

И.Я. Ривкинд, Т.И. Лысенко,  
Л.А. Черникова, В.В. Шакоцько

# Информатика

# 11 КЛАСС

Академический уровень,  
профильный уровень



*Рекомендовано Министерством образования и науки Украины  
(приказ МОН Украины. № 235 от 16.03.2011 г.)*

**Переведено с издания:**

**Информатика : підруч. для 11 кл. загальноосвіт. навч. закл. : академ. рівень, профіл. рівень / Й.Я. Ривкінд, Т.І. Лисенко, Л.А. Чернікова, В.В. Шакоцько; за заг. ред. М.З. Згуровського. - К. : Генеза, 2011. - 304 с. : іл.**

Перевод с украинского

И.Я. Ривкинда, Т.И. Лысенко, Л.А. Черниковой, В.В. Шакоцько

Научную экспертизу проводил Институт математики НАН Украины.

Психолого-педагогическую экспертизу проводил Институт педагогики НАПН Украины.

**И74 Информатика : учеб. для 11 кл. общеобразоват. учеб. заведе-  
ний : академ. уровень, профил. уровень : пер. с укр. / И.Я. Рив-  
кин, Т.И. Лысенко, Л.А. Черникова, В.В. Шакоцько; под общ.  
ред. М.З. Згуровского. - К. : Генеза, 2011. - 304 с. : ил.**

**ISBN 978-966-11-0093-9.**

Материал учебника разделен согласно программе на 4 главы. В основу изложения учебного материала положены объектный и алгоритмический подходы. В начале каждого пункта приведены вопросы для актуализации знаний учащихся, на которые опирается изучение нового материала. Для лучшего восприятия и усвоения учениками учебного материала учебник содержит большое количество рисунков, таблиц, схем, копий изображений экрана монитора и другие наглядные материалы. Для повышения интереса к изучению предмета учебник, кроме основного материала, содержит рубрики «Для тех, кто хочет знать больше», «Это интересно знать», «Интересные факты из истории», «Дополнительные источники информации».

В конце каждого пункта приводятся вопросы для самоконтроля и практические задания, для которых определены уровни их сложности. Учебник содержит 20 практических работ, выполнение которых, согласно программе, обязательно для всех учащихся.

**ББК 32.81я721**

**ISBN 978-966-11-0093-9 (рус.)  
ISBN 978-966-11-0068-7 (укр.)**

©Ривкин И.Я., Лысенко Т.И.,  
Черникова Л.А., Шакоцько В.В., 2011  
©Издательство «Генеза»,  
оригинал-макет, 2011



## Уважаемые одиннадцатиклассники!

Уже на протяжении нескольких лет вы изучаете информатику. За это время вы ознакомились с большим количеством прикладных программ, научились применять разнообразные информационно-коммуникационные технологии в своей деятельности. В 11-м классе вы продолжите изучение основ информатики. Овладеете навыками работы с программой создания компьютерных публикаций Microsoft Publisher и системой управления базами данных Microsoft Access 2007, научитесь обрабатывать мультимедийные данные и создавать видеоклипы, ознакомитесь с современными Интернет-технологиями Веб 2.0. Наибольшее внимание вы уделите изучению важных составляющих информатики - алгоритмизации и программированию.

Учебник состоит из 4 глав, каждая из которых разделена на несколько пунктов. Изучение каждой главы начинается с краткого вступления, в котором указано, о чем вы узнаете во время ее изучения. Все пункты построены по единой структуре. В начале пункта предложено несколько вопросов, ответы на которые помогут вам вспомнить ранее изученное для лучшего восприятия нового материала. В пунктах содержится теоретический материал по каждой теме и описание технологии обработки данных с использованием соответствующих программных средств, приведены алгоритмы и примеры выполнения основных способов деятельности. В конце каждого пункта размещены вопросы для самопроверки полученных вами знаний и перечень практических заданий для отработки навыков работы.

Для каждого вопроса и задания определен его уровень сложности, который обозначается так: <sup>0</sup> - начальный к среднему уровню; • - достаточный уровень; \* - высокий уровень. Задания подобраны таким образом, чтобы последовательно и целенаправленно формировать у вас прочные навыки работы с программным обеспечением, умение самостоятельно решать учебные задачи с использованием ИКТ, развивать ваше мышление и навыки индивидуальной и коллективной работы. Также в учебнике размещены задания 20 обязательных практических работ, результат выполнения которых будет демонстрировать уровень усвоения вами учебного материала по определенному вопросу.

Для облегчения восприятия вами учебного материала авторы дополнили учебник схемами, таблицами, экранными копиями, включили примеры для демонстрации основных положений информатики. Дополнительные рубрики учебника: «Для тех, кто хочет знать больше», «Это интересно знать», «Интересные факты, из истории», «Дополнительные источники информации» помогут вам лучше понять и глубже усвоить основы науки. Все это дает возможность использовать учебник и для самостоятельного овладения материалом.

Авторы предлагают он-лайн поддержку изучения курса информатики 11-го класса на сайте [www.allinf.at.ua](http://www.allinf.at.ua), где учителя и ученики смогут найти файлы-заготовки для выполнения практических заданий, дополнительные материалы к отдельным темам, пообщаться с авторами на форуме, оставить свои отзывы и предложения в гостевой книге.

Желаем вам успехов в учебе и овладении современными информационными технологиями!

*С уважением, авторы*

# Глава 1

## ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБУЧЕНИИ

В этой главе вы узнаете о:

- программных средствах учебного назначения и особенностях их использования;
- возможностях педагогических программных средств для обучения предметам естественно-математического цикла;
- математическом процессоре и его использовании для автоматизации математических вычислений;
- построении графиков функций, решении уравнений и систем уравнений средствами математического процессора.

### 1.1. Программные средства учебного назначения и особенности их использования



1. Какие программные средства учебного назначения вам известны? Для чего они предназначены?
2. Приведите примеры мультимедийных курсов, с которыми вам приходилось работать на уроках по разным учебным предметам.
3. Какие электронные средства обучения вы использовали на уроках естественно-математического цикла?

### Программные средства учебного назначения

Когда вы выполняете домашнее задание, подбираете материал для написания реферата, готовитесь к контрольной работе, вам нужны разнообразные источники информации: учебники и сборники заданий, атласы, словари, хрестоматии, коллекции фотографий и т. п. Для выполнения учебных заданий вы можете использовать и разнообразные электронные средства. Это могут быть ресурсы Интернета, электронные энциклопедии, коллекции цифровых изображений, а также прикладные программы общего назначения, которые применяются для проведения вычислений (например, **Калькулятор**), для подготовки рефератов (например, текстовый процессор **Word 2007**), для создания и редактирования рисунков (например, графический редактор **Paint**) и др.

Вместе с тем существуют программы, непосредственно предназначенные для обеспечения процесса обучения. Такие программы называют **программными средствами учебного назначения**, или **педагогическими программными средствами (ППС)**. Примерами таких программ являются программно-методический комплекс «Таблица Менделеева», Электронное пособие «Биология 8—9», Программная среда «Система линейных уравнений» и др., которые вы могли использовать на уроках и во время подготовки к учебным занятиям в предыдущих классах (рис. 1.1).



Преимуществом программных средств учебного назначения по сравнению с традиционными средствами обучения является наличие удобных в использовании средств визуализации учебного материала: статическое и динамическое представление объектов, процессов, явлений, их компонентов, наглядное представление закономерностей и результатов проведенных экспериментов, опытов, найденных решений задач. Достоинством ППС является также быстрая обратная связь между пользователем и средствами ИКТ, обеспечивающая реализацию диалога между учеником и программной учебной средой.



Рис. 1.1. Использование электронного атласа по географии в учебном кабинете, оборудованном мультимедийным проектором

Такую обратную связь называют интерактивностью программного средства. Благодаря использованию ППС, обучение может осуществляться в том темпе, который больше всего устраивает учеников.

Педагогические программные средства можно распределить на:

- **электронные пособия** - электронные учебные издания, которые дополняют учебники и содержат учебный материал по определенному предмету, отдельным разделам учебной дисциплины, факультативного курса или курса по выбору, чаще всего представленный с использованием мультимедийных средств; например, Педагогическое программное средство «Физика 10—11» (рис. 1.2, а), Педагогическое программное средство «Алгебра, 11 класс»;
- **электронные (виртуальные) практикумы** — электронные учебные сборники практических заданий и упражнений, в том числе:
  - **виртуальные лаборатории**, например Виртуальная химическая лаборатория. 8—11 кл. (рис. 1.2, б), Программно-методический комплекс «DG — динамическая геометрия»;
  - **электронные тренажеры**, например «Майстер-клас». Клавіатурний тренажер з української мови;
  - **электронные задачкиники**, например Электронный задачник «Физика. 7—9», Программная среда «Система линейных уравнений»;
- **электронные средства контроля учебных достижений учащихся** - компьютерные программы, предназначенные для создания тестовых заданий, проведения тестирования и фиксации результатов; например Система интерактивного тестирования «Школярлик»;



Рис. 1.2. Примеры педагогических программных средств

- *мультимедийные средства иллюстративного и справочного назначения:*

- о *электронные атласы*. - электронные коллекции изображений разных объектов (карты, чертежи, рисунки и др.) со средствами навигации и поиска; например **Электронный атлас «Экономическая и социальная география мира, 10—11 классы»** (рис. 1.2, в);

- о *электронные хрестоматии* - электронные учебные издания литературно-художественных, исторических и других печатных произведений, музыкальных произведений, произведений изобразительного и киноискусства или их фрагментов; например **фонохрестоматия «Школьная коллекция». Фонохрестоматия для 10—11 классов по английскому языку;**

- о *электронные энциклопедии* - электронные справочные издания, которые содержат основные сведения из одной или нескольких отраслей знаний и практической деятельности, представленных в коротких статьях, и дополнены аудио- и видеоматериалами, средствами поиска и отбора справочных материалов; например **Электронная база знаний «Человек, общество и мир», Электронная энциклопедия «Мифы народов мира»;**

- о *электронные словари* - электронные издания словарей государственного или иностранных языков, которые содержат средства поиска слов и словосочетаний, дополненные возможностью озвучивания фрагментов словаря; например **Учебная среда «1001 слово»** (рис. 1.2, г)

и др.

Существуют комбинированные ППС, в которых объединены программные средства из разных групп, например **Интегрированный электронный комплекс «Экономическая и социальная география мира»**. Они получили обобщенное название **мультимедийные, или интерактивные, курсы**.

Чаще всего педагогическое программное средство имеет такие компоненты:

- **меню программы**, которое отображает содержание материала в программном средстве или предоставляет доступ к его основным функциям;
- **гипертекстовая система навигации** между блоками учебного или справочного материала;
- **поисковая система** для быстрого перехода к нужному блоку учебного материала;
- **справка** по учебному материалу или по работе с программой.

В зависимости от его типа, в состав педагогического программного средства могут входить:

- текстовый информационный блок;
- коллекция графических изображений;
- коллекция аудио- и видеоматериалов;
- блок выполнения тренировочных упражнений и практических заданий;
- контролирующий блок

И др.


Некоторые ППС разработаны по клиент-серверной технологии. Предполагается, что на компьютере учителя устанавливается серверная часть таких средств, на компьютерах учеников — клиентская.

Выбор для работы того или иного педагогического программного средства зависит от учебного задания, которое необходимо выполнить.



ППС могут содержать не только программный материал по предмету, но и дополнительный материал, например теоретический, который выходит за пределы школьной программы, задачи повышенной сложности, средства для проведения учебных исследований и др. Это может быть полезным во время подготовки к олимпиадам и другим интеллектуальным и творческим соревнованиям, при написании работ МАН и т. п.

Программные средства учебного назначения для изучения различных предметов можно, например, скачать бесплатно на сайте Острів знань (<http://www.ostriv.in.ua>) в разделе **Сервіси => Файловий архів => Електронні засоби навчання**.

 Перед тем как использовать компьютерные учебные программы, их нужно установить на компьютер (инсталлировать), соблюдая «Правила использования компьютерных программ в учебных заведениях», которые утверждены Министерством образования и науки Украины. Все программное обеспечение должно быть сертифицировано, что отмечается в сопроводительной документации к программному средству.

Установка программного обеспечения производится с носителя, на котором оно записано, на жесткий диск компьютера. Для этого нужно выполнить такую последовательность действий:

1. Поставить диск с нужным педагогическим программным средством в устройство для работы с компакт-дисками.
2. Открыть окно с содержанием папок и файлов носителя.
3. Запустить файл с именем setup.exe (или install.exe).
4. Следовать последующим указаниям программы инсталляции.

### Проверьте себя


- 1°. Что такое педагогические программные средства? Приведите примеры учебных программ.
- 2°. Приведите примеры программных средств для поддержки учебных предметов естественно-математических дисциплин.
- 3°. На какие виды разделяют педагогические программные средства? Охарактеризуйте их.
- 4°. Какие возможности имеют современные педагогические программные средства?
- 5°. Какие компоненты имеются в педагогических программных средствах всех видов? Каково их назначение?
- 6°. Опишите ориентировочную последовательность действий для инсталляции учебных компьютерных программ.
- 7°. Каково назначение электронных пособий и с какой целью они могут использоваться?
- 8°. Какие средства относятся к электронным (виртуальным) практикумам? Каково их назначение и с какой целью они могут использоваться?
- 9°. Какие средства принадлежат к интерактивным компьютерным моделям? Каково их назначение и с какой целью они могут использоваться?
- 10°. Что может выполнять ученик, используя компьютерные программы для изучения физики, биологии, химии, географии, астрономии?
- 11°. Найдите в Интернете перечень педагогических программных средств, рекомендуемых МОН Украины для использования в учебных заведениях. Какие из этих средств можно использовать для преподавания предметов вашего профиля обучения?
- 12\*. Какие он-лайн ресурсы можно использовать на уроках для предметов вашего профиля обучения? Найдите их в Интернете.


**Выполните задания**

1. Установите соответствие между группами педагогических программных средств и их описаниями.

	<i>ППС</i>		<i>Описание</i>
1	Электронные пособия	А	Электронные учебные издания литературно-художественных, исторических и других печатных, музыкальных произведений, произведений изобразительного и киноискусства или их фрагментов
2	Электронные (виртуальные) практикумы	Б	Электронные издания словарей государственного или иностранных языков, которые содержат средства поиска языковых единиц и дополнены возможностью озвучивания фрагментов словаря
3	Электронные средства контроля учебных достижений	В	Электронные справочные издания основных сведений из одной или нескольких отраслей знаний и практической деятельности, представленных в коротких статьях, дополненных аудио- и видеоматериалами, средствами поиска и отбора справочных материалов
4	Электронные атласы	Г	Электронные учебные издания, которые дополняют учебники и содержат учебный материал по определенному предмету, отдельным разделам учебной дисциплины, факультативного курса или курса по выбору, чаще всего представлен с использованием мультимедийных средств
5	Электронные словари	Д	Электронные коллекции изображений разных объектов (карты, чертежи, рисунки и др.) со средствами навигации и поиска
6	Электронные энциклопедии	Е	Электронные учебные издания практических заданий и упражнений
7	Электронные хрестоматии	Ж	Компьютерные программы, предназначенные для создания тестовых заданий, проведения тестирования и фиксации результатов




2\*. Укажите виды педагогических программных средств, которые вы выберете для: а) подготовки к лабораторной работе по химии; б) написания реферата по математике; в) решения задачи по экономике; г) написания научно-исследовательской работы по географии; д) подготовки сообщения о выдающихся научных открытиях XX ст.

 3\*. Выясните, какие ППС, предназначенные для изучения математики, доступны на сайте Острів знань (<http://www.ostriv.in.ua>) в разделе *Сервісу => Файловий архів => Електронні засоби навчання*. Определите, к каким группам ППС они относятся.

4. Запустите на выполнение электронное пособие по математике, установленное на вашем компьютере. Изучите меню программы. Перейдите к первому разделу пособия, просмотрите его темы. Выясните назначение элементов управления и условных обозначений в тексте пособия. Составьте их описание. Приведите три примера использования пособия для решения учебных заданий.

5. Откройте главную страницу сайта Математика для школы (<http://www.formula.co.ua>). Ознакомьтесь с содержанием рубрик Алгебра и Геометрія. Найдите определение слова *вектор*. Переведите  $30^\circ$  в радианы, используя средства сайта, и вычислите значение выражения  $\sin 30^\circ + \cos 30^\circ$  с использованием встроенного калькулятора. Объясните, чем могут быть полезны для вас материалы этого сайта.



-  6\*. Загрузите из сайта Острів знань (<http://www.ostriv.in.ua>) педагогическое программное средство Динамічна геометрія. Проинсталлируйте программу на своем компьютере. Ознакомьтесь со справкой по использованию программы. Просмотрите режимы работы программы. К какому типу ППС можно отнести эту программу? Приведите три примера использования программы для решения учебных заданий. Подготовьте презентацию о возможностях программы.
- 7\*. Загрузите из сайта Острів знань (<http://www.ostriv.in.ua>) из раздела **Навчання => Шкільні предмети => Хімія => Електронні засоби для вивчення хімії** программно-методический комплекс Таблиця Менделєєва. Проинсталлируйте комплекс на своем компьютере. Ознакомьтесь со справкой по использованию программы. Ознакомьтесь с режимами работы программы. К какому типу ППС можно отнести эту программу? Приведите три примера использования программы для решения учебных заданий. Подготовьте презентацию о возможностях программы.
- 8\*. Откройте главную страницу сайта Шкільна фізика (<http://sp.bdpu.org>). Определите, какие материалы сайта могут быть вам полезны во время изучения физики. Ознакомьтесь в рубрике Демонстрації с интерактивной моделью движения тела, брошенного под углом к горизонту. Объясните физическую суть продемонстрированного явления.
-  9\*. Загрузите из сайта Google Earth (<http://earth.google.com>) бесплатную версию программы Google Earth и запустите ее на своем компьютере. Найдите на модели Земли карту и фотографии вашей местности и исторических мест вашей области.
-  10\*. Откройте страницу Дистанційне обучение. Примеры тренажерів сайту Сумського державного університета (<http://dl.sumdu.edu.ua/ru/entrant/examples-of-simulators>). Ознакомьтесь с перечнем интерактивных средств, разработанных сотрудниками лаборатории. Определите особенности использования программного средства по химии Окислительно-восстановительная реакция. Составьте инструкцию по его использованию.

### Практическая работа № 1. Использование программных средств при изучении физики, химии, биологии

Внимание! Во время работы с компьютером соблюдайте правила безопасности и санитарно-гигиенические нормы.

- Запустите на выполнение педагогическое программное средство по профильному предмету (физика, химия, биология и др.), установленное на вашем компьютере.
  - Изучите меню программы.
  - Ознакомьтесь с содержанием программного средства.
  - Выясните назначение элементов управления и условных обозначений в тексте пособия.
  - Составьте их описание в тетради.
  - Запишите в тетрадь три примера использования курса для решения учебных заданий.
- Откройте вики-сайт [wiki.ciit.zp.ua](http://wiki.ciit.zp.ua).
  - Перейдите по ссылке Интернет-ресурсы в рубрике Свободная зона.
  - Ознакомьтесь с перечнем он-лайн ресурсов для вашего профильного предмета.
  - Выберите один из них и перейдите на соответствующий сайт.
  - Ознакомьтесь с возможностями данного веб-ресурса.
  - Выясните его назначение, круг пользователей, особенности использования для решения учебных задач по профильному предмету.
  - Полученные сведения запишите в тетрадь.

## 1.2. Математические процессоры. Программно-методический комплекс GRAN и его использование



1. Какие программные средства учебного назначения вы использовали на уроках математики? Для чего они предназначены?
2. Какие правила ввода формул в табличном процессоре Excel 2007?
3. Что такое график функции; график уравнения?
4. Какие исследования функции вы выполняли на уроках математики?
5. Приведите алгоритм решения уравнения графическим способом.

### Математические процессоры

Среди программного обеспечения для обучения математики важное место занимают *математические процессоры*. Основные возможности этих прикладных программ:

- вычисление значений числовых выражений;
- построение графиков функций, заданных разными способами;
- трассировка графиков (построение таблицы значений функции на основе построенного графика);
- нахождение координат точек пересечения графиков двух функций на заданном промежутке;
- нахождение нулей и экстремумов функции на заданном промежутке;
- нахождение приближенного решения уравнений и их систем;
- графическое решение неравенств и их систем;
- вычисление площадей и объемов геометрических фигур

И др.

