clear;
Edit6.SetFocus
end
Else
Button1.SetFocus;
end;

// процедура связанная с кнопкой вычислить
procedure TForm1.Button1Click(Sender: TObject);
Var R1,R2:Boolean; {признаки решения первого и второго квадратных
уравнений }
P,Q:Real;// значения вычисляемых выражений
X1,X2:Real;// корни первого уравнения
Y1,Y2: Real;// корни второго уравнения
begin
// инициализация переменных
X1:=0;X2:=0;
Y1:=0;Y2:=0;
// ввод из полей ввода коэффициентов уравнений
A:=StrToFloat(Edit1.Text);
B:=StrToFloat(Edit2.Text);
C:=StrToFloat(Edit3.Text);
D:=StrToFloat(Edit4.Text);
F:=StrToFloat(Edit5.Text);
G:=StrToFloat(Edit6.Text);
// вызов процедуры вычисления корней квадратного уравнения
// (для двух уравнений), если уравнение не имеет
// действительных корней, признак решения
// принимает значение ложь
Root(A,B,C,R1,X1,X2);
Root(D,F,G,R2,Y1,Y2);
if not(R1) Then begin
ShowMessage('Уравнение 1 действительных корней не имеет');
Exit // прекращение выполнения программы

```

```

end
else
if not(R2) Then begin
ShowMessage('Уравнение 2 действительных корней не имеет ');
Exit
end
// если оба уравнения имеют действительные решения,
// то вычисляются значения P и Q, их значения выводятся
// на форму в поля вывода
else
begin
P:=(X1+Y1)/2;
Q:=(X2+Y2)/2;
Label11.Caption:='P= '+FloatToStrF(P,Fffixed,10,2)+
' Q= '+FloatToStrF(Q,Fffixed,10,2);
end;
end;
end.

```

Результаты работы программы

Выражение

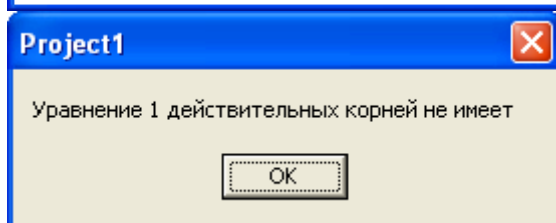
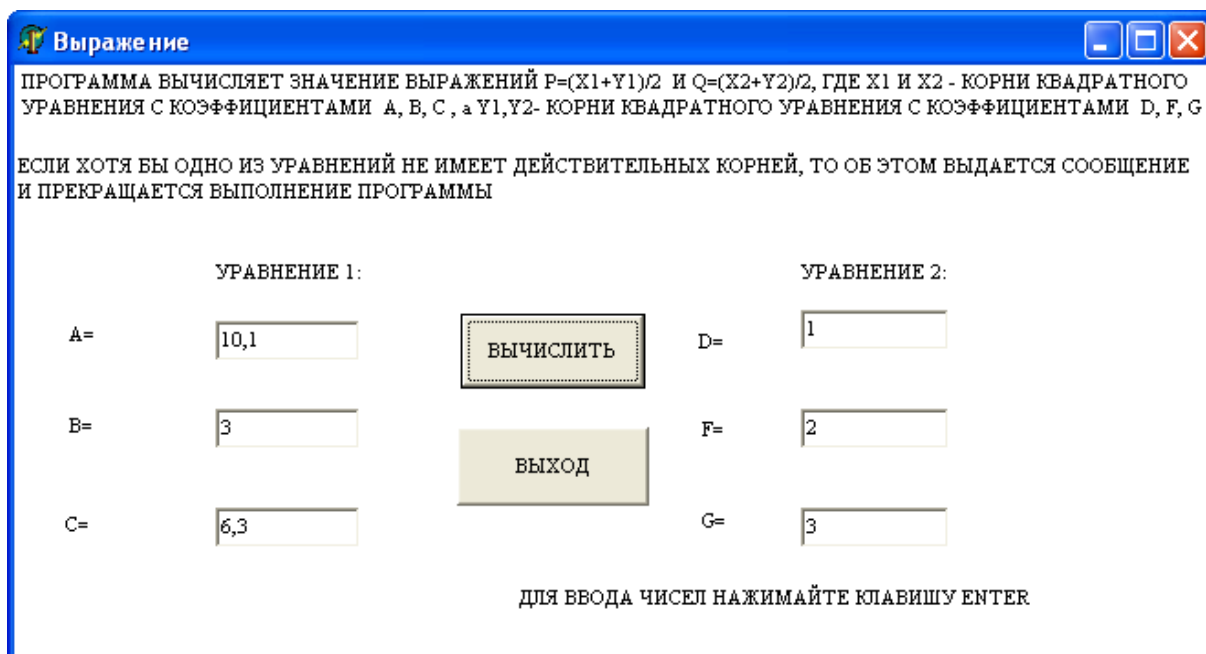
ПРОГРАММА ВЫЧИСЛЯЕТ ЗНАЧЕНИЕ ВЫРАЖЕНИЙ $P=(X1+Y1)/2$ И $Q=(X2+Y2)/2$, ГДЕ $X1$ И $X2$ - КОРНИ КВАДРАТНОГО УРАВНЕНИЯ С КОЭФФИЦИЕНТАМИ A, B, C , а $Y1, Y2$ - КОРНИ КВАДРАТНОГО УРАВНЕНИЯ С КОЭФФИЦИЕНТАМИ D, F, G

ЕСЛИ ХОТЯ БЫ ОДНО ИЗ УРАВНЕНИЙ НЕ ИМЕЕТ ДЕЙСТВИТЕЛЬНЫХ КОРНЕЙ, ТО ОБ ЭТОМ ВЫДАЕТСЯ СООБЩЕНИЕ И ПРЕКРАЩАЕТСЯ ВЫПОЛНЕНИЕ ПРОГРАММЫ

УРАВНЕНИЕ 1:			УРАВНЕНИЕ 2:	