Примерами таких программ являются GRAN, DG — Динамическая геометрия (рис. 1.3, а), Microsoft Mathematics (рис. 1.3, б), MathCad, EUREKA, MathLab, Maple, DERIVE, Advanced Grapher и др. Одни из этих программ - это мощные системы компьютерной математики с большими

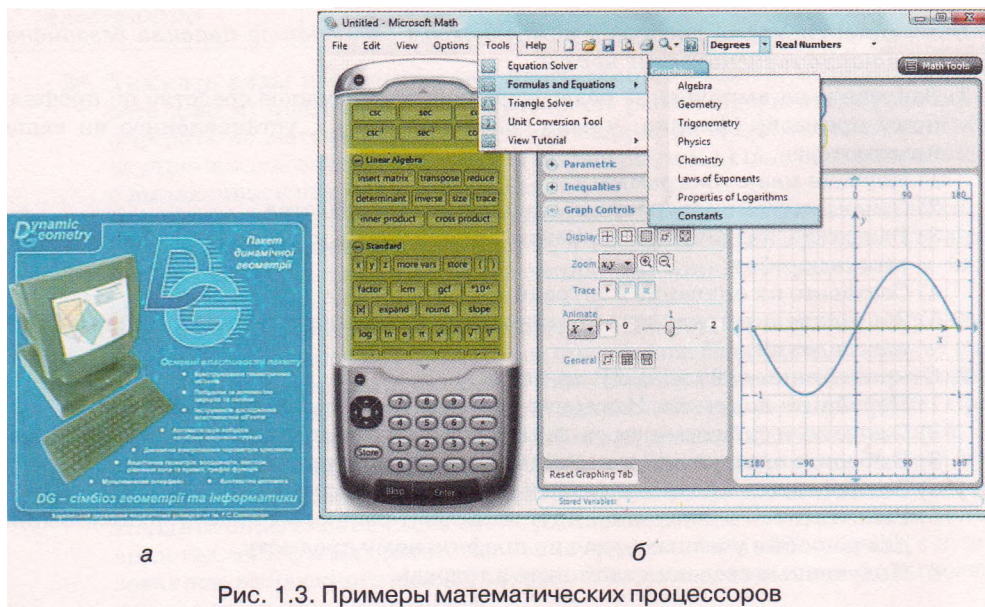


Рис. 1.3. Примеры математических процессоров



функциональными возможностями, которые рассчитаны на специалистов высокой квалификации, другие имеют значительно меньшие возможности и могут быть использованы учениками средней школы.


Использование математических процессоров для обучения дает возможность соединить вычислительные возможности компьютера в процессе исследования разнообразных функциональных зависимостей, освободившись от рутинных вычислений, с преимуществами графического представления информации, учесть индивидуальные возможности и способности каждого ученика по темпам обучения. Они также способствуют развитию графических навыков и образного мышления учащихся, геометрической интуиции и наглядных представлений с предметной отрасли.

### Программно-методический комплекс GRAN

Одним из первых в Украине педагогических программных средств был программный комплекс для поддержки обучения математики GRAN, разработка которого началась в 1989 году авторским коллективом под руководством известного украинского ученого Мирослава Ивановича Жалдака, академика АПН Украины, доктора педагогических наук, профессора (рис. 1.4).

Программно-методический комплекс GRAN, в состав которого входят педагогические программные средства GRAN1, GRAN-2D, GRAN-3D, обеспечивает поддержку изучения математики (планиметрии, стереометрии, тригонометрии, алгебры и начал анализа, начал теории вероятностей и математической статистики), а также отдельных разделов физики в школе (7-11 классы).

Рассмотрим использование математических процессоров для компьютерной поддержки уроков математики на примере программного средства GRAN1, которое предназначено для графического анализа функций, что и отображает его название G<sup>R</sup>aphic ANalysis.

Запуск программы осуществляется традиционным способом: *Пуск* => *GRAN1* => *GRAN1* или с использованием ярлыка . После этого на экране открывается окно программы (рис. 1.5), состоящее из трех внутренних окон: График, Список объектов, Ответы.

В верхней строке окна программы находится Строка заголовка, под ним - Строка меню. В программе реализована *контекстная справка*. Для ее получения нужно выбрать команду и нажать клавишу F1.

В окне График строятся графические изображения объектов и графиков функций, отображаются пояснительные тексты и др. При перемещении мыши по координатной плоскости в окне перемещается указатель координат -  $\vec{V}$  В верхней части окна График, над рабочим полем, отображаются текущие координаты указателя, в нижней части (Строке статуса) - максимальное и минимальное значения координат на плоскости.

Окно Список объектов состоит из двух частей. В верхней части находится поле со списком восьми типов зависимостей между переменными, которые можно выбрать для построения графического объекта, ниже - список всех введенных объектов, среди которых текущий объект отмечен меткой . Во второй части окна находятся сведения о текущем объекте: функция,



Рис. 1.4.  
М.И. Жалдак

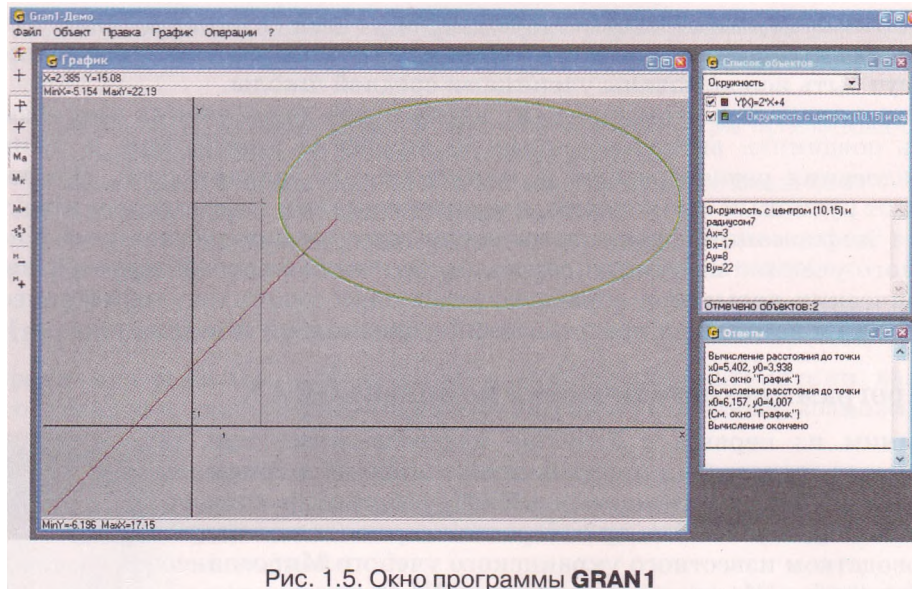


Рис. 1.5. Окно программы GRAN1

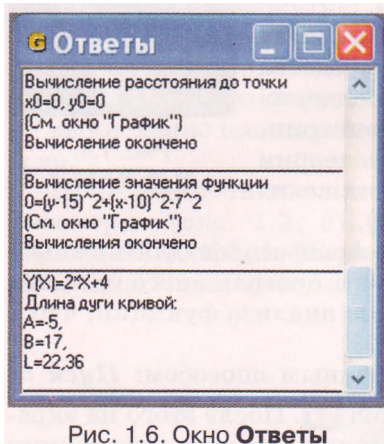


Рис. 1.6. Окно Ответы

промежутков, на котором она рассматривается, минимальное и максимальное значения функции или другие параметры выделенного графического объекта.

В окне Ответы (рис. 1.6) показаны результаты выполнения разнообразных операций, которые выбираются в меню *Операции* - вычисление расстояния до точки, вычисление длины дуги, определение длины ломаной и т. п. Это окно можно очистить, выполнив *Операции*  $\Rightarrow$  *Ответы*  $\Rightarrow$  *Очистить*.

Слева в окне программы находится панель инструментов, которая содержит такие элементы управления (табл. 1.1):

Таблица 1.1. Команды панели инструментов GRAN1

Пиктограмма	Команда	Пиктограмма	Команда	Пиктограмма	Команда
	Построить график		Открыть файл	$M_n$	Масштаб пользователя
	Очистить		Сохранить файл	$M_a$	Автоматический масштаб
	Декартовы координаты		Создать объект	$M_*$	Предыдущий масштаб
	Полярные координаты	$M_-$ $M_+$	Уменьшить масштаб Увеличить масштаб	$M_{-5}$ $M_5$	Начальный масштаб



## Выполнение вычислений в GRAN 1

Одной из самых распространенных функций математических процессоров является выполнение разнообразных математических вычислений. Программное средство GRAN1 имеет для этого широкие возможности.

Вычисления выполняются в окне Калькулятор (*Операции => Калькулятор*). Ввод выражения можно осуществлять либо с клавиатуры, либо используя кнопки окна Калькулятор. После завершения ввода выражения нужно нажать Enter или выбрать кнопку Вычислить в окне Калькулятор. Числовые выражения записываются по правилам, близким к правилам табличного процессора Excel 2007. Все допустимые функции и операции представлены на кнопках Калькулятора (рис. 1.7).

Пробелы в записи выражения не допускаются. Дробная часть в записи чисел отделяется от целой части точкой. Приоритет операций общепринятый. Для его изменения используются скобки. Редактирование введенного выражения осуществляется традиционными способами.

В таблице 1.2 приведены условные обозначения арифметических операций и функций.

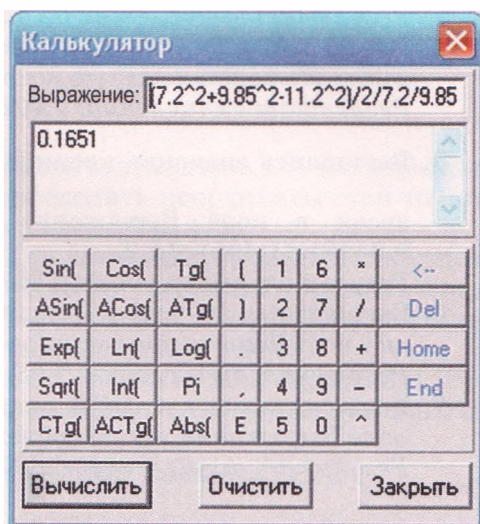


Рис. 1.7. Окно Калькулятор

Таблица 1.2. Условные обозначения арифметических операций и функций

Обозначение	Функция	Обозначение	Функция	Обозначение	Функция	Обозначение	Арифметическая операция
Sin	Синус	Cos	Косинус	Tg	Тангенс	*	Умножение
ASin	Арксинус	ACos	Арккосинус	ATg	Арктангенс	/	Деление
Exp	Экспонента	Ln	Логарифм натуральный	Log	Логарифм с произвольным основанием	+	Сложение
Sqrt	Арифметический квадратный корень	Int	Целая часть числа	Pi	Число $\pi$	-	Вычитание
CTg	Котангенс	ACTg	Арккотангенс	Abs	Модуль	^	Возведение в степень

Рассмотрим использование Калькулятора на таком примере:

Задача 1. Даны стороны треугольника:  $AB = 7,2$ ,  $BC = 11,2$ ,  $AC = 9,85$ .

Вычислить меры его углов (в радианах).



Для решения данной задачи нужно выполнить:

1. Запустить программу GRAN1.
2. Выполнить *Операции*  $\Rightarrow$  *Калькулятор*.
3. Вычислить значение косинуса угла  $A$  по формуле:  $\cos A = \frac{AB^2 + AC^2 - BC^2}{2 \cdot AB \cdot AC}$ , введя в поле **Выражение** такую последовательность символов  $(7.2^2 + 9.85^2 - 11.2^2) / 2 / 7.2 / 9.85$  (рис. 1.7).  
*Результат вычислений: 0.1651.*
4. Вычислить меру угла  $A$  (в радианах) по формуле  $A = \arccos(\cos(A))$ , введя в поле **Выражение** такую последовательность символов  $ACos(0.1651)$ .  
*Результат вычислений: 1.405.*
5. Вычислить значение косинуса угла  $B$  по формуле:  $\cos B = \frac{AC \cdot \sin A}{BC}$ ,

введя в поле **Выражение** такую последовательность символов  $9.85 \cdot \sin(1.405) / 11.2$ .

*Результат вычислений: 0.8674.*

6. Вычислить меру угла  $B$  (в радианах) по формуле:  $B = \arcsin(\sin(B))$ , введя в поле **Выражение** такую последовательность символов  $ASin(0.8674)$ .  
*Результат вычислений: 1.05.*
7. Вычислить меру угла  $C$  (в радианах) по формуле:  $C = \pi - A - B$ , введя в поле **Выражение** такую последовательность символов  $Pi - 1.405 - 1.05$ .  
*Результат вычислений: 0.687.*

## Построение графиков зависимости между переменными в GRAN 1 и их анализ

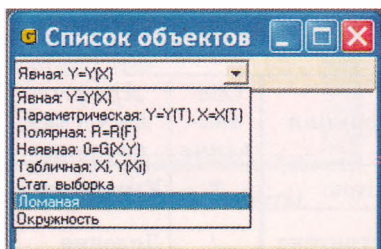


Рис. 1.8. Типы зависимостей между переменными

В GRAN1 можно построить графики восьми основных типов зависимостей между переменными (рис. 1.8). Одновременно в окне можно отобразить до пяти графиков, все они автоматически рисуются разными цветами. Цвет линий каждого объекта отображается в окне **Список объектов** возле уравнения функции и выбирается при создании графика.

Общий алгоритм построения графика зависимости между переменными:

1. Выбрать в окне **Список объектов** тип зависимости между переменными.
2. Выбрать в меню **Объект** команду **Создать**.
3. Ввести в поле диалогового окна **Ввод функции** соответствующее выражение, выбрать цвет графика и выбрать кнопку **ОК**.
4. Выбрать в меню **График** команду **Построить**.

Рассмотрим алгоритм построения графика функции на таком примере.

**Задача 2.** Построить график функции  $y = |x^2 - 8|x| + 7|$ .

Для этого нужно выполнить такую последовательность действий:

1. Запустить программу GRAN 1.
2. Выбрать в окне **Список объектов** тип зависимости **Явная: Y=Y(X)**.

3. Выбрать в меню Объект команду Создать.
4. Ввести в поле  $Y(X)=$  диалогового окна Ввод функции выражение  $ABS(X^2-8*ABS(X)+7)$ .
5. Выбрать на палитре цветов в окне Ввод функции цвет графика.
6. Выбрать в меню График команду Построить.

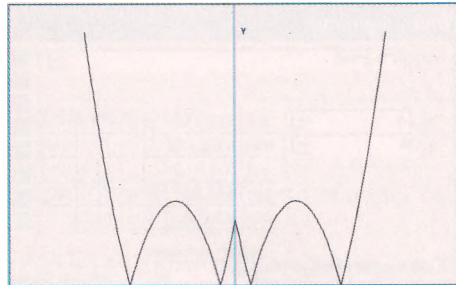


Рис. 1.9. График функции  
 $y = |x^2 - 8|x| + 7|$

Полученный результат представлен на рисунке 1.9.

Анализируя построенный график функции, можно определить:

- *нули функции.* Для этого следует установить указатель в точке пересечения графика функции с осью  $Ox$  и определить координаты этой точки. Они будут отображаться в верхней части окна График;
- *точки экстремумов и экстремумы.* Для этого следует установить указатель в наивысшую или самую нижнюю точку графика и выяснить координаты указателя в окне График. Максимальное и минимальное значения функции также отображаются в окне Список объектов;
- *промежутки возрастания и убывания.* Сначала нужно определить точки экстремумов, а затем записать промежутки возрастания и убывания.

### Решение уравнений и систем уравнений графическим способом

Рассмотрим, как можно использовать программное средство GRAN1 для графического решения уравнения с одной переменной. Алгоритм решения данного типа задач состоит из трех этапов:

1. Построить график зависимости.
2. Отметить на координатной плоскости точку пересечения графика функции с осью  $Ox$ .
3. Определить координаты указателя, которые отображаются в верхней строке окна График. Это и будет приближенным корнем уравнения.

Полученные таким способом значения являются приближенными. Погрешность возникает за счет того, что перемещение указателя на Рабочем поле имеет определенный шаг.

Рассмотрим графическое решение уравнений с одной переменной.

Задача 3. Решить уравнение  $x^3 - 2x + 6 = 0$  графическим способом.

Для этого нужно выполнить такую последовательность действий:

1. Запустить программу GRAN 1.
2. Выбрать в окне Список объектов тип зависимости Явная:  $Y=Y(X)$ .
3. Выбрать в меню Объект команду Создать.
4. Ввести в поле  $Y(X)=$  диалогового окна Ввод функции выражение  $X^3-2*X+6$ , выбрать цвет графика (рис. 1.10).
5. Выбрать в меню График команду Построить. На экране в окне График получим график введенной функции.
6. Выполнить *График => Список точек на графике => Запись*.
7. Установить указатель последовательно в точки пересечения графика функции с осью  $Ox$  (рис. 1.11).

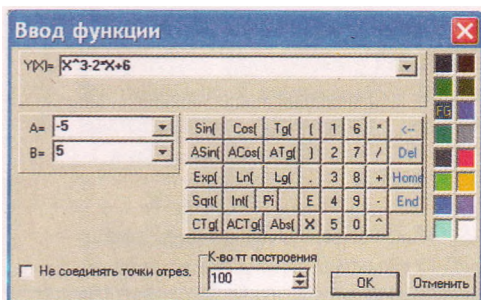


Рис. 1.10. Диалоговое окно **Ввод функции**

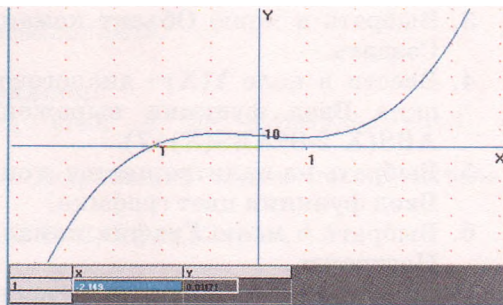


Рис. 1.11. Получение приближенного значения корня уравнения

8. Определить приближенное значение корня уравнения, которое отображается в нижней части окна График.

Ответ:  $x = -2.149$ .

Если точек пересечения графика с осью  $Ox$  будет несколько, то уравнение будет иметь несколько решений, которые и будут отображаться в таблице значений в окне График под координатной плоскостью.

Для системы уравнений алгоритм нахождения приближенного решения будет выглядеть так:

1. Построить графики каждого из уравнений системы.
2. Отметить на координатной плоскости точки пересечения построенных графиков.
3. Определить координаты указанных точек.

Задача 4. Решить систему уравнений  $\begin{cases} x + 3y = 9, \\ 2x - y = 4 \end{cases}$  графическим способом.

Для этого следует выполнить:

1. Запустить программу GRAN 1.
2. Выбрать в окне Список объектов тип зависимости Неявная:  $0=G(X,Y)$ .
3. Выбрать в меню Объект команду Создать.
4. Ввести в поле диалогового окна Ввод функции выражение  $X+3*Y-9$ .
5. Выбрать в меню График команду Построить.
6. Выбрать в окне Список объектов тип зависимости Неявная:  $0=G(X,Y)$ .
7. Выбрать в меню Объект команду Создать.
8. Ввести в поле диалогового окна Ввод функции выражение  $2*X-Y-4$ .
9. Выбрать в меню График команду Построить.
10. Выполнить *График => Список точек на графике => Запись*.
11. Выбрать указателем точку пересечения графиков (рис. 1.12).

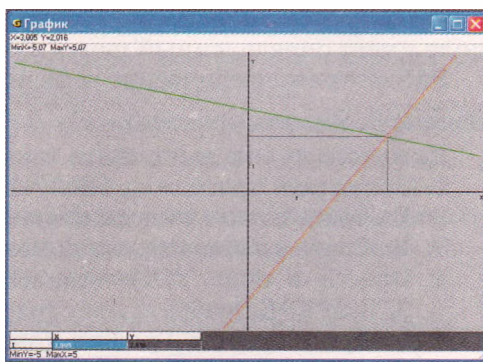


Рис. 1.12. Графическое решение системы уравнений  $\begin{cases} x + 3y = 9, \\ 2x - y = 4 \end{cases}$



## 12. Определить координаты точки пересечения в окне График.

Ответ: (3.005; 2.016).



## Решение простейших задач по планиметрии с использованием GRAN1

Рассмотрим использование программного средства **GRAN1** для решения геометрической задачи.

**Задача 5.** Определить вид четырехугольника  $ABCD$ , который задан координатами вершин  $A(1; 1)$ ,  $B(2; 3)$ ,  $C(0; 4)$ ,  $D(-1; 2)$ .

Чтобы решить эту задачу с использованием программы **GRAN1**, нужно выполнить такую последовательность действий:

1. Запустить программу **GRAN1**.
2. Выбрать в окне **Список объектов** тип зависимости **Ломаная**.
3. Выбрать в меню **Объект** команду **Создать**.
4. Ввести в диалоговое окно **Координаты вершин ломаной** (рис. 1.13) координаты вершин четырехугольника и выбрать кнопку **OK**.
5. Выбрать в меню **График** команду **Построить**.
6. Измерить по очереди длины сторон и диагоналей, выполнив *Операции => Длина ломаной*.

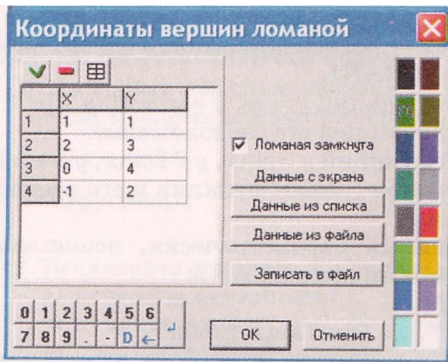


Рис. 1.13. Диалоговое окно Координаты вершин ломаной

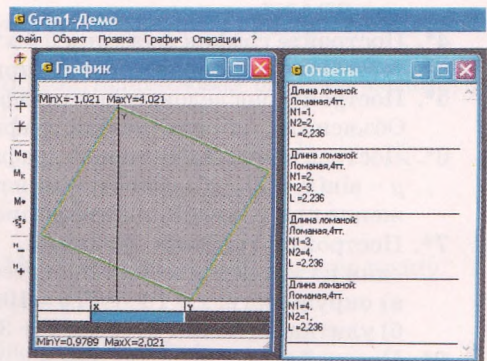


Рис. 1.14. Решение задачи по геометрии с использованием программы GRAN1

В окне **График** получили построенный четырехугольник, а в результате измерения в окне **Ответы** (рис. 1.14) получили такие значения: все стороны четырехугольника равны 2,236 линейные единицы, диагонали равняются 4,472 линейные единицы. Следовательно, заданный четырехугольник является квадратом.

## Проверьте себя

- 1°. Каково назначение математических процессоров? Приведите примеры математических процессоров.
- 2°. Опишите интерфейс окна программы GRAN1.
- 3°. Какие правила ввода числовых выражений в программе GRAN1?
- 4\*. Сравните правила ввода формул в математическом процессоре GRAN1 и табличном процессоре Excel 2007.
- 5°. Какие типы зависимости между переменными используются в GRAN1?
- 6°. Сколько графиков одновременно можно построить в GRAN1?
- 7°. Как очистить окно График?

8. Приведите алгоритм построения графика функции.  
 9. Приведите алгоритм построения графика уравнения.  
 10. Какие исследования функции можно осуществить на основе ее графика?  
 11. Объясните, как определить нули функции; экстремумы функции.  
 12. Приведите общий алгоритм решения уравнения графическим способом.  
 13. Приведите общий алгоритм решения системы уравнений графическим способом.  
 14. Объясните, почему решение уравнения или системы уравнений, полученные графическим и аналитическим способами, отличаются.

### Выполните задания

1. Рассчитайте значения выражений, используя одну из математических программ, например GRAN1:
- а)  $y = \cos 2x$  при  $x = 0; 1; 1,57$ ;  
 б)  $y = x^2 - 4|x| + 3$  при  $x = -3; 3$ ;  
 в)  $y = 1/(x^2 - 3)$  при  $x = -2; 0; 15$ .
2. Вычислите периметр и площадь прямоугольного треугольника, если один из катетов равен 7 см, а противоположный ему угол  $60^\circ$ , используя одну из математических программ, например GRAN1.
3. Вычислите периметр и площадь треугольника по трем заданным сторонам ( $a = 7, b = 5, c = 8$ ), используя одну из математических программ, например GRAN1.
- 4\*. Постройте график функции  $y = x^4 - 5x^2 + 3$  с использованием одного из математических процессоров, например GRAN1.
5. Постройте последовательно графики функций  $y = |x|, y = |x + 1|, y = |x| + 1$ .  
 Объясните, как изменяется график функции и его расположение.
6. Постройте последовательно графики функций  $y = \sin x, y = 2\sin x, y = \sin 2x, y = \sin(x + 2)$ . Объясните, как изменяется график функции и его расположение при изменении коэффициентов.
- 7\*. Постройте графики функций, заданных параметрически, используя один из математических процессоров, например GRAN1:
- а) окружность:  $x = 10\cos T, y = 10\sin T$ ;  
 б) улитка Паскаля:  $x = 5\cos^2 T + 3\cos T, y = 5\cos T \sin T + 3\sin T$
8. Постройте график функции  $y = |x^2| + x + 7$ , используя один из математических процессоров, например GRAN1. Исследуйте функцию на основе графика по приведенному плану и запишите ответы в тетрадь:
- а) четность, нечетность;  
 б) нули функции;  
 в) промежутки знакопостоянства: 1)  $y > 0$ ; 2)  $y < 0$ ;  
 г) точки экстремумов;  
 д) экстремумы;  
 е) промежутки возрастания и убывания: 1) возрастает; 2) убывает.
9. Решите уравнение  $x^3 = \sqrt{x}$  графическим способом, используя один из математических процессоров, например GRAN1.

10\*. Решите систему уравнений  $\begin{cases} x^2 + y^2 = 25 \\ xy = 12 \end{cases}$  графическим способом, используя один из математических процессоров, например GRAN1.

11\*. Решите задачу, используя один из математических процессоров, например GRAN1. Даны точки: A(0; 1), B(1; 0), C(1; 2), D(2; 1). Докажите равенство длин векторов: а)  $\overline{AB}$  и  $\overline{CD}$ ; б)  $\overline{AC}$  и  $\overline{BD}$ .

### Практическая работа № 2. Автоматизация математических вычислений

Внимание! Во время работы с компьютером соблюдайте правила безопасности и санитарно-гигиенические нормы.

- Вычислите значения выражений, используя одну из математических программ, например GRAN1. Запишите в тетрадь введенные выражения и результаты вычислений.

а)  $y = 2\sin x$  при  $x = 0$ ;

г)  $y = ||x - 1| - |x - 2||$  при  $x = -2, 15$ ;

б)  $y = \cos 2x$  при  $x = 0$ ;

д)  $y = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$  при  $a = 2, b = 8, c = 3$ ;

в)  $y = (x - 1)^2 / (x - 2)^3$  при  $x = -3$ ;

е)  $y = \pi r^2$  при  $r = -3, 9$ .

- Радиус Луны равен 1740 км. Вычислите площадь ее поверхности и объем космического тела, используя одну из математических программ, например GRAN1. Запишите в тетрадь введенные выражения и результаты вычислений.

### Практическая работа № 3. Построение графиков функций и их анализ

Внимание! Во время работы с компьютером соблюдайте правила безопасности и санитарно-гигиенические нормы.

- Постройте графики указанных функций, используя один из математических процессоров, например GRAN1:

а)  $y = 2\sin x$ ;

в)  $y = 2x$ ;

д)  $y = ||x - 1| - |x - 2||$ ;

б)  $y = \cos 2x$ ;

г)  $y = \frac{x - 1}{x - 2}$ ;

е)  $y = (1 + x)^x$ .

- Исследуйте функции а, в, д задания № 1 по их графикам согласно плану:

а) четность, нечетность;

б) нули функции;

в) промежутки знакопостоянства: 1)  $y > 0$ ; 2)  $y < 0$ ;

г) точки экстремумов;

д) экстремумы;

е) промежутки возрастания и убывания: 1) возрастает; 2) убывает.

### Практическая работа № 4. Нахождение приближенных решений уравнений и их систем

Внимание! Во время работы с компьютером соблюдайте правила безопасности и санитарно-гигиенические нормы.

- Найдите приближенные решения уравнений графическим способом, используя один из математических процессоров, например GRAN1:

а)  $y = x^2 - 4|x| + 3$ ;

б)  $y = \frac{1}{x^2 - 3}$ ;

в)  $y = \sqrt{||x| - 2|}$ .

- Найдите приближенные решения системы уравнений графическим способом, используя один из математических процессоров, например GRAN1:

а)  $\begin{cases} 3x - 7y = 9, \\ 5x + 13y = 7; \end{cases}$

б)  $\begin{cases} 3x^2 - 4y^3 = 12, \\ 2x^3 - 5y^2 = 8; \end{cases}$

в)  $\begin{cases} x + y = 1, \\ |xy| = \frac{1}{2}; \end{cases}$

г)  $\begin{cases} x + y + 1 = 0, \\ \sin(x + y) = 0. \end{cases}$



# Глава 2

## ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

В этой главе вы узнаете о:

- понятиях модели и моделирования, типах моделей;
- создании информационных моделей;
- алгоритмах и их свойствах, основных алгоритмических структурах;
- этапах решения задач с использованием компьютера;
- компьютерном моделировании;
- компьютерных программах;
- создании и отладке компьютерных программ в интегрированной среде визуальной разработки;
- методах обработки данных с использованием компьютерных программ.

### 2.1. Модели и моделирование. Типы моделей



1. Приведите примеры объектов, их свойств и значений этих свойств.
2. Кто такие модельеры? Что создают представители этой профессии?
3. Какие модели объектов вы знаете? Для чего они создаются? На каких уроках и как использовались модели объектов?
4. Какие процессы или явления могут быть описаны функцией  $y = kx$ ?

#### Понятие модели. Моделирование

Для изучения свойств и взаимосвязей объектов (предметов, процессов или явлений) люди проводят разнообразные исследования. Но не всегда можно или целесообразно исследовать сами предметы, процессы или явления непосредственно. В таких случаях создают и исследуют не сами объекты, а их модели. Термин *модель* происходит от латинского слова *modulus* - образец, аналог.

Модели создают для исследования объектов тогда, когда сам объект недоступен и его невозможно исследовать непосредственно (например, звезда созвездия Большой Медведицы или извержение вулкана), либо когда исследования объекта могут привести к его разрушению (например, мостовой переход), либо когда для его изготовления необходимы значительные средства (например, застройка нового микрорайона) и т. п.

Модели объектов создают не только тогда, когда они недоступны или дорого стоят, но и тогда, когда нужно исследовать конкретное свойство или группу свойств объекта. В таких случаях создают модель объекта, обязательно имеющую те свойства, которые исследуются, а другие свойства, несущественные для данного исследования, могут у модели отсутствовать.

Моделями являются знакомые вам карта Украины и глобус в кабинете географии, модели геометрических тел в кабинете математики, модели органов человека в кабинете биологии, план спортивной площадки вашей

школы, рисунок нового автомобиля, словесное описание незнакомого вам человека.

Люди создают модели одежды для изучения спроса на них, модели самолетов для изучения их аэродинамических свойств, модели корпусов автомобилей для изучения их прочности, модели архитектурных ансамблей для изучения их совместимости с уже существующими, моделируют химический эксперимент для изучения его последствий и т. п. Ученые-физики моделируют движение тела под действием различных сил, процесс ядерной реакции, создали модель так называемой ядерной зимы для иллюстрации последствий применения ядерного оружия, сейсмологи моделируют землетрясения, чтобы изучить их последствия для разных регионов планеты, экономисты создают модели развития общества.



Множество всех предметов, свойства которых и отношения между которыми исследуются, называют *предметной областью* этого исследования.

Так, предметной областью исследования для зоологов является множество животных, предметной областью для математиков при исследовании делимости чисел является множество целых чисел.



*Модель объекта* – это новый объект, который имеет свойства данного объекта, существенные для определенного исследования.

Процесс создания и исследования моделей называется *моделированием*.

Отметим, что в разных науках исследуют разные свойства объектов. И поэтому для каждого объекта могут существовать разные модели. Это зависит от того, какие именно свойства исследуются. Так, разными будут модели человека в исследованиях физика, химика, биолога, врача, модельера.

С другой стороны, разные объекты могут иметь одну и ту же модель. Так, прямоугольный параллелепипед может быть моделью книги, шкафа для одежды, дома и многих других объектов. А функция  $y = kx$  может служить моделью прямолинейного движения материальной точки с постоянной скоростью, изменения напряжения электрической сети в зависимости от силы тока при постоянном сопротивлении, стоимости покупки картофеля в зависимости от массы покупки и др.

Можно утверждать, что любая умственная деятельность человека оперирует образами предметов, процессов и явлений, которые являются, в сущности, их моделями. Действительно, рассуждая о конкретном объекте, человек выделяет из всех его свойств лишь отдельные, касающиеся цели его умственной деятельности о предметах или явлениях, которые исследуются.

Очевидно, что правильное построение моделей объектов и их исследование способствуют точности и правильности научных и инженерных выводов, предложений, решений. В современной науке и технике построение моделей, а также их исследование проводятся с использованием компьютеров, специальных компьютерных программ.

Кроме того, создают специальные компьютерные программы, которые реализуют модель объекта. Такие программы называют компьютерной моделью объекта. В наше время компьютерные модели широко используются

для исследования объектов, проведения вычислительных экспериментов в тех случаях, когда проведение реальных экспериментов невозможно, либо требует немалых средств, либо имеет непредсказуемые последствия.

### Типы моделей

Модели классифицируют по разным признакам (рис. 2.1):

- по способу представления,
- по отраслям использования,
- по фактору времени

и др.

По способу представления модели распределяют на материальные и информационные.

Игрушки, чучела животных, манекены, муляжи, глобус, модель водяной мельницы - это примеры материальных моделей. Материальные модели предназначены для проведения практических исследований.

Физическая карта Украины, уравнение химической реакции, математическая функция, рассказ о берегах Днепра - это примеры информационных моделей. Информационные модели предназначены для проведения теоретических исследований.



*Материальная модель объекта* - это модель объекта, представленная в виде его предметной копии.

*Информационная модель объекта* - это модель объекта, представленная в виде его описания.

Как и материальные модели, информационные модели одного и того же объекта будут разными, в зависимости от цели исследования. Например, информационная модель объекта «помидор» для поставщика будет содержать данные о размерах, условиях хранения, факторах и сроках созревания, максимальных сроках хранения и т. п. А для фермера информационная модель этого же объекта будет содержать данные о времени сеяния, регулярности прополки и поливки, рациональном использовании удобрений и т. п.

Информационные модели в свою очередь распределяют на:

- словесные (устные и письменные описания);
- графические (рисунки, чертежи, пиктограммы, карты и др.);
- структурные (таблицы, графики зависимости, диаграммы, схемы и др.);
- алгоритмические (правила, планы действий и др.);
- математические (формулы, уравнения, неравенства, функции и др.);
- специальные (химические формулы и уравнения, нотные записи, записи шахматных партий и др.).

Для создания информационной модели объекта нужно:

1. Определить объект исследования, для которого создается модель, и цель исследования.
2. Выделить те свойства объекта, которые являются существенными для указанного исследования.
3. Установить взаимосвязи между выбранными свойствами и выразить их, используя одну из форм представления.





Рис. 2.1. Типы моделей

По отраслям использования модели распределяют на:

- учебные - создаются для обучения (наглядные пособия, тренажеры, например тренажер самолета для обучения летчиков, тренажер для имитации перегрузок во время вывода космического корабля на орбиту и др.);
- исследовательские — создаются для проведения исследований (модель речного теплохода, вертолета, гидростанции для проведения исследований на прочность, плавучесть, аэродинамические свойства, влияние на экологию, модель компьютера, прибор, имитирующий разряд молнии, и др., а также формулы, уравнения, функции и др., которые можно использовать для исследования объектов);
- игровые - создаются с целью моделирования ситуаций для разработки и проверки разнообразных стратегий поведения, адаптации к определенным условиям (военные, экономические, спортивные, деловые игры и др.).

По фактору времени модели распределяют на:

- статические — это модели, в которых зафиксировано состояние объекта в определенный момент времени и последующие изменения этого объекта не учитываются, например гипсовая модель пирамиды Хеопса;
- динамические - это модели, которые предназначены для исследования изменения значений свойств объекта со временем, например модель изменения уровня воды в горных реках во время таяния снегов; модель погоды в данном регионе; модель Солнечной системы, которая изменяется в зависимости от результатов новых исследований.



## Проверьте себя

- 1°. Что такое предметная область исследования? Приведите примеры.
- 2°. Что такое модель? Приведите примеры моделей.
- 3°. Для чего создаются модели? Приведите примеры.
- 4°. Поясните, в чем заключается процесс моделирования.
- 5°. Чем модель объекта отличается от реального объекта?
- 6°. Поясните, почему один и тот же объект может иметь разные модели. Приведите примеры.
- 7°. Поясните, почему один и тот же объект может быть моделью разных объектов. Приведите примеры.
- 8°. Как классифицируют модели по способу представления? Охарактеризуйте их.
- 9°. Какие модели называют материальными? Приведите примеры.
- 10°. Какие модели называют информационными? Приведите примеры.
- 11°. Как классифицируют информационные модели? Опишите разные виды информационных моделей.
- 11°. Как классифицируют модели по отраслям использования? Приведите примеры.
- 12°. Как классифицируют модели по фактору времени? Приведите примеры.
- 13°. Что такое компьютерное моделирование?



## Выполните задания

1. К какому типу моделей (материальная или информационная) относятся такие модели:
  - а) рисунок пирамиды;
  - б) начерченный план школы;
  - в) формула площади прямоугольника;
  - г) нотная запись песни;
  - д) макет застройки участка;
  - е) кулинарный рецепт;
  - ж) химический опыт;
  - з) модель самолета;
  - и) модель атома;
  - к) запись шахматной партии;
  - л) проведение физического опыта;
  - м) уравнение химической реакции;
  - н) чучело птицы;
  - о) сценарий проведения праздника;
  - п) фотография местности;
  - р) глобус?
2. К какому виду относятся приведенные информационные модели:
  - а) график дежурства;
  - б) сценарий пьесы;
  - в) формула закона Ома;
  - г) ноты новой песни;
  - д) реклама по телевидению;
  - е) словесная формулировка теоремы Пифагора;
  - ж) уравнение химической реакции;
  - з) таблица результатов чемпионата Украины по баскетболу;
  - и) автобиография?
3. Создайте информационную модель объектов «цветок» и «арбуз» с точки зрения художника, медика, ботаника, поставщика, покупателя.
4. Создайте информационную модель объекта «дачный дом» с точки зрения художника, строителя, покупателя.
5. Расстояние между городами  $A$  и  $B$  равно 50 км. Из этих городов одновременно навстречу друг другу выехали два велосипедиста со скоростями 15 км/ч и 12 км/ч соответственно. Создайте математическую модель для определения расстояния между ними в любой момент времени до их встречи.
6. Один оператор может набрать рукопись за 20 ч, а второй - за 30 ч. Создайте математическую модель для определения части рукописи, которую они наберут через  $x$  ч, если будут работать одновременно.



7\*. Назовите процессы, математической моделью которых является линейная функция; квадратичная функция; обратная пропорциональность.

## 2.2. Алгоритмы и их свойства. Формы представления алгоритмов



1. Какие предложения называются побудительными?
2. Готовили ли вы какое-либо блюдо, пользуясь рецептом его приготовления? Как вы это делали?
3. На каких уроках вы пользовались инструкциями? Какими именно?

### Понятие алгоритма

Люди ежедневно используют разнообразные правила, инструкции, рецепты ит. п., которые состоят из определенной последовательности команд (указаний). Некоторые из них настолько вошли в нашу жизнь, что мы исполняем их почти не задумываясь, иногда говорят, автоматически.

Так, для приготовления яичницы нужно выполнить последовательность команд:

1. Поставить сковороду на плиту.
2. Положить на сковороду кусочек сливочного масла.
3. Включить конфорку.
4. Ожидать, пока масло на сковороде растает.
5. Разбить два яйца и вылить их содержимое на сковороду.
6. Посолить.
7. Ожидать, пока загустеет белок.
8. Выключить конфорку.