A=	<input type="text" value="10,1"/>	<input type="button" value="ВЫЧИСЛИТЬ"/> <input type="button" value="ВЫХОД"/>	D=	<input type="text" value="1"/>
B=	<input type="text" value="31,3"/>		F=	<input type="text" value="15,1"/>
C=	<input type="text" value="6,3"/>		G=	<input type="text" value="3"/>

П= -8,89 Q= -0,21

ДЛЯ ВВОДА ЧИСЕЛ НАЖИМАЙТЕ КЛАВИШУ ENTER



Задачи для самостоятельного решения

Группа А

№ 8.1. Составить программу для вычисления $u = \min(a, b)$, $v = \min(\min(ab, a+b), \min(u, 3.14))$, где a и b — действительные числа. Нахождения минимума оформить как функцию.

№ 8.2. Даны действительные числа s, t . Получить

$$h(s, t) + \max(h^2(s, st), h^2(s-t, s+t)), \text{ где}$$

$$h(a, b) = \frac{a}{1+b^2} + \frac{b}{1+a^2} - (a-b)^3.$$

Вычисление $h(a, b)$ и \max оформить в виде процедур.

№ 8.3. Даны действительные числа s, t .

Получить $f(t, -2s, 1.17) + f(2.2, t, s-t)$,

где

$$f(a, b, c) = \frac{2a - b - \sin c}{5 + |c|}.$$

Для вычисления $f(a, b, c)$ в программе предусмотреть процедуру.

№ 8.4. Даны действительные числа s, t . Получить

$$g(1.2, s) + g(t, s) - g(2s-1, st),$$

$$\text{где } g(a,b) = \frac{a^2 + b^2}{a^2 + 2ab + 3b^2 + 4}.$$

Вычисление g в программе оформить в виде процедуры.

№ 8.5. Составить программу для вычисления суммы ряда

$$\sum_{n=0}^m (-1)^n \frac{x^{2n}}{(n!)^2 2^{2n}},$$

где x — вещественное число;

n, m — натуральные числа.

Для возведения в степень и вычисления факториала разработать функции.

№ 8.6. Даны натуральные числа m и n . Получить

$$\frac{m!+n!}{(m+n)!}$$

Для вычисления факториала в программе предусмотреть функцию.

№ 8.7. Даны коэффициенты двух квадратных уравнений $a_1x^2+b_1x+c_1=0$ и $a_2x^2+b_2x+c_2=0$.

Составить программу для вычисления значений:

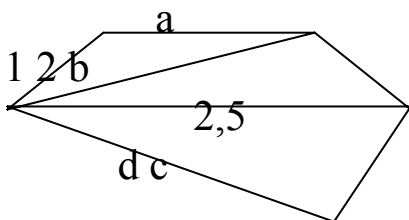
$$Z_1 = \frac{x_1 + y_1}{2}; Z_2 = \frac{x_2 + y_2}{2},$$

где x_1 и y_1 — корни 1-го уравнения;

x_2 и y_2 — корни 2-го уравнения.