А для того чтобы определить вид треугольника по его углам, если известны его три стороны, нужно выполнить такую последовательность команд:

1. Определить сторону треугольника, которая не меньше каждой из двух других.
2. Вычислить косинус угла треугольника, который лежит против стороны, определенной как результат выполнения команды 1.
3. Если косинус угла отрицательный, то сообщить «треугольник тупоугольный», если нет, то если косинус угла равен нулю, то сообщить «треугольник прямоугольный», если нет, то сообщить «треугольник остроугольный».

Такие последовательности команд (указаний) называют алгоритмами.



*Алгоритм* — это конечная последовательность команд (указаний), которая определяет, какие действия и в каком порядке нужно выполнить, чтобы достичь поставленной цели.

Каждая команда алгоритма является побудительным предложением, которое указывает, какое действие должен выполнить исполнитель алгоритма. Исполнителем алгоритма может быть человек, животное, автоматическое устройство, такое как робот, компьютер и т. п.

Множество всех команд, которые может выполнять данный исполнитель, называют системой команд этого исполнителя. Например, в систему команд исполнителя, который будет исполнять второй из приведенных выше алгоритмов, должны входить такие команды:



1. Сравнить длины сторон треугольника и выбрать из них не меньшую каждой из двух других.
2. Вычислить косинус угла треугольника по известным трем сторонам.
3. Сравнить число с нулем (больше нуля, меньше нуля или равно нулю).
4. Вывести сообщение.

*Обращаем ваше внимание:* разрабатывая алгоритм, нужно прежде всего определить, для какого исполнителя он предназначен, и использовать в алгоритме только те команды, которые входят в систему команд этого исполнителя.



Слово **алгоритм** происходит от имени выдающегося ученого средневекового Востока **Мухаммеда бен Мусы аль-Хорезми** (783-850) (рис. 2.2), который в своих научных трудах по математике, астрономии и географии описал и использовал индийскую позиционную систему счисления, а также сформулировал в общем виде правила выполнения четырех основных арифметических действий. Европейские ученые ознакомились с его трудами благодаря переводам их на латынь. При переводе имя автора было подано как **Algorithmus**. Отсюда и пошло слово **алгоритм**. А разработанные им правила выполнения арифметических действий считают первыми алгоритмами.

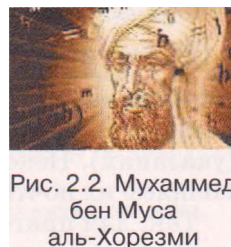


Рис. 2.2. Мухаммед бен Муса аль-Хорезми

### Свойства алгоритма

Свойствами алгоритма являются дискретность, определенность, исполняемость, конечность, результативность и массовость.

**Дискретность** (лат. *discretus* - разделенный, прерывистый) алгоритма означает, что его исполнение сводится к выполнению отдельных действий (шагов) в определенной последовательности. Причем каждая команда алгоритма должна выполняться за конечный интервал времени.

**Определенность** (или **детерминированность** (лат. *determinans* - определяющий)) алгоритма означает, что для заданного набора значений начальных (входных) данных алгоритм однозначно определяет порядок действий исполнителя и результат этих действий. Алгоритм не должен содержать команды, которые могут восприниматься исполнителем неоднозначно, например «Взять 2-3 ложки сахара», «Немного подогреть молоко» и т. п. Кроме того, в алгоритмах недопустимы ситуации, когда после выполнения очередной команды исполнителю непонятно, какую команду он должен выполнять следующей.

**Исполняемость** алгоритма означает, что алгоритм, предназначенный для определенного исполнителя, может содержать только команды, которые входят в систему команд этого исполнителя. Например, алгоритм для исполнителя «Ученик первого класса» не может содержать команду «Построй биссектрису данного угла», хотя такая команда может быть в алгоритме, предназначенном для исполнителя «Ученик восьмого класса».

Отметим, что исполнитель должен лишь уметь выполнять каждую команду из своей системы команд, и не важно, понимает он ее или нет. Говорят, что исполнение алгоритмов исполнителем имеет формальный характер: исполнитель может не понимать ни одну из команд, может не знать цели выполнения алгоритма, и все равно получит результат. Например, станок с программным управлением не понимает ни одной из команд, которые он выполняет, но благодаря своей конструкции успешно изготавливает детали.

Конечность алгоритма означает, что его исполнение закончится после конечного (возможно, достаточно большого) числа шагов и за конечное время при любых допустимых значениях начальных данных. Приведенные выше последовательности команд являются конечными, а следующая последовательность команд - бесконечная:

1. Взять число 2.
2. Умножить взятое число на 10.
3. Прибавить к полученному числу 5.
4. Если полученное число положительное, то выполнить команду 3, если нет, то прекратить выполнение алгоритма.

Результативность алгоритма означает, что по окончании его исполнения обязательно получают результаты, соответствующие поставленной цели. Результативными считаются также алгоритмы, которые определяют, что данную задачу нельзя решить или данная задача не имеет решений при заданном наборе начальных данных.

Массовость алгоритма означает, что алгоритм может быть применен к целому классу однотипных задач, для которых общими являются условие и ход решения и которые отличаются только начальными данными.

Таким, например, есть алгоритм решения квадратного уравнения, который дает возможность найти действительные корни квадратного уравнения с произвольными действительными коэффициентами или определить, что при данных значениях коэффициентов уравнение не имеет действительных корней. Массовым также является, например, алгоритм построения биссектрисы произвольного угла с использованием циркуля и линейки.

Однако, кроме массовых алгоритмов, создаются и применяются алгоритмы, которые не являются массовыми. Таким, например, является алгоритм решения конкретного квадратного уравнения (например,  $2x^2 + 5x + 2 = 0$ ) или алгоритм приготовления конкретного салата (например, греческого) на конкретное количество людей.

## Формы представления алгоритмов

Рассмотрим алгоритм решения такой задачи.

**Задача 1.** Есть сосуд вместимостью 8 л, который заполнен жидкостью, и два пустых сосуда вместимостью 5 л и 3 л. Нужно получить в одном из сосудов 1 литр жидкости и сообщить, в каком.

Рассмотрим исполнителя, имеющего такую систему команд:

1. Перелить жидкость из одного сосуда в другой.
2. Наполнить один сосуд жидкостью из другого сосуда.
3. Вывести сообщение.

Для этого исполнителя алгоритм решения этой задачи будет таким:

1. Наполнить 3-литровый сосуд из 8-литрового.
2. Перелить жидкость из 3-литрового сосуда в 5-литровый.
3. Наполнить 3-литровый сосуд из 8-литрового.
4. Наполнить 5-литровый сосуд из 3-литрового.
5. Вывести сообщение: «1 л получен в 3-литровом сосуде».

Приведенный алгоритм, как и рассмотренные ранее, записан в виде последовательности команд, каждая из которых имеет свой порядковый но-

# Глава 2





мер и записана на русском языке, то есть на языке человеческого общения. Такая форма представления алгоритма называется словесной.

В таком алгоритме команды выполняются последовательно, то есть после выполнения каждой команды выполняется команда, записанная следующей.

Для более сложных алгоритмов словесная форма представления алгоритма не всегда является удобной и наглядной. Поэтому, кроме такой формы представления, часто используют графическую форму представления алгоритма, или представление алгоритма в виде блок-схемы.

В блок-схеме алгоритма каждая команда записывается в геометрической фигуре (блоке) определенного вида. Блоки соединяются между собой стрелками, указывающими направление перехода для выполнения следующей команды алгоритма. Приведем некоторые элементы (блоки) блок-схемы алгоритма (табл. 2.1).

Таблица 2.1. Некоторые элементы (блоки) блок-схемы алгоритма

Наименование	Обозначение	Назначение
Терминатор		Начало или конец алгоритма
Процесс		Выполнение одной или нескольких команд
Данные		Ввод входных данных (аргументов) или вывод выходных данных (результатов)
Решение		Принятие решения в зависимости от результата проверки указанного условия

Блок-схема алгоритма решения задачи 1 представлена на рисунке 2.3.



Характерными чертами этого алгоритма является то, что все его команды выполняются в записанной последовательности, каждая команда алгоритма обязательно выполняется, причем только один раз. Такие алгоритмы (или фрагменты алгоритма) называются *линейными*.

Рассмотрим еще одну задачу - задачу на вычисление значения выражения, и составим алгоритм ее решения.

Задача 2. Вычислить значение выражения  $(a - b) * (c - d)$ , где  $a, b, c, d$  - действительные числа (символом  $*$  в информатике обозначается операция умножения).

Рассмотрим исполнителя, который умеет получать значения переменных, выполнять арифметические операции над действительными числами, запоминать результаты выполнения этих операций и сообщать эти результаты. Передача данных исполнителю называется вводом данных, а сообщение исполнителем результатов исполнения алгоритма - выводом данных (результатов).

Запишем алгоритм решения этой задачи для такого исполнителя в словесной форме.



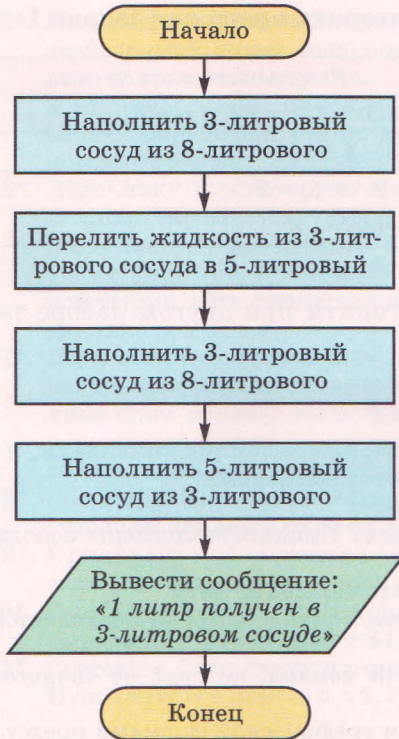


Рис. 2.3. Блок-схема алгоритма решения задачи 1

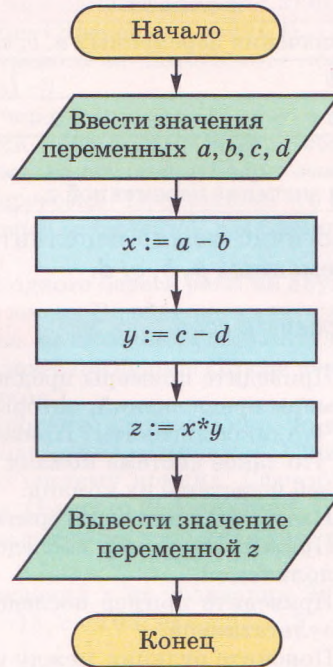


Рис. 2.4. Блок-схема алгоритма решения задачи 2

1. Ввести значения переменных  $a, b, c, d$  (в результате выполнения этой команды исполнитель запоминает введенные данные как значения соответствующих переменных).
2. Вычислить значение выражения  $a - b$  и результат присвоить переменной  $x$  (запомнить как значение этой переменной).
3. Вычислить значение выражения  $c - d$  и результат присвоить переменной  $y$ .
4. Вычислить значение выражения  $x * y$  и результат присвоить переменной  $z$ .
5. Вывести значение переменной  $z$ .

В командах 2, 3 и 4 вычисляется значение выражения, и результат вычисления присваивается определенной переменной (запоминается как значение определенной переменной). Такие команды называются командами присваивания. Для них удобно использовать такую форму записи:

2.  $x := a - b$  (читается: переменной  $x$  присвоить значение выражения  $a - b$ ).
3.  $y := c - d$ .
4.  $z := x * y$ .

Знак  $:=$  называется знаком присваивания и состоит из двух символов: двоеточие и равно, которые записываются без пробела между ними.

Блок-схема этого алгоритма представлена на рисунке 2.4.

Проиллюстрируем выполнение этого алгоритма для значений переменных  $a = 3, b = 4, c = -2, d = -5$  (табл. 2.2).

Таблица 2.2. Пример исполнения алгоритма решения задачи 1

Команда	Результат выполнения
Ввести значения переменных $a, b, c, d$	$a = 3, b = 4, c = -2, d = -5$
$x := a - b$	$x = 3 - 4 = -1$
$y := c - d$	$y = -2 - (-5) = 3$
$z := x * y$	$z = -1 * 3 = -3$
Вывести значение переменной $z$	$z = -3$

Аналогично можно исполнить этот алгоритм при другом наборе значений переменных  $a, b, c, d$ .

## Проверьте себя

- 1°. Приведите примеры предложений, которые являются командами, и примеры предложений, которые не являются командами.
- 2°. Что такое алгоритм? Приведите примеры.
- 3°. Что такое система команд исполнителя? Приведите примеры исполнителей и системы их команд.
- 4°. Назовите свойства алгоритма. Поясните каждое из них.
- 5\*. Приведите пример последовательности команд, которая не является исполняемой.
- 6\*. Приведите пример последовательности команд, которая не является результативной.
- 7°. Поясните отличия между словесной и графической формами представления алгоритма.
- 8°. Назовите элементы блок-схемы алгоритма и поясните их назначение.
- 9°. Какой алгоритм (фрагмент алгоритма) называется линейным?
- 10\*. Приведите примеры исполнителей из окружающей среды и их системы команд. Приведите примеры алгоритмов, которые исполняет каждый из них.
- 11°. Что такое команда присваивания? Как она записывается?

## Выполните задания

1. Укажите команды среди приведенных предложений:
  - а) Закрой окно.
  - б) Который час?
  - в)  $3 + 2 = 5$ .
  - г) Не мешай читать.
  - д) Если идет дождь, возьми зонтик.
  - е) Я живу в Киеве.
- 2\*. Сформулируйте линейные правила-алгоритмы, изучавшиеся на уроках:
  - а) русского языка;
  - б) математики;
  - в) других предметов.
 Представьте эти алгоритмы в виде блок-схем.
- 3°. Исполните алгоритм:

1. Начертить отрезок  $AB$  длиной 5 см.
  2. Поставить острие циркуля в точку  $A$ .
  3. Построить окружность, радиус которой равен длине отрезка  $AB$ .
  4. Поставить острие циркуля в точку  $B$ .
  5. Построить окружность, радиус которой равен длине отрезка  $AB$ .
  6. Провести прямую через точки пересечения построенных окружностей. Какое бы название вы дали этому алгоритму? Какие геометрические задачи можно решать по этому алгоритму? Составьте его блок-схему.
- 4°. Составьте алгоритм приготовления вашего любимого блюда. Представьте его в словесной форме.

5. Есть координатная прямая с обозначенными на ней целыми числами. На этой прямой живет исполнитель Стрекоза, которая умеет перемещаться по ней, выполняя команды: 1) прыгни на 3 единицы вправо; 2) прыгни на 2 единицы влево. Начальное положение Стрекозы - точка 0. Составьте блок-схему алгоритма, по которому Стрекоза за наименьшее количество прыжков окажется в точке: а) 24; б) 7; в) -3.
6. Лодочнику нужно перевезти в лодке через реку волка, козу и капусту. В лодке, кроме лодочника, помещается либо только волк, либо только коза, либо только капуста. На берегу нельзя оставлять козу с волком или козу с капустой. Составьте алгоритм перевозки. Представьте его в словесной форме. (Эта старинная задача впервые встречается в математических рукописях VIII в.)
7. Двум солдатам нужно переправиться с одного берега реки на другой. Они увидели двух мальчиков на маленькой лодке. В ней могут переправляться либо один солдат, либо один или двое мальчиков. Составьте алгоритм переправки солдат. Представьте его в словесной форме. (После переправки солдат лодка должна остаться у мальчиков.)
8. Составьте блок-схему алгоритма построения треугольника по трем его сторонам длиной 5 см, 6 см и 4 см с использованием циркуля и линейки.
- 9\*. Придумайте исполнителя. Задайте его систему команд. Сформулируйте задачу и составьте алгоритм ее решения для этого исполнителя.
10. Составьте блок-схему алгоритма вычисления на калькуляторе значения выражения  $(81 - 12)(58 + 84)$ .
11. Составьте блок-схему алгоритма нахождения  $x$  из уравнения  $2x + a = c$ . Исполните его при: а)  $a = 5, c = 7$ ; б)  $a = -15, c = 105$ ; в)  $a = 5, c = 5$ .

### 2.3. Компьютерные программы и языки программирования. Этапы решения задач с использованием компьютера



1. Для чего предназначено и из чего состоит программное обеспечение компьютеров?
2. Что такое алгоритм? Что такое система команд исполнителя алгоритма? В чем суть формальности исполнения алгоритма исполнителем?
3. В чем заключается процесс кодирования сообщений? Для чего оно используется? В чем суть двоичного кодирования?
4. Что такое модель объекта? Какие виды моделей вы знаете?
5. Назовите основные отрасли применения современных компьютеров.

#### Компьютерные программы

Вы знаете, что компьютер работает под управлением программного обеспечения, которое состоит из компьютерных программ (далее - программ) разнообразного назначения. Работая с компьютером в школе и дома, вы использовали текстовый процессор и графический редактор, архиваторы и антивирусные программы, табличный процессор и редактор компьютерных презентаций, учебные и контролирующие программы, игровые программы и т. п.



*Программа* — это набор команд (указаний, инструкций), предназначенный для исполнения компьютером в определенной последовательности.



Программы создаются для исполнения компьютером алгоритмов. Эти алгоритмы образуют логику программы (программную логику).

В процессе своей работы программа обрабатывает данные. Данные, которые вводит в программу непосредственно пользователь программы либо программа их получает от определенного устройства (например, от датчика температуры), либо от другой программы, либо из другого источника (например, из текстового файла), называются входными (начальными) данными. Некоторые программы работают без входных данных. Данные, получение которых является целью использования программы, называются выходными (результатирующими) данными. Во время выполнения программы образуются и обрабатываются и другие данные, которые называются промежуточными данными.

Большинство современных программ в процессе своей работы предоставляют пользователю определенный набор средств для взаимодействия с программой и устройствами. К этим средствам относятся элементы управления (кнопки, меню и др.), средства ввода данных (поля, счетчики и др.), средства вывода данных (надписи, поля и т. д.) и др. Совокупность таких средств, а также методов их использования образует интерфейс пользователя.



Первым в мире программистом считается Ада Лавлейс (1815-1852), дочь известного английского поэта Джорджа Гордона Байрона. Она работала с Чарльзом Беббиджем (1791-1871), разработчиком механической вычислительной машины (**аналитической машины**), и впервые описала основные принципы разработки программ для вычислительных машин. К сожалению, эта вычислительная машина так и не была создана. В честь Ады Лавлейс один из языков программирования назван **Ada**.

### Языки программирования

Составляя алгоритмы, предназначенные для исполнения человеком, пользуются языком общения людей: украинским, русским, английским, немецким и др.

Но для алгоритмов, которые должно исполнять автоматическое устройство (в частности, компьютер), язык общения людей сложен, имеет неоднозначные конструкции (например, омонимы). Поэтому для записи алгоритмов, предназначенных для исполнения автоматическими устройствами, разрабатывают и используют специальные языки - языки программирования.



**Язык, который используется для записи алгоритмов, предназначенных для исполнения компьютером, называется языком программирования.**

Каждый язык программирования имеет такие компоненты:

- 1) алфавит - множество символов, из которых можно образовывать слова и предложения этого языка;
- 2) словарь - набор специальных (зарезервированных, ключевых) слов;
- 3) синтаксис - правила составления и записи языковых конструкций (не словарных слов и предложений);
- 4) семантику - установленное однозначное толкование языковых конструкций, правил их исполнения.

Использование символов, не входящих в алфавит, неправильное написание словарных слов, нарушение синтаксических правил приводит к невозможности выполнения компьютером соответствующей команды. Такие нарушения называются синтаксическими ошибками.

За последние 70 лет создано около трех тысяч различных языков программирования. Некоторые из них уже не используются, для некоторых постоянно появляются более совершенные версии, каждая следующая из которых более удобная для составления программ и имеет более широкие возможности, постоянно создаются новые языки программирования.

Некоторые языки программирования используются с целью составления программ для решения задач из различных отраслей науки, техники, производства, сферы быта и др. (например, Delphi, C++, C#, Java), а некоторые созданы специально для составления программ для решения специального круга задач (например, Prolog (англ. *Programming in Logic* - программирование в логике), Lisp (англ. *List Processing* - обработка списков)).