Для решения квадратного уравнения предусмотреть в программе процедуру.

№ 8.8. Даны действительные числа a, b, c, d . Найти площадь пятиугольника, изображенного на рисунке ниже. Определить процедуру вычисления площади треугольника по трем сторонам.



№ 8.9. Составить программу для вычисления

$$Q = \frac{1}{2c(n-1)x^{n-1}},$$

где ($n \geq 1$).

Возведение в степень оформить в виде функции.

Группа Б

№ 8.10. Написать программу для определения взаимного расположения точек с координатами X_1, Y_1 и X_2, Y_2 по отношению к прямой, которая задается уравнением $AX+BY+C=0$. Для этого в программе необходимо сравнить знаки выражений, вычисляемых по формулам $AX_1+BY_1+C=0$ и $AX_2+BY_2+C=0$. При этом возможны следующие ситуации:

— если знаки обоих выражений совпадают, то обе точки лежат по одну сторону от прямой,

— если знаки у выражений различны, то точки лежат по разные стороны от прямой,

— если одно или оба выражения равны нулю, то соответственно одна или обе точки лежат на прямой.

Для вычисления выражений вида $AX+BY+C=0$ разработать функцию.

В программе предусмотреть следующую форму запросов при вводе данных:

A, B, C=

X1, Y1=

X2, Y2=

В качестве результата выводить одно из следующих сообщений «ПО РАЗНЫЕ СТОРОНЫ» или «ПО ОДНУ СТОРОНУ» или «ТОЧКА номер точки ЛЕЖИТ НА ПРЯМОЙ».

№ 8.11. Написать программу для определения углов треугольника α, β, γ по заданным значениям сторон A, B, C, используя теорему косинусов:

$$\cos \alpha = \frac{B^2 + C^2 - A^2}{2BC}, \cos \beta = \frac{A^2 + C^2 - B^2}{2AC},$$

$$\cos \gamma = \frac{A^2 + B^2 - C^2}{2AB}.$$

В программе предусмотреть проверку существования треугольника с заданными сторонами. Условие существования:

$$(A+B>C)\wedge(A+C>B)\wedge(B+C>A).$$

Если указанное условие не выполняется, выдавать сообщение «ИЗ ВВЕДЕННЫХ СТОРОН ТРЕУГОЛЬНИК ПОСТРОИТЬ НЕЛЬЗЯ», затем передавать управление оператору ввода исходных данных.

Для определения угла использовать преобразование

$$\operatorname{tg} \alpha = \frac{\sqrt{1 - \cos^2 \alpha}}{\cos \alpha}.$$

В этом случае:

$$\alpha = \begin{cases} \operatorname{arctg} \frac{\sqrt{1 - \cos^2 \alpha}}{\cos \alpha}, & \text{если } \cos \alpha > 0 \\ \operatorname{arctg} \frac{\sqrt{1 - \cos^2 \alpha}}{\cos \alpha} + 180, & \text{если } \cos \alpha < 0 \\ 90, & \text{если } \cos \alpha = 0 \end{cases}$$

Аналогично для β и γ .

Значения углов выводить в градусах, минутах и секундах, округляя до 1 сек.

Для определения и вывода значения углов в гр., мин., сек. использовать процедуру.

В программе предусмотреть следующую форму запросов при вводе исходных данных:

СТОРОНЫ А, В, С =

Форма вывода результатов:

УГОЛ АЛЬФА = значение α в гр., мин., сек.

УГОЛ БЕТА = значение β в гр., мин., сек.

УГОЛ ГАММА = значение γ в гр., мин., сек.

№ 8.12. Написать программу вычисления площади треугольного участка по известному значению боковой стороны А в метрах и углов при основании α , β в градусах, минутах и секундах.

$$S = \frac{A^2 \sin \beta (\cos \beta + \sin \beta \operatorname{ctg} \alpha)}{2}.$$

Площадь вычислять в гектарах с точностью до 0,1 га.