Процессор компьютера может выполнять команды, представленные только машинным языком. Машинный язык - это язык программирования, в котором команды представляются последовательностями двоичных кодов. Машинный язык программирования ориентирован на процессоры конкретной архитектуры, то есть машинные языки для разных процессоров могут отличаться один от другого.

Для исполнения процессором программ, написанных не машинным языком программирования, их нужно сначала перевести на машинный язык и только потом исполнить. Для перевода программ на машинный язык созданы и используются специальные программы - компиляторы. Эти программы анализируют полностью весь текст программы на наличие синтаксических ошибок, и если такие ошибки отсутствуют, переводят текст программы на машинный язык, формируя машинный код программы. Этот код, в зависимости от режима работы компилятора, либо сохраняется в памяти компьютера, либо записывается на диск в виде исполняемого файла (например, ехе-файла). После получения исполняемого файла его можно отправить на исполнение процессором. При этом сама программа-компилятор уже не используется. Поэтому исполняемый файл может использоваться и на тех компьютерах, где нет компилятора.

При наличии в программе синтаксических ошибок компилятор либо останавливается на первой из них и выводит на экран сообщение о ней, либо анализирует программу до конца и выводит на экран общий список сообщений об имеющихся ошибках. После этого нужно исправить все синтаксические ошибки и снова начать процесс компиляции.



Для некоторых языков программирования созданы другие специальные программы - **интерпретаторы**. Эти программы не создают исполняемых файлов, а анализируют программу покомандно и сразу же эти команды выполняют. Поэтому исполнить программу, которая интерпретируется, а не компилируется, можно лишь на том компьютере, где установлена соответствующая программа-интерпретатор.

Для некоторых современных языков программирования используют комбинацию компиляции и интерпретации. Сначала программа компилируется в некоторый промежуточный код (не машинный), после чего интерпретируется специальной программой, написанной для этого кода.

## Этапы решения задач с использованием компьютера

Вам уже известно, что первые компьютеры были созданы для быстрого выполнения громоздких вычислений. А современные компьютеры способны решать не только вычислительные задачи, но и создавать и редактировать рисунки, текстовые документы, презентации, мультфильмы и видеофильмы, выбирать необходимые данные из больших массивов данных, управлять работой различных устройств и т. п. Использовать компьютер для решения некоторых из этих задач вы уже научились в 10-м классе, некоторые из них будут рассмотрены в 11-м классе.

Все эти задачи разные по своей сути, каждая из них требует своего, индивидуального подхода. Но все же ход решения каждой из задач можно разделить на несколько этапов, общих для всех задач, решаемых с использованием компьютера. Рассмотрим эти этапы на примере такой задачи.

**Задача I.** Используя компьютер, разработать проект спортивной площадки.

Для решения этой задачи сначала нужно выяснить размеры спортивной площадки, которую нужно спроектировать, какие объекты нужно на ней разместить (для данной задачи - баскетбольная и волейбольная площадка, теннисный корт, гимнастические тренажеры), в каком виде нужно представить этот проект (для данной задачи - в виде чертежей всей площадки и ее объектов). Этот этап называется *анализ условия задачи, определение начальных (входных) данных и конечных результатов*.

После этого нужно определить, какие свойства спортивной площадки и ее объектов будут отображены на плане, какие значения они будут иметь. Некоторые из этих значений являются стандартными, а некоторые мы должны определять сами. Например, размеры баскетбольных щитов, высота их нижнего края от земли, высота кольца от земли являются стандартными, а вот длину перекладин и их количество в шведской стенке, расстояния между объектами мы можем определить сами. На этом же этапе нужно установить и зависимости между объектами, например, какие площадки будут совмещены. Этот этап решения задачи называется *создание информационной модели*.

Дальше нужно выбрать средство, которое будет использовано для создания этого проекта. В данном случае это может быть одна из прикладных программ: векторный или растровый графический редактор, специальный пакет программ для построения чертежей и т. п. А можно создать собственную программу, которая по данным задачи на основе информационной модели предложит разные варианты проектов такой площадки. Этот этап называется *выбор программных средств для решения задачи*.

Если мы решим использовать одну из прикладных программ, то дальше нужно выбрать те средства этой программы, которые дадут возможность создать нужный проект, составить алгоритм его создания, учитывая систему команд соответствующей программы-исполнителя, после чего выполнить составленный алгоритм и получить компьютерную модель проекта площадки. Эти этапы называются *выбор средств в среде прикладной программы, составление алгоритма, исполнение алгоритма и получение компьютерной модели объекта*.

Если же мы решим составить собственную программу для решения этой задачи, то прежде всего необходимо выбрать язык программирования. После этого нужно:



1. Составить алгоритм решения задачи, учитывая то, что алгоритм будет выполняться компьютером.
2. Записать составленный алгоритм на выбранном языке программирования и разработать интерфейс пользователя программы - средства взаимодействия пользователя с программой (окна, кнопки, меню, поля, переключатели, флажки и другие элементы управления, их расположение и использование), то есть составить программу.
3. Запустить составленную программу на исполнение, ввести входные данные и получить конечные результаты.

Эти этапы называются *выбор языка программирования, составление алгоритма решения задачи, составление программы, исполнение программы и получение компьютерной модели объекта.*

Во время исполнения алгоритма или программы может случиться так, что одну из команд алгоритма или программы исполнитель не сможет выполнить. Это значит, что алгоритм или программа содержат ошибку. Нужно определить все имеющиеся ошибки (они называются синтаксическими ошибками) и устранить их. Этот этап называется *отладка алгоритма или программы.* После отладки нужно опять исполнить алгоритм или программу решения данной задачи.

После того как мы получим результат, используя прикладную программу или специально созданную программу, его нужно исследовать на соответствие условию поставленной задачи. Если все требования условия задачи выполняются, то можно считать, что задача решена правильно и мы получили правильное ее решение, то есть то, которое полностью соответствует условию задачи. Если же нет, нужно искать ошибки на предыдущих этапах решения задачи (они называются логическими ошибками), исправить их и опять искать правильное решение задачи. Этот этап называется *исследование полученных результатов на соответствие условию задачи.*

*Обращаем ваше внимание, что последний этап чаще всего проводится не на одном наборе входных данных, а на целой серии таких наборов (тестовые наборы данных), которые подбирают таким образом, чтобы как можно полнее со всех сторон исследовать полученные результаты на их реальность и соответствие условию задачи.*

Из рассмотренного примера можно выделить основные этапы решения задач с использованием компьютера, представленные на рисунке 2.5.

Проследим теперь, как реализуются рассмотренные выше этапы на примере решения другой задачи.

**Задача 2.** На торговой базе хранятся товары, которые изготовлены в Украине и импортируются в разные страны. Известны их названия и цены в гривнах. Перевести цены товаров в доллары, евро и фунты стерлингов.

**I этап.** Анализ условия задачи, определение начальных (входных) данных и конечных результатов

*Входные данные:*

- названия товаров,
- цены товаров в гривнах,
- курсы доллара, евро и фунта стерлингов относительно гривны.

*Конечные результаты:*

- цены товаров в долларах, евро и фунтах стерлингов.

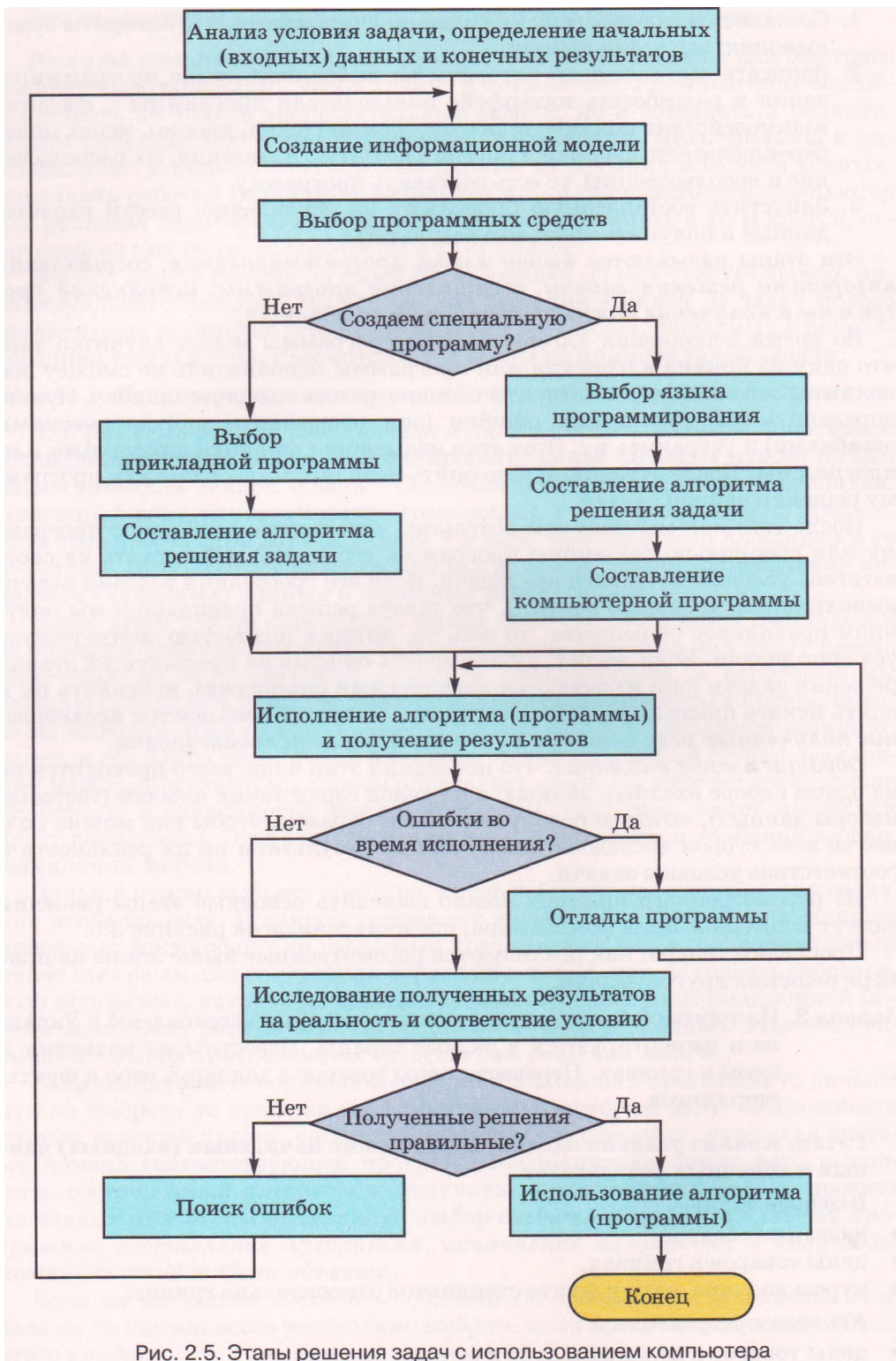


Рис. 2.5. Этапы решения задач с использованием компьютера

**II этап. Создание информационной модели**

Очевидно, информационной моделью решения этой задачи является математическая модель, состоящая из формул, по которым цена из гривен переводится в доллары, евро и фунты стерлингов.

Эти формулы будут выглядеть так:

$$D = \frac{G}{KD}, E = \frac{G}{KE}, F = \frac{G}{KF},$$

где  $G$  - цена товара в гривнах,  $D, E, F$  - цены товара соответственно в долларах, евро и фунтах стерлингов,  $KD, KE, KF$  - соответственно курсы доллара, евро и фунта стерлингов относительно гривны.

Кроме того, полученные цены желательно округлить до сотых.

**III этап. Выбор программных средств для решения задачи**

Прикладным программным средством для решения задачи может быть, например, программа-калькулятор или табличный процессор. Или можно создать специальную программу, используя один из языков программирования.

**IV этап. Составление алгоритма решения задачи**

Если будет выбрано прикладное программное средство табличный процессор Excel 2007, то алгоритм решения этой задачи может быть таким:

1. Ввести в ячейки A3:A5 текст в соответствии с таблицей 2.3.
2. Ввести в ячейки B3:B5 текущие курсы соответствующих валют.
3. Ввести в ячейки D2:H2 текст в соответствии с таблицей 2.3.
4. Ввести в ячейки столбца D, начиная с ячейкой D3, названия товаров.
5. Ввести в ячейки столбца E, начиная с ячейки E3, цены товаров в гривнах.
6. Ввести в ячейку F3 формулу =ROUND(E3/\$B\$3;2).
7. Ввести в ячейку G3 формулу =ROUND(E3/\$B\$4;2).
8. Ввести в ячейку H3 формулу =ROUND(E3/\$B\$5;2).
9. Выделить ячейки диапазона F3:H3.
10. Скопировать содержимое ячеек выделенного диапазона в соответствующие ячейки столбцов F:H.

**V этап. Получение компьютерной модели**

В итоге мы получим таблицу, фрагмент которой отображен в таблице 2.3.

*Таблица 2.3. Таблица решения задачи в табличном процессоре Excel 2007*

1	A	B	C	D	E	F	G	H
2				Товар.	Цена в гривнах	Цена в долларах	Цена в евро	Цена в фунтах стерлингов
3	Курс доллара	8,02		Миксер	115,22	14,37	9,98	8,93
4	Курс евро	11,55		Торшер	216,57	27	18,75	16,79
5	Курс фунта стерлингов	12,9		Телевизор	6615	824,81	572,73	512,79
6				Пылесос	723,25	90,18	62,62	56,07

VI этап. Исследование полученных результатов на реальность и соответствие условию

После получения итоговой таблицы мы должны проанализировать ее реальность. Например, если курс доллара к гривне 8,02, то цена в долларах должна быть приблизительно в 8 раз меньше, чем цена в гривнах. Можно также проверить несколько значений, используя другие вычислительные средства. Можно ввести в таблицу такие контрольные входные значения, для которых результат предварительно известен, например 802 грн. = = \$100. Аналогично нужно проверить работу формул и в других столбцах.

Если такое исследование обнаруживает ошибки, нужно искать причины, исправлять их и опять проходить этапы, следующие за тем, на котором оказались ошибки.

Если бы на III этапе было выбрано как средство решения этой задачи создание специальной программы, то дальше этапы решения этой задачи выглядели бы так:

IV этап. Составление алгоритма решения задачи

Алгоритм решения этой задачи, на основе которого создавалась бы компьютерная программа, мог бы быть, например, таким, как показано на рисунке 2.6.

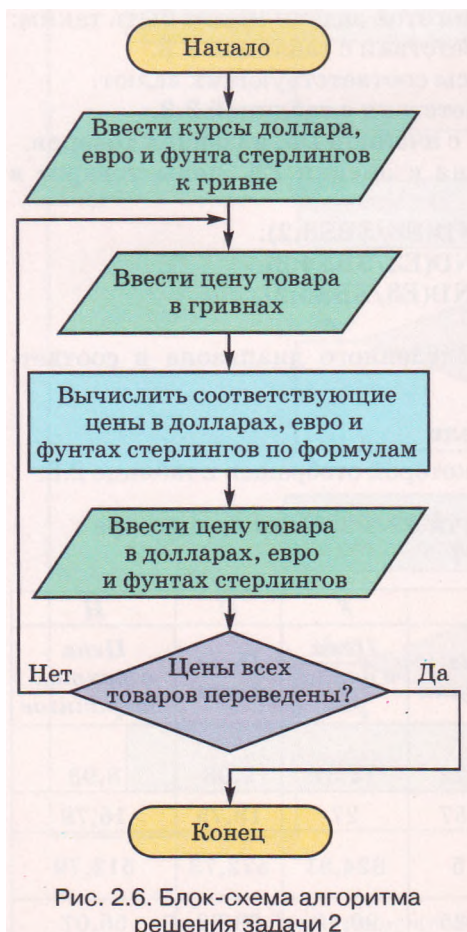


Рис. 2.6. Блок-схема алгоритма решения задачи 2

Можно составить и другие алгоритмы решения этой задачи. Например, сначала ввести цены всех товаров в гривнах, а затем их по очереди переводить в соответствующие валюты.

V этап. Составление компьютерной программы

После составления алгоритма нужно выбрать язык программирования, разработать интерфейс для работы пользователя с программой и составить программу. Этот этап мы детально изучим в следующих пунктах.

VI этап. Отладка программы, исследование результатов на реальность и соответствие условию

После создания компьютерной программы нужно ее исполнить на нескольких наборах тестовых примеров. Если результаты исполнения не соответствуют ожидаемым, то это значит, что программа содержит ошибки. Нужно обнаружить эти ошибки и исправить их.

Подбор этих тестовых примеров является достаточно непростым и важным делом и должен соответствовать целому ряду требований. Самые главные из них такие:




1. Тестовые примеры должны быть такими, чтобы можно было легко определить конечные результаты и сравнить их с теми, которые выдает компьютерная программа.
2. Набор этих примеров должен быть таким, чтобы он охватывал все ситуации, которые могут случиться во время использования программы для реальных данных.




### VII этап. Использование программы

После того как мы удостоверимся, что для всех тестовых примеров программа работает правильно, можно исполнить ее для всех заданных значений входных данных и получить необходимые результаты.

#### Проверьте себя

- 1°. Что такое компьютерная программа?
- 2°. Что такое программная логика и интерфейс пользователя?
- 3°. Какие данные называют входными, выходными, промежуточными?
- 4°. Что такое язык программирования?
- 5°. Назовите компоненты, из которых состоит язык программирования.
- 6°. Опишите каждый компонент языка программирования.
- 7°. Что такое синтаксическая ошибка?
- 8°. Что такое машинный язык программирования? Какой вид имеют команды на этом языке программирования?
- 9°. Какие программы называются компиляторами? Опишите общий принцип их работы.
-  10\*. Какие программы называются интерпретаторами? Опишите общий принцип их работы.
- 11°. Назовите и опишите каждый этап решения задач с использованием компьютера, если будет использовано прикладное программное обеспечение.
- 12°. Назовите и опишите каждый этап решения задач с использованием компьютера, если будет использована специально составленная программа.

#### Выполните задания

-  1. Найдите информацию о годах создания первых версий языков программирования Basic, PL/I, Cobol, ALGOL, Fortran, Prolog, Lisp, Java, C, C++, C#. Коротко запишите основные характерные черты каждой из них.
-  2\*. Подготовьте сообщение о языках программирования, на которых составлялись программы для компьютеров украинского производства в 1950-1960-е годы.
3. Для каждой из приведенных задач детализируйте каждый этап ее решения с использованием компьютера:
  - а°) Определить стоимость краски для покраски пола в классной комнате,
  - б°) Определить стоимость обоев для оклеивания кабинета информатики,
  - в°) Два поезда вышли одновременно из пунктов  $A$  и  $B$  навстречу друг другу. Известны расстояние между пунктами и скорости поездов. Определить расстояние между ними через 2 часа после выхода; через 3 часа. Через сколько часов они встретятся?
  -  г°) Два поезда вышли одновременно из пунктов  $A$  и  $B$  в противоположных направлениях. Известны расстояние между пунктами и скорости поездов. Определить расстояние между ними через 1 час после выхода; через 3 часа.
  - д°) В треугольнике известны длины двух сторон и мера угла между ними. Найти длину третьей стороны и его площадь.

## 2.4. Язык программирования Delphi и среда разработки Turbo Delphi 2006



1. Что такое компьютерная программа?
2. Что такое программная логика и интерфейс пользователя?
3. Что такое язык программирования? Назовите основные компоненты языка программирования. Поясните назначение каждого из них.
4. Назовите этапы решения задач с использованием компьютера. Поясните, в чем суть каждого из них.

### Язык программирования Delphi и интегрированная среда визуальной разработки проектов Turbo Delphi 2006

Мы с вами будем изучать объектно-ориентированный язык программирования Delphi. Этот язык используют как школьники для изучения основных принципов объектно-ориентированного программирования, так и программисты-профессионалы для создания больших и сложных программ.

Для создания, редактирования и отладки программ на языке программирования Delphi будем использовать интегрированную среду визуальной разработки Turbo Delphi 2006. Эта среда состоит из текстового редактора, компилятора для перевода программы с языка Delphi на машинный язык, средств отладки программы, справочной системы и других компонентов. Поэтому такая среда разработки называется интегрированной.

Эта среда разработки дает возможность использовать большую библиотеку визуальных компонентов (англ. *Visual Component Library (VCL)*). Это уже знакомые вам элементы управления: кнопки, поля, надписи, флажки, переключатели, счетчики, списки, полосы прокрутки и др. При размещении компонентов на форме автоматически генерируется соответствующий фрагмент программы на языке Delphi. Это значительно уменьшает и упрощает работу по созданию программ. Такой способ разработки называют визуальным.

Программы, созданные в таких средах, часто называют проектами. Чаще всего они состоят из нескольких файлов.



Язык программирования Delphi - это результат развития и усовершенствования языка программирования Pascal. Язык программирования Pascal был создан в 1970 году группой сотрудников Института информатики Швейцарской высшей политехнической школы под руководством профессора Никлауса Вирта (род. в 1934 г., рис. 2.7).

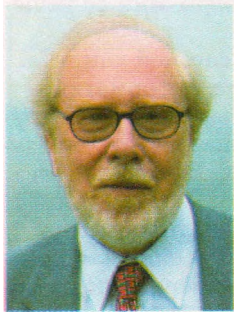


Рис. 2.7. Никлаус Вирт

Язык программирования Pascal создавался специально для обучения студентов основным принципам и методам процедурного программирования. Но впоследствии выяснилось, что этот язык программирования настолько удобен, что возникла идея применять его не только для обучения, но и для практической работы. Так появились первые компиляторы с языка программирования Pascal и среда разработки Turbo Pascal, которые со временем получили широкую известность.

С годами язык программирования Pascal развивался и совершенствовался. Новые версии этого языка, которые уже имели элементы объектно-ориентированного языка программирования, появились во второй половине 1980-х годов. Неко-

торые версии этого языка получили название Object Pascal. В начале 2000-х годов очередная новая версия Object Pascal получила название Delphi.

## Начальное окно среды Turbo Delphi 2006

Запустить Turbo Delphi 2006 можно традиционными способами:

- выполнить Пуск => Все программы => Borland Developer Studio 2006 => Turbo Delphi;

использовать ярлык  на Рабочем столе;

дважды щелкнуть на значке файла с расширением имени bdsproj.

В первых двух случаях открывается начальное окно Turbo Delphi 2006 (рис. 2.8). В верхней части этого окна размещены традиционные Строка меню и Панель инструментов. В левой и правой частях этого окна расположены подчиненные окна:

**Structure** (англ. *structure* - структура) - в нем отображается структура компонентов (элементов управления) проекта;

**Object Inspector** (англ. *object inspector* - инспектор объекта) - в нем отображаются свойства и методы текущего компонента проекта;

**Project Manager** (англ. *project manager* - управляющий проектом) - в нем отображается структура файлов проекта;

**Tool Palette** (англ. *tool palette* - палитра инструментов) - в нем отображается список объектов, доступных для использования в текущем режиме.

Рядом с кнопками закрытия этих окон расположены кнопки их сворачивания. Они похожи на обычные канцелярские кнопки. При выборе

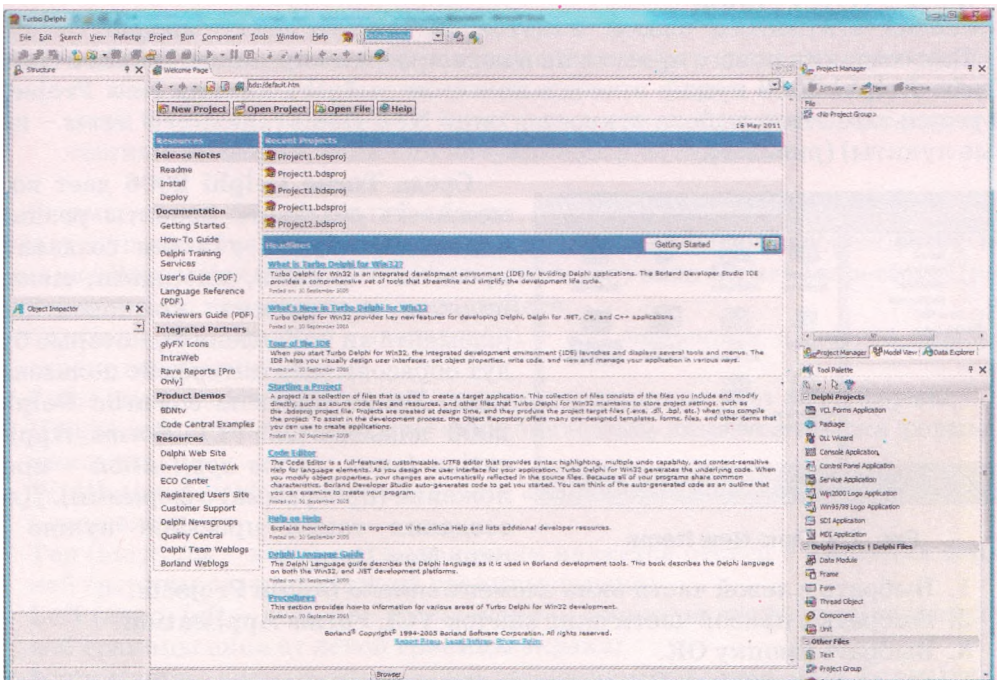



Рис. 2.8. Начальное окно среды Turbo Delphi 2006

этих кнопок они изменяют свой вид на такой , а сами окна сворачиваются. Если эти окна закрыть, то повторно открыть их можно, выполнив соответствующие команды меню View (англ. *view* - вид).

Иногда в начальном окне Turbo Delphi 2006 могут быть открыты другие подчиненные окна. Это зависит от режима просмотра этого окна, который устанавливается в Строке меню в поле с открывающимся списком. Для нашей последующей работы нужно, чтобы был установлен режим Default Layout (англ. *default layout* - расположение по умолчанию) или Classic Undocked (англ. *classic undocked* - классический отстыкованный).

В центральной части открыта вкладка Welcome Page (англ. *welcome page* - страница приглашения или стартовая страница). На этой вкладке расположены:

- кнопка New Project (англ. *new project* - новый проект) - для открытия заготовки нового проекта;
- кнопка Open Project (англ. *open project* - открыть проект) - для открытия проекта, записанного на носителе;
- кнопка Help (англ. *help* - помощь) - для открытия системы помощи и др.

Кнопка закрытия вкладки находится в правой части строки с именем вкладки. Рядом с ней находится кнопка открытия списка открытых вкладок.

Если выбрать третий из перечисленных выше способов запуска Turbo Delphi 2006, то начальное окно среды не открывается, а сразу открывается главное окно среды и в нем открывается соответствующий проект.

## Создание, исполнение и сохранение нового проекта в среде Turbo Delphi 2006

Для создания нового проекта на языке программирования Delphi в среде Turbo Delphi 2006 нужно в начальном окне выбрать кнопку New Project. В результате этого выбора откроется окно New Items (англ. *new items* - новые пункты) (рис. 2.9).

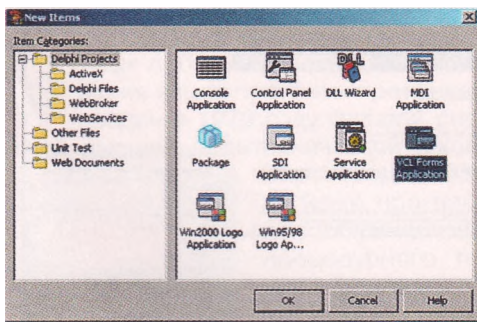


Рис. 2.9. Окно New Items

Среда Turbo Delphi 2006 дает возможность создавать проекты разных видов. Мы будем учиться создавать проекты с окнами, кнопками, меню, полями и другими компонентами (элементами управления), которые будут образовывать интерфейс пользователя. Такие проекты в Turbo Delphi 2006 называются VCL Forms Application (англ. *forms application* - приложение (программа) с формами). Для создания таких проектов нужно в окне New Items:

1. Выбрать в левой части окна элемент списка Delphi Projects.
2. Выбрать в правой части окна значок VCL Forms Application.
3. Выбрать кнопку ОК.

После выбора кнопки ОК открывается главное окно среды Turbo Delphi 2006 (рис. 2.10).



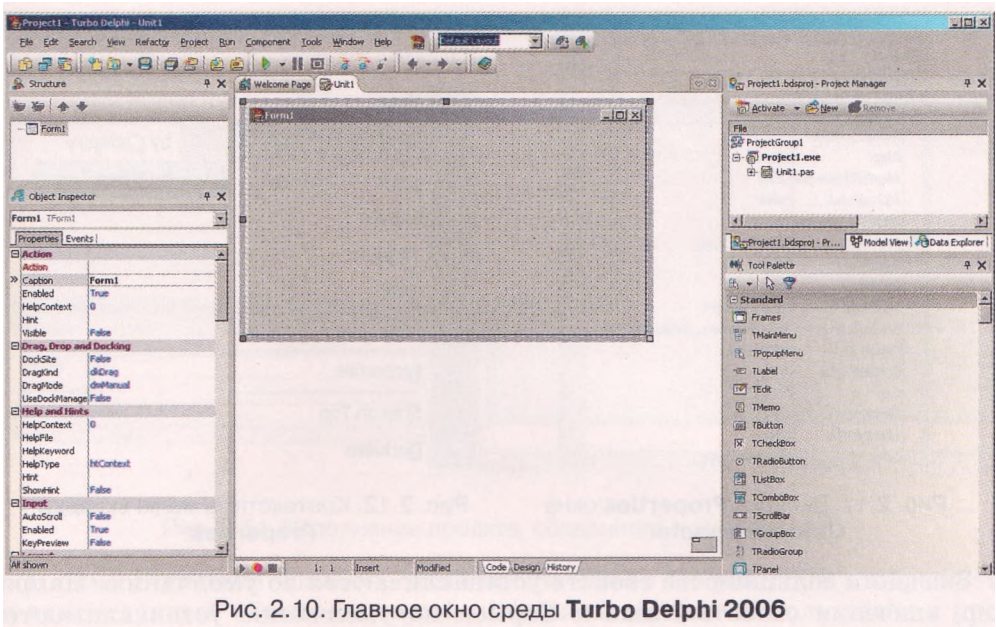


Рис. 2.10. Главное окно среды Turbo Delphi 2006

В центральной части этого окна находится открытая вкладка Unit1 (англ. *unit* - блок, часть), которая имеет три внутренние вкладки: Code (англ. *code* - код), Design (англ. *design* - дизайн, проект, план), History (англ. *history* - история). По умолчанию открывается вкладка Design, на которой расположена форма.



**Форма** — это объект, на котором можно разместить кнопки, поля, надписи, меню и другие компоненты для создания интерфейса пользователя. При исполнении проекта форма открывается как окно Windows, на котором будут расположены компоненты, размещенные на форме, если они визуальные и не скрытые.

Форма, как и любой другой объект, имеет свойства. Если форма является текущим объектом, то ее свойства и их значения отображаются на вкладке Properties (англ. *properties* - свойства) окна Object Inspector. Приведем примеры некоторых из этих свойств:

- **Caption** (англ. *caption* - заголовок) - его значением является текст в строке заголовка формы;
- **Color** (амер. *color* - цвет) - его значением является цвет фона формы;
- **Height** (англ. *height* - высота) - его значением является высота формы в пикселях;
- **Width** (англ. *width* - ширина) - его значением является ширина формы в пикселях;
- **Top** (англ. *top* - верх) - его значением является отступ в пикселях верхней границы окна от верхней границы экрана;
- **Left** (англ. *left* - левый) - его значением является отступ в пикселях левой границы окна от левой границы экрана;
- **Name** (англ. *name* - имя) - его значением является имя формы как объекта.

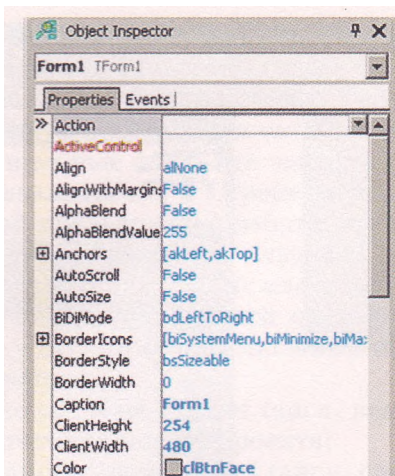


Рис. 2.11. Вкладка **Properties** окна **Object Inspector**

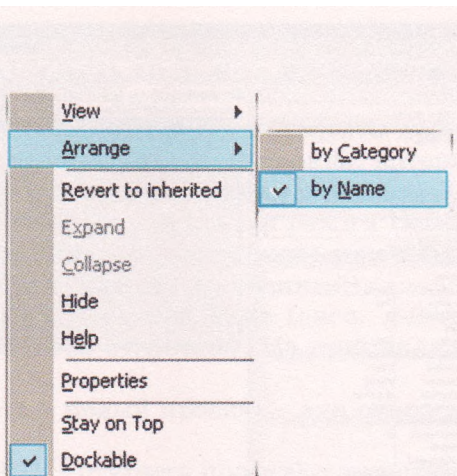


Рис. 2.12. Контекстное меню вкладки **Properties**

Значения большинства свойств устанавливаются по умолчанию. Например, значения свойств `Name` и `Caption` по умолчанию устанавливаются `Form1`, значение свойства `Height` - 338, значение свойства `Width` - 651. Эти значения, как и значения других свойств, можно изменить или установить на вкладке `Properties`.

Список свойств может быть упорядочен по категориям (рис. 2.10) или по именам (рис. 2.11). Изменить порядок расположения свойств в списке можно, используя команду `Arrange` контекстного меню вкладки `Properties` (англ. *arrange* - расставлять в определенном порядке, *by category* - по категориям, *by name* - по именам) (рис. 2.12). Советуем вам для удобства поиска необходимых свойств упорядочивать список свойств по именам.

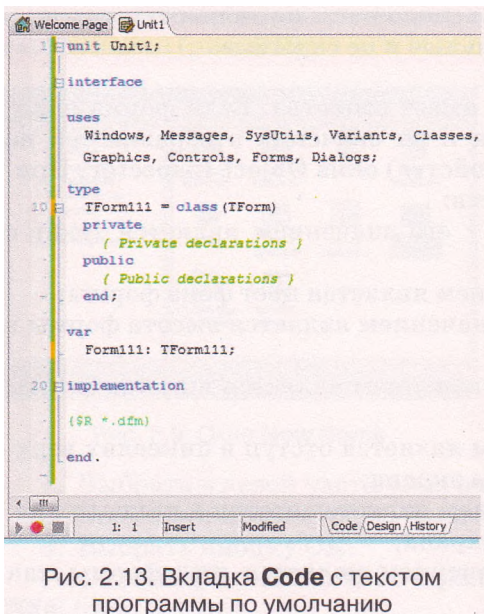


Рис. 2.13. Вкладка **Code** с текстом программы по умолчанию

Открыв вкладку `Code`, можно просмотреть текст программы, который создается по умолчанию в среде Turbo Delphi 2006 во время создания нового проекта (рис. 2.13). Даже этого текста программы достаточно, чтобы исполнить созданный по умолчанию проект. Для исполнения проекта можно использовать команду `Run` (англ. *run* - выполнение, запуск) меню `Run` или кнопку `Run` панели инструментов.

Во время исполнения проекта, который создается по умолчанию, на экране открывается пустое окно, значения свойств которого (цвет, размеры, положение на экране, текст в строке заголовка и др.) установлены по умолчанию (рис. 2.14).

Для завершения исполнения проекта нужно закрыть его окно.



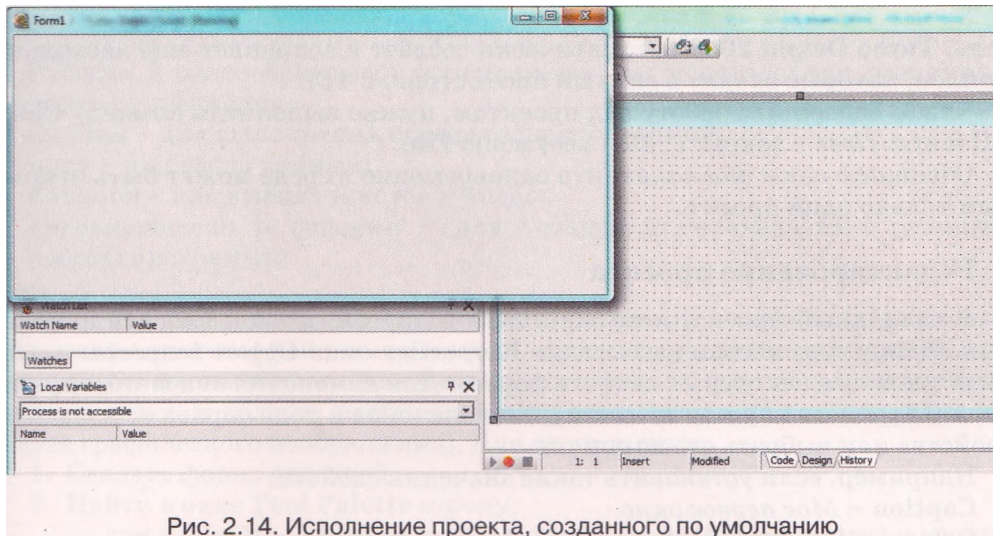


Рис. 2.14. Исполнение проекта, созданного по умолчанию

В окне Structure главного окна среды Turbo Delphi 2006 отображается единственный объект, который на данный момент используется в проекте - форма с именем *Form1*, в окне Tool Palette - список всех доступных компонентов, которые можно использовать в проекте, а окно Project Manager содержит список некоторых файлов, входящих в состав проекта (см. рис. 2.10).

Для сохранения всех файлов проекта нужно использовать команду Save All (англ. *save all* - сохранить все) меню File (англ. *file* - файл) или кнопку Save All панели инструментов.

В открывшемся окне целесообразно прежде всего для каждого проекта создать отдельную папку, в которой и сохранить файлы проекта. Turbo Delphi 2006 последовательно предлагает сохранить два файла проекта:

- файл с именем Unit1 и расширением имени pas;
- файл с именем Project1 и расширением имени bdsproj.

Расширение имен изменять не стоит, а имена файлов можно изменить.

Обращаем ваше внимание, что в Turbo Delphi 2006 имя файла может содержать только буквы английского алфавита, цифры и знак подчеркивания, и цифра не должна быть первым символом имени.

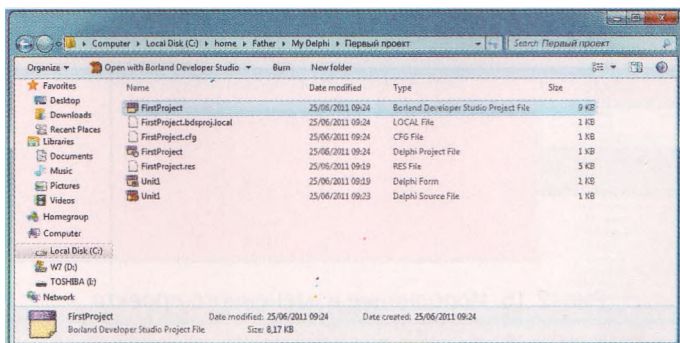


Рис. 2.15. Папка **Первый проект** с файлами проекта

Кроме тех двух файлов, которые нам предлагается записать на носитель, Turbo Delphi 2006 автоматически создает и сохраняет еще несколько файлов, которые входят в данный проект (рис. 2.15).

Чтобы завершить работу над проектом, нужно выполнить команду Close All (англ. *close* - закрыть, *all* - все) меню File.

Обращаем ваше внимание, что одновременно в среде может быть открытым только один проект.

### Редактирование проекта

В начальный проект, а также в любой другой, можно вносить изменения. Например, можно на вкладке Properties окна Object Inspector изменить значения некоторых свойств формы. Для изменения значения определенного свойства нужно ввести нужное значение в поле справа от названия свойства или выбрать его из списка.

Например, если установить такие значения свойств:

- **Caption** = *Мое первое окно*
- **Color** = *clRed* (амер. *color* - цвет, *red* - красный)
- **Height** = *450*
- **Width** = *700*
- **Left** = *600*
- **Top** = *350*

и исполнить измененный проект, то откроется окно другого цвета, других размеров, расположенное в другом месте экрана, с другим текстом в строке заголовка (рис. 2.16).

Обращаем ваше внимание, что изменить значения свойств **Height** и **Width** формы можно перетягиванием маркеров на ее границе.