В программе предусмотреть проверку $\alpha + \beta < 180$ (для этого углы необходимо предварительно перевести в градусы). В случае нарушения указанного соотношения, вывести сообщение «УГЛЫ НЕРЕАЛЬНЫ» и передать управление оператору ввода исходных данных.

В программе воспользоваться функцией пользователя для перевода углов из градусной меры в радианную.

Форма запросов при вводе исходных данных:

СТОРОНА =

ПЕРВЫЙ УГОЛ: ГР., МИН., СЕК =

ВТОРОЙ УГОЛ: ГР., МИН., СЕК =

Форма вывода результата:

ПЛОЩАДЬ = значение S ГА.

№ 8.13. Написать программу для вычисления площади треугольного участка по известному значению основания C в метрах и углов при основании α , β в градусах, минутах и секундах. Площадь определить в гектарах с точностью до 0,01 га.

В программе предусмотреть проверку соотношения $\alpha + \beta < 180^\circ$ (для этого α и β необходимо предварительно перевести в градусы с десятичной дробной частью). В случае невыполнения этого соотношения вывести сообщение «СУММА УГЛОВ ПРЕВОСХОДИТ 180 ГР.» и передать управление оператору ввода исходных данных.

Формула для вычисления площади:

$$S = \frac{C^2 \sin \alpha \sin \beta}{2 \sin(\alpha + \beta)}.$$

В программе воспользоваться внутренней функцией для перевода угла из градусной меры в радианную.

Форма запросов при вводе исходных данных:

СТОРОНА =

ПЕРВЫЙ УГОЛ: ГР., МИН., СЕК. =

ВТОРОЙ УГОЛ: ГР., МИН., СЕК. =

Форма вывода результата:

ПЛОЩАДЬ = значение S ГА.

№ 8.14. Написать программу для вычисления площади четырехугольного участка по заданным значениям двух противоположных углов α , β и образующих эти углы сторон A , B и C , D

$$S = (A \cdot B \sin \alpha + C \cdot D \sin \beta) / 2.$$

Углы вводить в градусах, минутах и секундах, стороны — в метрах, значение площади выводить в гектарах, округляя до 0,01 га. В программе предусмотреть проверку существования четырехугольника с указанными значениями сторон и противоположных углов.

$$|A^2 + B^2 - 2AB \cos \alpha - C^2 - D^2 + 2CD \cos \beta| < 0,1.$$

Если указанное неравенство не выполняется, вывести сообщение «УЧАСТОК С УКАЗАННЫМИ ПАРАМЕТРАМИ НЕ СУЩЕСТВУЕТ» и передать управление операторам ввода исходных данных.

В программе предусмотреть следующие запросы при вводе исходных данных:

СТОРОНЫ A , B =

УГОЛ МЕЖДУ A , B : ГР., МИН., СЕК.=

СТОРОНЫ C , D =

УГОЛ МЕЖДУ C , D : ГР., МИН., СЕК.=

Форма вывода результата:

ПЛОЩАДЬ= значение S ГА.

В программе воспользоваться функцией для перевода угла из градусной меры в радианную.

№ 8.15. Из вершины O участка, изображенного на рис. 6.7, измерены расстояния A , B , C до остальных его вершин и углы α_1 , α_2 . Написать программу для вычисления площади этого участка по формуле

$$S = \frac{B(A \sin \alpha_1 + C \sin \alpha_2)}{2}.$$

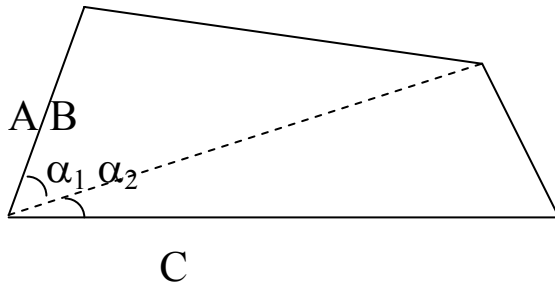


Рис. 6.7

Расстояния А, В, С измерены в метрах, углы α , β в градусах, минутах, секундах. Площадь вычислить в гектарах с точностью до 0,01 га.

В программе предусмотреть следующие запросы при вводе:

А, В, С=

УГОЛ МЕЖДУ А и В: ГР., МИН., СЕК.=

УГОЛ МЕЖДУ С и D: ГР., МИН., СЕК.=

Для вычисления синусов углов, заданных в градусах, минутах и секундах разработать функцию.