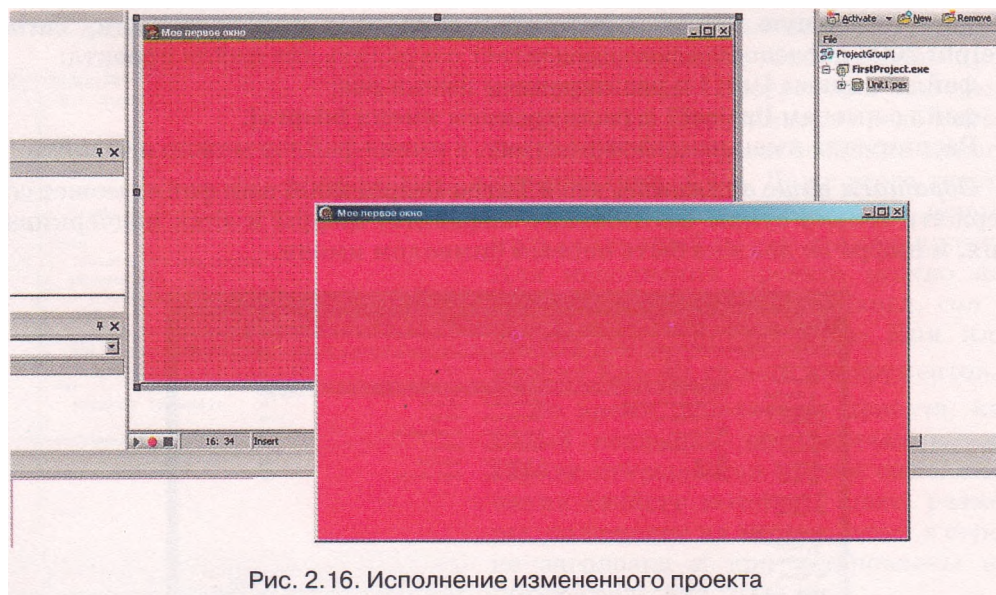


Рис. 2.16. Исполнение измененного проекта



## Размещение элементов управления на форме

Работая с разнообразными программами, вы использовали различные элементы управления:

- кнопки - для выполнения определенных операций;
- поля - для ввода данных;
- надписи - для вывода текстов и чисел;
- переключатели и флажки - для установки определенных режимов работы программы

и др.

Для того чтобы во время исполнения программы элементы управления образовывали интерфейс пользователя, нужно перед запуском программы разместить их на форме. Рассмотрим, как разместить на форме кнопку, поле, надпись и область изображения (предназначена для размещения на форме графического изображения). Для этого нужно:

1. Сделать форму текущей.
2. Найти в окне Tool Palette строку:
  - для кнопки - TButton (англ. *type* - тип; *button* - кнопка) в группе Standard (англ. *standard* - стандарт);
  - для поля - TEdit (англ. *edit* - редактировать) в группе Standard; для надписи - TLabel (англ. *label* - метка, надпись) в группе Standard;
  - для области изображения - TImage (англ. *image* - изображение) в группе Additional (англ. *additional* - дополнительный).
3. Дважды щелкнуть на соответствующей строке.

После исполнения этого алгоритма выбранный объект появляется в центральной части формы. Далее можно перетягиванием разместить объект в другом месте формы (рис. 2.17). При размещении нового элемента управления на форме в окне Structure изменяется структура элементов управления, используемых в проекте, - значок этого объекта и его имя

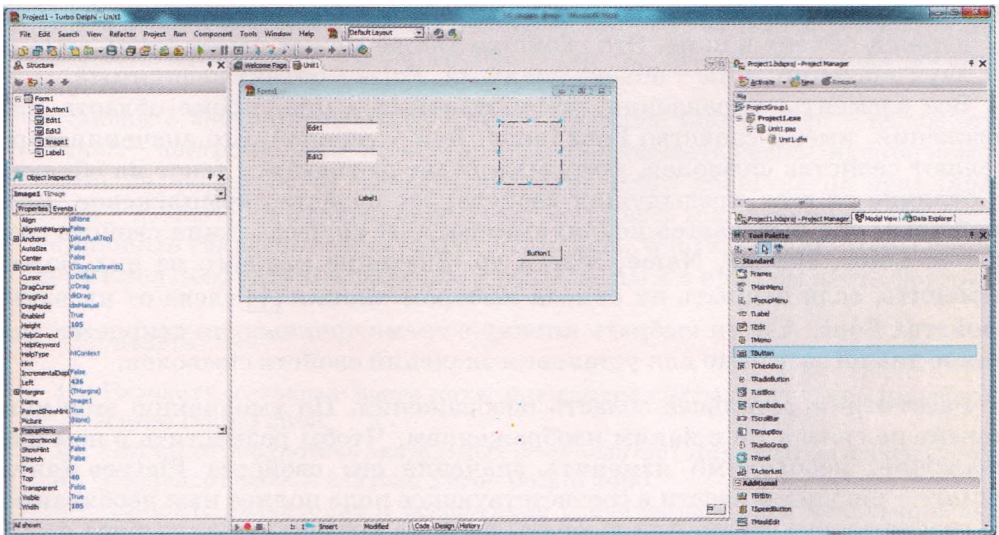


Рис. 2.17. Форма с размещенными на ней двумя полями, надписью, кнопкой и областью изображения

добавляются к структуре, в окне Object Inspector на вкладке Properties отображаются свойства этого объекта и их значения, а в поле, которое находится над вкладками этого окна, появляется его имя. В списке этого поля указываются имена всех элементов управления, которые используются в проекте. Используя этот список, можно сделать соответствующий элемент текущим.


Кнопка, поле, надпись и область изображения, как и форма, имеют свойства Name, Height, Width, Left, Top, а также целый ряд других свойств. Значения свойств этих объектов во время размещения на форме устанавливаются по умолчанию. Их можно изменить, введя новые значения на вкладке Properties окна Object Inspector. Значение некоторых из них можно изменить перетягиванием границ объекта или самого объекта по форме.

*Обращаем ваше внимание:*

1. Для кнопки, поля, надписи и для области изображения, в отличие от формы, значения свойств Left и Top определяют положение объекта не относительно левой и верхней границ экрана, а относительно левой и верхней границ формы, на которой они размещены.
2. Поле и надпись имеют свойство Color, а кнопка и область изображения этого свойства не имеют.
3. Кнопка и надпись имеют свойство Caption, а поле и область изображения этого свойства не имеют.
4. Вместо свойства Caption поле имеет свойство Text (англ. *text* - текст), значение которого определяет, какой текст находится в поле.

Рассмотрим еще два свойства, которые имеют все элементы управления, рассмотренные выше:

- Enabled (англ. *enabled* - доступный) - его значение определяет, доступен ли объект для операций над ним. Это свойство может иметь лишь одно из двух значений: *True* (англ. *true* - истина, правда) - объект доступный, *False* (англ. *false* - ошибка, неправда) - объект недоступный;
- Visible (англ. *visible* - видимый) - его значение определяет, видимый ли данный объект в окне. Это свойство также может иметь лишь одно из двух значений: *True* - объект видимый, *False* - объект невидимый.

Все элементы управления, рассмотренные выше, кроме области изображения, имеют свойство Font (англ. *font* - шрифт). Его значения определяют свойства символов, которыми будет выводиться текст на объекте. В отличие от всех предыдущих свойств, это свойство комплексное. Комплексным оно называется потому, что состоит из нескольких свойств символов: Color, Height, Name, Size и др. Значение каждого из них можно изменить, если открыть их список выбором кнопки  слева от названия свойства Font. А если выбрать кнопку с тремя точками, то откроется знакомое диалоговое окно для установки значений свойств символов.

Рассмотрим детальнее область изображения. По умолчанию этот компонент не связан ни с каким изображением. Чтобы разместить в нем изображение, необходимо изменить значение его свойства Picture (англ. *picture* - рисунок): ввести в соответствующее поле полное имя необходимого графического файла или выбрать файл на носителе. Для выбора файла на носителе нужно:

1. Выбрать кнопку в правом конце поля Picture.

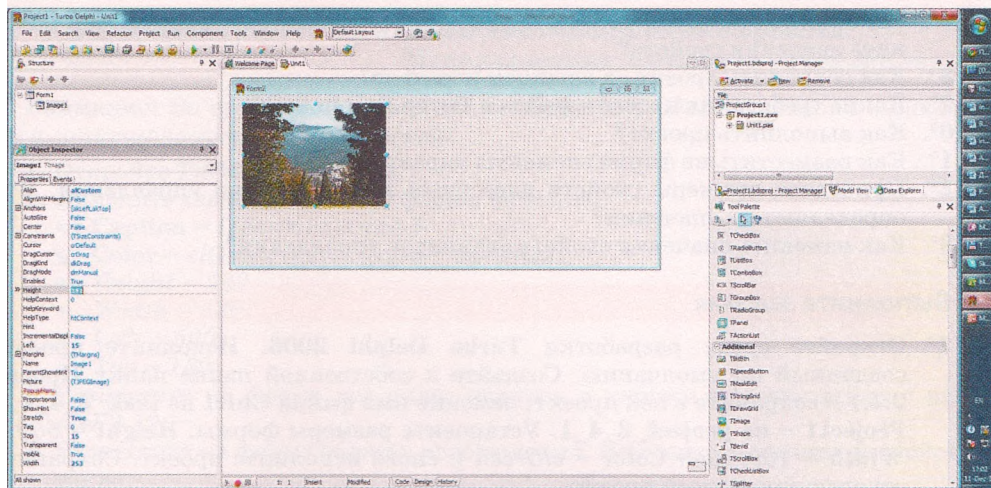


Рис. 2.18. Изображение, размещенное на форме, с измененными значениями некоторых его свойств

2. Выбрать кнопку Load (англ. *load* - загрузить) в окне Picture Editor (англ. *editor* - редактор).
3. Открыть в окне Load Picture необходимую папку, выбрать нужный графический файл и кнопку Open (англ. *open* — открыть).
4. Выбрать кнопку ОК.

Открытое изображение будет размещено в области изображения, причем, в зависимости от размеров компонента и изображения, может быть видимой лишь часть изображения. После этого целесообразно изменить значения таких свойств области изображения (рис. 2.18):

- установить значение свойства Stretch = *True* (англ. *stretch* - растянуть), чтобы изображение заняло всю область компонента;
- установить значение свойства Proportional = *True* (англ. *proportional* - пропорциональный), чтобы изображение не искажалось при растягивании;
- установить значение свойства Align = *alCustom* (англ. *align* - выровнять, *custom* - настроить), чтобы можно было настраивать расположение компонента на форме (если выбрать значение этого свойства *alClient* (англ. *client* - клиент), то изображение будет покрывать всю поверхность формы);
- установить такие значения свойств Top, Left, Width, Height, чтобы компонент имел необходимые размеры и располагался в нужном месте формы.

### Проверьте себя

- 1°. Назовите составные части интегрированной среды визуальной разработки Turbo Delphi 2006.
- 2°. Что такое библиотека визуальных компонентов? Для чего она используется?
- 3°. Назовите способы запуска Turbo Delphi 2006.
- 4°. Опишите назначение и содержание подчиненных окон начального окна Turbo Delphi 2006.
- 5°. Как создать новый проект в Turbo Delphi 2006?
- 6°. Что такое форма?

- 7°. Назовите и опишите, что определяют значения свойств формы. Как изменить значения свойств?
- 8°. Как сохранить проект на внешнем носителе?
- 9°. Какие требования к имени файла в Turbo Delphi 2006?
- 10°. Как выполнить проект?
- 11°. Как разместить на форме элементы управления?
- 12°. Приведите примеры свойств известных вам элементов управления. Что определяют их значения?
- 13°. Как изменить значения свойств элементов управления?



### Выполните задания

- 1°. Откройте среду разработки Turbo Delphi 2006. Исполните проект, созданный по умолчанию. Создайте в собственной папке папку Проект 2.4.1 и сохраните в ней проект, изменив имя файла Unit1 на task\_2\_4\_1, а Project1 — на project\_2\_4\_1. Установите размеры формы: Height = 500 и Width = 100, цвет Color = clGreen и снова исполните проект. Сохраните отредактированный проект.
- 2°. Откройте Turbo Delphi 2006. Увеличьте размеры формы. Измените надпись в строке заголовка формы на Я учусь программировать. Измените значения свойств формы Top и Left, чтобы во время исполнения проекта окно открылось в правой нижней части экрана. Создайте в собственной папке папку Проект 2.4.2 и сохраните в ней проект.
- 3°. Откройте Turbo Delphi 2006. Создайте проект, исполнение которого откроет окно синего цвета, размерами 300 на 300 пикселей, левая граница которого находится на расстоянии 300 пикселей от левой границы экрана, а верхняя - на расстоянии 400 пикселей от верхней границы экрана. Создайте в собственной папке папку Проект 2.4.3 и сохраните в ней проект.
- 4°. Откройте Turbo Delphi 2006. Разместите на форме область изображения. Задайте значения свойств этого компонента, чтобы его размеры были 150 на 200 пикселей, левая граница находилась на расстоянии 30 пикселей от левой границы формы, а верхняя - на расстоянии 20 пикселей от верхней границы формы. Свяжите с этим компонентом графическое изображение, например из файла рис 2.4.7. Установите такие значения свойств области изображения, чтобы оно занимало всю область. Выполните проект. Создайте в собственной папке папку Проект 2.4.4 и сохраните в ней проект.
- 5\*. Откройте Turbo Delphi 2006. Разместите на форме область изображения. Разместите в ней произвольное изображение. Исследуйте экспериментально, что определяют другие значения свойства Align. Запишите выводы в тетрадь.
- 6\*. Откройте Turbo Delphi 2006. В меню File есть команды Save, Save As, Save Project As. Выясните назначение каждой из них и их отличие от команды Save All. Запишите выводы в тетрадь.
- 7\*. Откройте Turbo Delphi 2006. Выясните отличие команд Close и Close All меню File. Запишите выводы в тетрадь.
- 8°. Откройте готовый проект, например из папки Проект 2.4.8. Выясните значения пяти свойств формы и четырех свойств кнопки. Запишите их в тетрадь.
- 9°. Откройте готовый проект, например из папки Проект 2.4.9. Сравните значение свойств поля и надписи. Заполните в тетради сравнительную таблицу.
- 10°. В Справке найдите информацию о 5-6 свойствах формы, кнопки, поля, надписи, которые не были рассмотрены в данном пункте. Сделайте соответствующие записи в тетрадь.





**Практическая работа № 5. Создание и выполнение проекта.  
Использование элементов управления**

**Внимание!** Во время работы с компьютером соблюдайте правила безопасности и санитарно-гигиенические нормы.

1. Откройте **Turbo Delphi 2006**.
2. Измените значения свойств формы на такие:
  - **Caption** = *Практическая 5*
  - **Color** = *clGreen* (англ. *green* - зеленый)
  - **Height** = *580*
  - **Width** = *580*
  - **Left** = *500*
  - **Top** = *150*
3. Разместите на форме два поля, надпись и кнопку в соответствии с рисунком 2.19.
4. Разместите на форме область изображения. Измените значения ее свойств на такие:
  - **Stretch** = *True*
  - **Align** = *alCustom*
  - **Left** = *10*
  - **Top** = *30*
  - **Height** = *150*.
  - **Width** = *130*

Расположите в этой области изображение, например из файла **практическая 5.4.jpg**. Установите значение свойства области, чтобы изображение занимало всю область.

5. Разместите на форме еще одну область изображения. Измените значения ее свойств на такие, чтобы она имела размеры 60 на 60 пикселей и располагалась в правом верхнем углу окна. Расположите в области изображение, например из файла **практическая 5.5.jpg**.
6. Создайте в собственной папке папку **Практическая 5** и сохраните в ней проект.
7. Исполните проект.

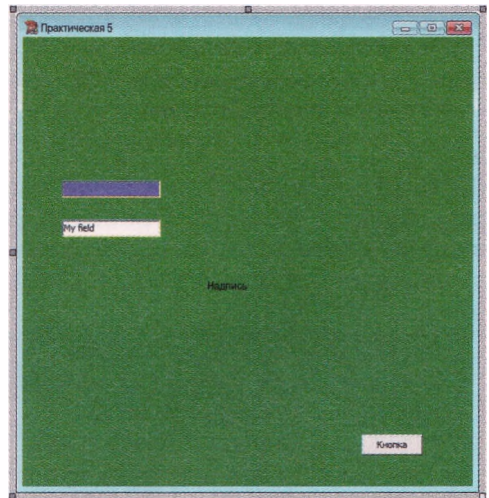


Рис. 2.19

## 2.5. События и их обработки



1. Назовите окна Turbo Delphi 2006 и поясните их назначение.
2. Что такое форма? Назовите несколько ее свойств. Как изменить их значения?
3. Какие общие свойства имеют кнопка, надпись, поле и область изображения? Какие свойства имеют одни из них и не имеют другие?
4. Сформулируйте правила записи выражений в формулах Excel.
5. Что такое алгоритм? Какой алгоритм называется линейным? Какой вид имеет блок-схема линейного алгоритма?

## События и их обработчики

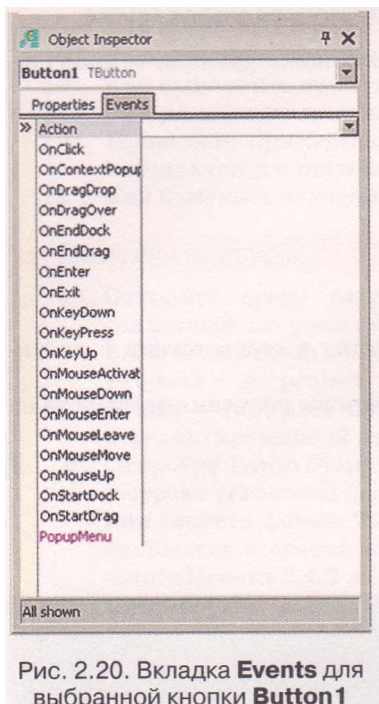


Рис. 2.20. Вкладка **Events** для выбранной кнопки **Button1**

Работая с разнообразными программами, вы привыкли к тому, что выбор кнопки всегда приводит к наступлению определенного события: открытию или закрытию определенного окна, изменению значений свойств определенных объектов, перемещению определенного объекта по экрану и т. п. Но если выбрать кнопку в любом из проектов, которые вы создавали во время изучения предыдущего пункта, то ничего не произойдет.

Причиной этому является то, что самого лишь размещения на форме кнопки недостаточно, чтобы при ее выборе что-то произошло. Программе нужно «сообщить», какое новое событие должно состояться при выборе кнопки.

Если выделить кнопку, то в окне Object Inspector на вкладке Events (англ. *events* - события) можно указать, какое событие должно произойти как реакция на наступление другого события, например выбор кнопки (рис. 2.20). Для этого нужно дважды щелкнуть в поле справа от надписи OnClick (англ. *on click* - на щелчок кнопкой мыши). В результате этих действий в поле OnClick на вкладке Events появляется текст Button1Click (Button1 - это имя выделенной кнопки), а в центральной части окна среды разработки открывается вкладка Code, на которой создается заготовка процедуры TForm1.Button1Click, команды которой и будут выполняться после выбора кнопки Button1 (рис. 2.21).

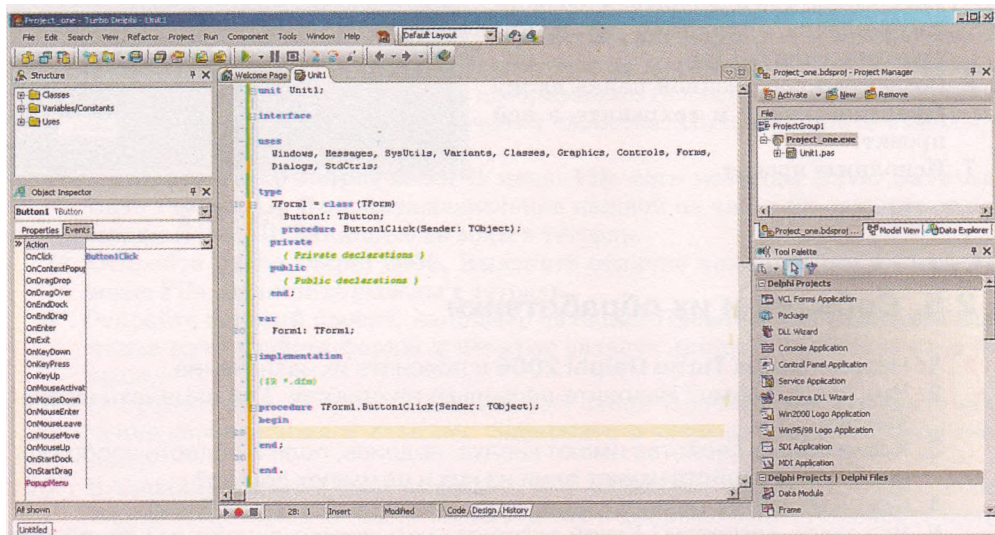


Рис. 2.21. Вкладка **Code** с заготовкой процедуры **TForm1.Button1Click**



*Процедура* (лат. *procedere* — продвигаться, идти вперед) — это часть программы, которая имеет имя и которую можно по этому имени вызвать на исполнение в разных частях программы.