Форма вывода результата:

ПЛОЩАДЬ= значение S ГА.

№ 8.16. Написать программу, которая определяет, как расположена точка с координатами X_3, Y_3 по отношению к прямой, проведенной через точки X_1, Y_1 и X_2, Y_2 и точке с координатами X_0, Y_0 (рис. 6.8).

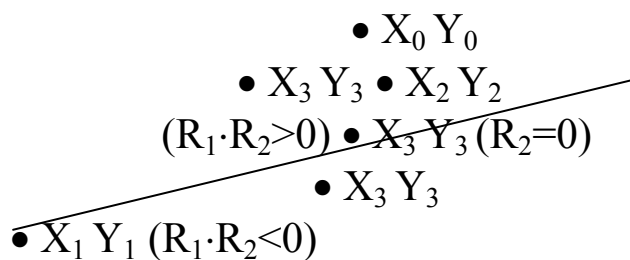


Рис. 6.8

Для этого программа должна вначале вычислить величину

$$R_1 = \frac{Y_0 - Y_1}{Y_2 - Y_1} - \frac{X_0 - X_1}{X_2 - X_1},$$

Предусмотреть проверку знаменателя на равенство 0. Если знаменатель оказался нулевым ввести другие координаты точек.

Затем проверить условие $R_1 = 0$. Если это условие выполняется, необходимо вывести сообщение «КОНТРОЛЬНАЯ ТОЧКА

ЛЕЖИТ НА ПРЯМОЙ» и передать управление операторам ввода X_0, Y_0 для исправления координат контрольной точки.

Если условие $R_1=0$ не выполняется, необходимо вычислить величину

$$R_2 = \frac{Y_3 - Y_1}{Y_2 - Y_1} - \frac{X_3 - X_1}{X_2 - X_1}.$$

Если R_1 и R_2 совпадут, значит, точки лежат по одну сторону от прямой, если не совпадут, значит, точки лежат по разные стороны от прямой, если $R_2=0$, значит, точка $X_3 Y_3$ лежит на прямой.

Для вычисления R_1 и R_2 использовать внутреннюю функцию.

Форма запросов при вводе:

X1, Y1=

X2, Y2=

X0, Y0=

X3, Y3=

В качестве результата выводить одно из следующих сообщений:

«ТОЧКА номер точки ЛЕЖИТ ПО ОДНУ СТОРОНУ С КОНТРОЛЬНОЙ ТОЧКОЙ» или «ТОЧКА номер точки ЛЕЖИТ ПО РАЗНЫЕ СТОРОНЫ С КОНТРОЛЬНОЙ ТОЧКОЙ» или «ТОЧКА номер точки ЛЕЖИТ НА ПРЯМОЙ».

№ 8.17. Написать программу для вычисления по формуле :

$$\frac{\pi}{2} + \sum_{n=0}^{\infty} (-1)^{n+1} \cdot \frac{x^{2 \cdot n+1}}{2 \cdot n+1} = \frac{\pi}{2} - \left(x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots \right)$$

Для $|x| < 1$. Для возведения в степень разработать функцию пользователя. Число членов в разложении должен задать пользователь программы.

№ 8.18. Написать программу для вычисления по формуле:

$$\frac{\pi}{2} + \sum_{n=0}^{\infty} (-1)^{n+1} \cdot \frac{1}{(2 \cdot n+1) \cdot x^{2 \cdot n+1}} = \frac{\pi}{2} - \left(\frac{1}{x} - \frac{1}{3x^3} + \frac{1}{5x^5} - \frac{1}{7x^7} + \dots \right)$$

Для $|x| > 1$. Число членов в разложении должен задать пользователь программы. Для возведения в степень разработать функцию пользователя.

№ 8.19. Написать программу для вычисления по формуле:

$$1 - \sum_{n=1}^{\infty} (-1)^{n+1} \cdot \frac{2^{2 \cdot n-1} \cdot x^{2 \cdot n}}{(2 \cdot n)!} = 1 - \left(\frac{2^1 \cdot x^2}{2!} - \frac{2^3 \cdot x^4}{4!} + \frac{2^5 \cdot x^6}{6!} - \frac{2^7 \cdot x^8}{8!} + \dots \right)$$

Для $x < 1$. Число членов в разложении должен задать пользователь программы. Для возведения в степень и вычисления факториала разработать функции пользователя.

Оглавление

ВВЕДЕНИЕ	3
Глава 1. Работа в среде текстового редактора Word	5
§1. Общие сведения, терминология Microsoft Word.....	5
§2. Создание и сохранение нового документа	6
§3. Загрузка и редактирование документа.....	23
§4. Создание таблиц.....	35
§5. Редактор формул.....	44
§6. Инструмент Smart Art и создание фигур.....	50
§7. Режимы отображения документа.....	55
§8. Верстка многостраничных документов	58
§9. Работа с документами на электронных носителях, в которых содержатся гиперссылки.....	68
§10. Microsoft Power Point	73

Глава 2. Операционная система “WINDOWS XP”	78
§1. Общие сведения о системе “WINDOWS XP”	78
§2. Общие сведения о программе Проводник	82
§3 Сервисное обслуживание файловой системы	84
§4 Программы архивации	90
§5. Несколько слов о Windows Vista	92
Задания для самостоятельного решения	93
Глава 3. Работа в среде табличного процессора	97
§1. Вид окна Excel 2007	97
§2. Выделение ячеек, редактирование информации в ячейках	98
§3. Копирование и перемещение информации, содержащейся в ячейках	99
§4. Выполнение арифметических операций	101
§5. Создание прогрессий	107
§6. Форматы ячеек	108
§7. Расчёты с использованием функций	109
§8. Формулы массива	123
§9. Индикация ошибок при использовании формул	126
§10. Решение уравнений с одним неизвестным с использованием функций Excel 2007	127
§11. Решение оптимизационных задач линейного и нелинейного программирования с помощью процедуры «Поиск решения»	129
§12. Графическое представление данных с помощью диаграмм	137
§13. Использование Excel для управления базами данных	141
§14. Анализ данных в Excel с помощью сводных таблиц	146
§15. Создание и использование макросов в Excel	149
§16. Одновременное использование Excel и Word	153
Задачи для самостоятельного решения	155
Глава 4. Система управления базой данных Access	164

Условные обозначения и сокращения.....	164
§ 1. Введение.....	164
§ 2. Технология построения реляционной базы данных	168
§3. Создание базы данных	176
§ 4. Установка и исправление свойств полей	181
§ 5. Определение структурной связи между таблицами	192
§ 6. Ввод, просмотр и корректировка данных. Подстановки.....	196
§ 7. Создание форм для ввода/редактирования данных в таблицах.....	204
§ 8. Вывод на печать характеристик и содержимого таблиц	215
§ 9. Сортировка, фильтрация и организация выборки данных.....	217
§ 10. Создание и выполнение запросов на выборку данных.....	223
§ 11. Создание отчетов.....	231
Глава 5. Основы работы в среде Интернет	241
§ 1. Введение.....	241
§ 2. История Интернета	243
§ 3. Интернет в России.....	245
§ 4. Адреса и имена серверов	246
§ 5. Виды доступа в Интернет.....	247
§ 6. World Wide Web — «Всемирная паутина».....	248
§ 7. URL и протоколы доступа к ресурсам Интернета	249
§ 8. HTTP- и FTP-серверы	251
§ 9. Броузер Internet Explorer.....	251
§ 10. Поисковые системы Интернета	255
§ 12. Почтовая программа Outlook Express.....	260
§ 13. Почтовая система Mail.ru	266
§ 14. Программы архивации и разархивации	270
§ 15. Текстовый процессор Word и Интернет	271

§ 16. Размещение набора Web-страниц на сайте Narod.ru	273
§ 17. Создание сайтов средствами Мастерской Narod.ru	274
§ 18. Чат и чат-серверы.....	276
§ 19. Интернет-магазины и интернет-услуги.....	277
§ 20. Безопасность работы в Интернете.....	278
Приложение 1. Темы рекомендуемых лабораторных работ.....	282
Приложение 2. Кодировка русских букв латинскими	283
Приложение 3. Список полезных сайтов.....	284
Глава 6 Программирование в среде Delphi.....	284
§1. Разработка линейных программ	284
§2. Стандартные и библиотечные функции.....	310
§ 3. Программы с разветвлениями.....	335
§4. Циклы	355
§ 5. Использование массивов в программах,.....	373
§ 6. Разработка многооконных проектов.	384
§7. Внешние файлы.....	401
§8. Разработка внутренних процедур и функций.....	438