Итак, после выбора кнопки `Button1` (событие `Click`) наступает событие `OnClick`, которое заключается в исполнении команд процедуры `TForm1.Button1Click`.

Процедуры в Delphi являются одним из видов подпрограмм. Другим видом подпрограмм в Delphi являются функции. Их рассмотрим детальнее в следующих пунктах.



Процедура, которая выполняется при наступлении некоторого события, называется *обработчиком этого события*.

Процедура, связанная с определенным объектом, называется *методом этого объекта*.

Так, процедура `TForm1.Button1Click` является обработчиком события `OnClick`, которое наступает после выбора кнопки `Button1`, а также является методом объекта «Кнопка `Button1`».

Аналогично создаются обработчики других событий, например:

- `OnMouseMove` (англ. *on mouse move* - на перемещение мыши) - это событие наступает после наведения указателя на объект;
  - `OnKeyPress` (англ. *on key press* - на нажатие клавиши) - это событие наступает после нажатия клавиши клавиатуры;
  - `OnStartDrag` (англ. *on start drag* - на начало перетягивания) - это событие наступает после начала перетягивания объекта
- и др.

Аналогично можно создавать методы и других объектов, например формы. Список событий на вкладке `Events` для формы содержит события, которые опеределены для кнопки, а также другие. Например:

- `OnCreate` (англ. *on create* - на создание) - это событие наступает после начала создания формы; команды обработчика этого события выполняются во время создания формы, перед ее открытием;
- `OnDblClick` (англ. *on double click* - на двойной щелчок) - это событие наступает после двойного щелчка на форме.

## Создание процедур-обработчиков событий в Delphi

Рассмотрим детальнее структуру процедуры `TForm1.Button1Click` - обработчика события `OnClick`:

```
procedure TForm1.Button1Click (Sender: TObject);
begin
end;
```

Первая строка процедуры - строка заголовка. Она состоит из стандартного слова `procedure` (англ. *procedure* - процедура), имени процедуры `TForm1.Button1Click` и круглых скобок, в которых указывается, что данная процедура будет выполняться в результате наступления определенного события с определенным объектом, в данном случае события `Click` с кнопкой `Button1`. Заканчивается строка заголовка точкой с запятой (;).

Команды процедуры, которые будут выполняться при наступлении события OnClick, записываются между двумя стандартными словами begin (англ. *begin* - начало) и end (англ. *end* - конец). Команды процедуры образуют тело процедуры. Слова begin и end определяют, где начинается и где заканчивается тело процедуры. Можно сказать, что они играют роль открывающей и закрывающей скобки, между которыми записывается тело процедуры, поэтому их называют операторными скобками. После слова end должна стоять точка с запятой (;). Каждая команда процедуры также должна заканчиваться точкой с запятой (;).

В предыдущем пункте мы устанавливали значения свойств элементов управления перед запуском проекта на исполнение. Теперь же можно изменять значения этих свойств во время исполнения проекта. Для этого нужно добавить к тексту программы обработчик определенного события.

Если нужно, чтобы в результате исполнения процедуры изменились значения свойств одного из объектов, процедура должна содержать команды установки значения свойств объекта. Общий вид этих команд таков:

`<имя объекта>.<имя свойства> := <значение или выражение>;`

Например,

`Form1.Color := clGreen;` - установить цвет фона окна - *зеленый*;

`Form1.Left := 300;` - установить отступ левой границы окна - *300 пикселей*;

`Form1.Width := Form1.Width*2;` - увеличить текущую ширину окна вдвое;

`Button1.Caption := 'Увеличить';` — установить текст заголовка на кнопке - *Увеличить* (если значением свойства является текст, то его нужно заключать в одинарные кавычки);

`Label1.Font.Color := clRed;` - установить цвет символов для отображения текста в надписи — *красный*.

*Обращаем ваше внимание, что эти команды являются разновидностями команд присваивания, которые мы уже использовали при составлении алгоритмов.*

Приведем пример процедуры TForm1.Button1Click, исполнение которой изменит цвет окна на синий, изменит текст в строке заголовка, уменьшит его ширину на 150 пикселей, изменит положение кнопки и текст на ней:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  Form1.Color := clBlue;
  Form1.Caption := 'Значения свойств изменились';
  Form1.Width := Form1.Width - 150;
  Button1.Left := 60;
  Button1.Top := 100;
  Button1.Caption := 'Изменить!';
end;
```

На рисунке 2.22 слева приведен вид окна после запуска проекта, а справа - после выбора кнопки Button1.

Среда разработки Turbo Delphi 2006 имеет удобные средства помощи во время ввода текста проекта:

- после ввода первой буквы имени объекта можно нажать комбинацию клавиш Ctrl+Пробел и откроется список имен объектов, свойств, про-



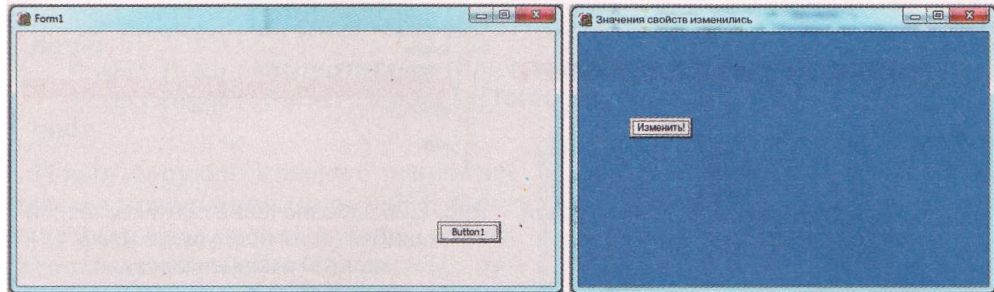


Рис. 2.22. Окно исполнения проекта до выбора кнопки **Button1** и после ее выбора

цедур, которые начинаются с этой буквы (рис. 2.23); из этого списка можно выбрать нужное и этот текст будет вставлен в текст проекта;

- при вводе текста проекта среда разработки следит за синтаксическими правилами написания команд; если эти правила нарушаются, то текст с ошибками или текст сразу после них подчеркивается красной волнистой линией (рис. 2.23, подчеркивается буква L; рис. 2.24, слово Label1 написано правильно, но после точки не указано свойство этого объекта, поэтому подчеркивается следующее слово); нужно внимательно следить за этим и своевременно исправлять все синтаксические ошибки;
- Turbo Delphi 2006 не различает большие и малые буквы; но если слово образовано из нескольких слов, принято (и удобно) писать первую букву каждого слова большой, например TForm1.Button1Click;
- после ввода точки, которая разделяет имя объекта и имя свойства, автоматически открывается список свойств данного объекта (рис. 2.24); дальше имя свойства можно не вводить с клавиатуры, а найти в списке и нажать клавишу Enter; такой способ ввода помогает не допускать синтаксических ошибок, особенно тем, кто не знает английского языка;
- после запуска проекта на исполнение компилятор анализирует текст на наличие синтаксических ошибок; если такие ошибки будут найдены, то после анализа всех ошибок компиляция проекта прерывается и красным цветом выделяется фон первой из тех строк текста проекта, которая содержит синтаксические ошибки, или следующей за ней строки (на рис. 2.25 неправильно написано слово Label1, поэтому красным

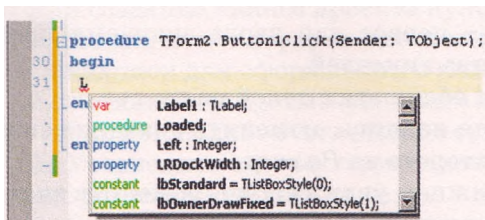


Рис. 2.23. Список, который открывается после нажатия **Ctrl+Пробел**

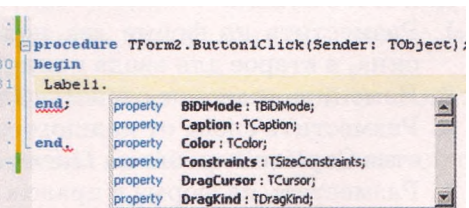


Рис. 2.24. Список свойств объекта **Label1**, который открывается после ввода точки

```

procedure TForm2.Button1Click(Sender: TObject);
30 begin
31 Label1.Color := clRed;
   Edit1.Color := clRed
end;
end.

```

Рис. 2.25. Обозначение синтаксической ошибки при вводе слова **Label1**

```

procedure TForm2.Button1Click(Sender: TObject);
30 begin
   Label1.Color := clRed
32 Edit1.Color := clRed
end.
end.

```

Рис. 2.26. Обозначение синтаксической ошибки, если пропущена точка с запятой в конце команды

фоном обозначена именно эта строка, а на рисунке 2.26 пропущена точка с запятой в конце первой команды, в таких случаях красным фоном обозначается следующая строка); все другие ошибки подчеркиваются красной волнистой линией; кроме того, в окне Structure появляются сообщения о видах ошибок; нужно исправить в тексте программы все синтаксические ошибки и опять запустить проект на исполнение.

Мы рассмотрели, как создать и использовать процедуру TForm1.Button1Click - обработчик события OnClick. Аналогично можно создать другие процедуры - обработчики других событий.

## Ввод и вывод данных во время исполнения проекта

Мы рассмотрели, как изменить значения свойств элементов управления во время исполнения проекта, используя процедуру TForm1.Button1Click - обработчик события OnClick. Но в рассмотренном примере при каждом выборе кнопки будут выполняться одни и те же команды процедуры, а следовательно, каждый раз одинаково будут изменяться значения указанных в командах процедуры свойств объектов. Если необходимо во время исполнения проекта каждый раз по-разному изменять значения свойств указанных объектов, то нужно значения этих свойств каким-то образом вводить в программу во время ее исполнения.

Для ввода данных во время исполнения проекта можно использовать поля. Вы уже знаете, что текст, который находится в поле, является значением его свойства Text. Воспользуемся этим.

Создадим такую процедуру TForm1.Button1Click - обработчик события OnClick, исполнение которой дает возможность установить произвольный цвет окна и увеличить его высоту на произвольное целое количество пикселей.

Для этого нужно:

1. Разместить на форме два поля - первое для ввода значения цвета окна, а второе для ввода количества пикселей.
2. Изменить значение свойства Text обоих этих полей на пустое.
3. Разместить слева от каждого поля надпись, изменить значение свойства Caption первого на *Цвет*, а второго на *Размер*.
4. Разместить на форме в правом нижнем углу кнопку, изменить значение ее свойства Caption на *Изменить*.
5. Создать заготовку процедуры TForm1.Button1Click, которая будет выполняться после выбора кнопки.
6. Ввести текст процедуры TForm1.Button1Click:

```

procedure TForm1.Button1Click (Sender: TObject);
begin
    Form1.Color := StringToColor (Edit1.Text);
    Form1.Height := Form1.Height + StrToInt (Edit2.Text);
end;
    
```

После запуска проекта откроется окно его исполнения (рис. 2.27). В соответствующие поля нужно ввести нужные значения цвета формы, например *clRed*, и количества пикселей для увеличения высоты формы, например *300*, после чего выбрать кнопку Изменить.

После выбора кнопки наступит событие *OnClick*, а следовательно, запустится на исполнение процедура *TForm1.Button1Click* - обработчик события *OnClick*.

При выполнении первой команды процедуры текст, который будет введен в первое поле и станет значением свойства *Text* этого поля, будет преобразован в обозначение цвета (с использованием функции *StringToColor* (амер. *string to color* - строка в цвет)) и присвоен свойству *Color* формы, в результате чего цвет формы станет заданным.

Несмотря на то что во второе поле формы введено число, программа будет воспринимать его как текст, то есть не сможет выполнять с ним математические операции. Поэтому нужно сначала преобразовать этот текст в целое число. Для этого используется функция *StrToInt* (англ. *string to integer* - строка в целое). После чего это число прибавляется к текущему значению высоты окна.

Таким образом, при каждом запуске проекта на исполнение или перед каждым выбором кнопки можно вводить в поля разные значения, достигая тем самым установки нужного цвета окна и увеличения его высоты.

Надписи можно использовать не только для оформления подписей объектов, но и для вывода результатов исполнения проекта.

Создадим простой калькулятор, который будет складывать два произвольных действительных числа. Во время исполнения проекта пользователь будет вводить по одному произвольному действительному числу в каждое поле и выбирать кнопку с заголовком *Сложить*. При выборе кнопки наступит событие *OnClick*, и в результате выполнится процедура *TForm1.Button1Click*, которая будет складывать эти числа и выводить сумму в надпись.

Для создания такого проекта нужно (рис. 2.28):

1. Разместить на форме два поля, надпись для вывода результата, три надписи для оформления и кнопку.
2. Установить пустые значения свойства *Text* двух полей и свойства *Caption* надписи для вывода суммы.
3. Установить соответствующие значения свойства *Caption* для других трех надписей и кнопки.
4. Создать заготовку процедуры *TForm1.Button1Click*, которая будет исполняться после выбора кнопки.
5. Ввести текст процедуры *TForm1.Button1Click* (рис. 2.29).

Рассмотрим назначение каждой команды приведенной процедуры.

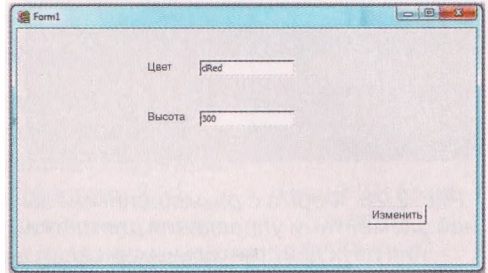


Рис. 2.27. Окно исполнения проекта для установки цвета окна и увеличения его высоты



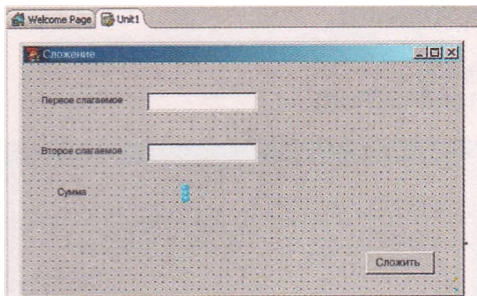


Рис. 2.28. Форма с размещенными на ней элементами управления для сложения двух действительных чисел

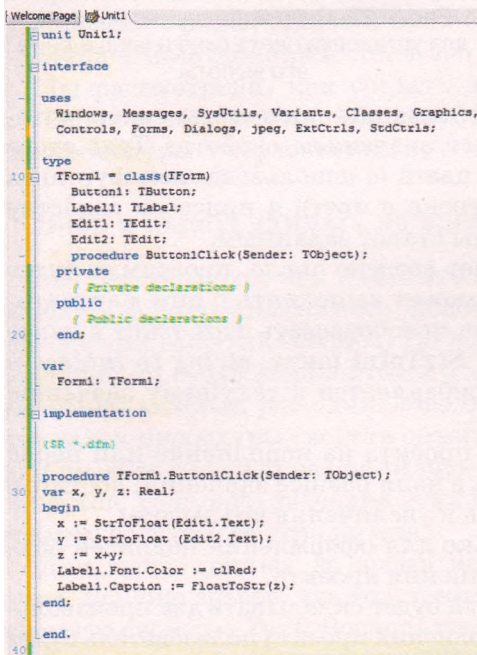


Рис. 2.29. Текст проекта на вкладке **Code**

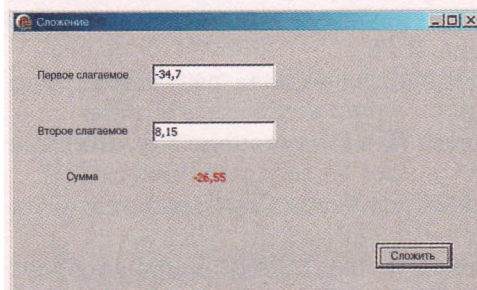


Рис. 2.30. Исполнение проекта сложения двух действительных чисел

Как мы уже отмечали выше, число, которое вводится в поле, становится значением его свойства `Text` и воспринимается программой как текст. Поэтому первые две команды процедуры являются командами присваивания, которые предназначены для преобразования текстового представления действительных чисел в сами действительные числа (для этого использована функция `StrToFloat` (англ. *string to float* - строка в число с плавающей десятичной точкой, то есть в действительное число) и присваивания этих чисел переменным `x` и `y` соответственно.

Третья команда складывает эти числа и присваивает результат переменной `z`.

Четвертая команда устанавливает красным цвет символов, которым будет выводиться результат в надпись.

Пятая команда преобразовывает результат сложения (значение переменной `z`) из числа в текст, используя функцию `FloatToStr` (англ. *float to string* - число с плавающей десятичной точкой в строку) и присваивает этот текст значению свойства `Caption` надписи с именем `Label1` для вывода результата.

Во время исполнения созданного проекта пользователь вводит в поля числа (например `-34,7` и `8,15`), выбирает кнопку *Сложить* и получает результат `-26,55` (рис. 2.30).

Не закрывая проект, пользователь может ввести в поля другие числа, опять выбрать кнопку *Сложить* и получить новый результат. И так можно повторять произвольное количество раз. Это иллюстрирует такое свойство алгоритма как массовость.

Иногда удобно выводить сообщение не в надписи на форме, а в специальное отдельное окно. Это можно осуществить, включив в текст программы команду `ShowMessage(<Текст>)` (англ. *show message* - показать сообщение). По этой команде открывается специальное окно вывода сообщения, в котором выводится текст, указанный



в команде. Если такую команду включить в рассмотренную выше процедуру вывода суммы двух введенных действительных чисел вместо последней команды вывода результата в надпись или после этой команды, то она будет выглядеть так:

```
ShowMessage ('Сумма '+FloatToStr(z));
```

Знак *плюс* в скобках обозначает соединение двух указанных текстов. Пробел после слова Сумма используют для того, чтобы в окне отделить число от текста. Результат выполнения этой команды представлен на рисунке 2.31.

При использовании окна сообщения для завершения исполнения проекта сначала следует закрыть окно сообщения, а затем - окно проекта.

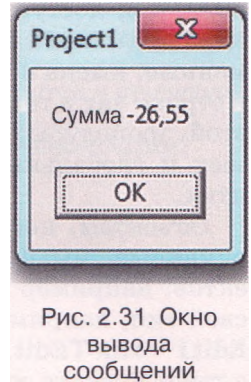


Рис. 2.31. Окно вывода сообщений

## Переменные и их свойства. Правила записи математических выражений в Delphi

В последней рассмотренной процедуре были использованы три переменные:  $x$ ,  $y$ ,  $z$ . В языке Delphi каждая переменная имеет имя, значение и тип.



*Тип переменной* определяет, какие значения может принимать эта переменная, какие операции над ней можно выполнять и участок памяти какого размера программа выделит для хранения значения этой переменной.

Переменные  $x$ ,  $y$ ,  $z$ , использованные в последней процедуре, по условию задачи могут принимать любые действительные значения. Таким значениям соответствует тип Real (англ. *real* - действительный). Переменные этого типа могут принимать целые и дробные значения от  $-5 \cdot 10^{-324}$  до  $1,7 \cdot 10^{308}$ , для каждого значения переменных этого типа выделяется 8 байт памяти.

Типы переменных, используемых в процедуре, указываются после ключевого слова var (англ. *variable* - переменная) в строке, которая находится между строкой заголовка процедуры и строкой со словом begin (см. рис. 2.29). Эта строка называется строкой объявления переменных. Общий вид строки объявления переменных такой:

```
var <список имен переменных одного типа, разделенных запятыми>:  
<имя типа>; <список имен переменных другого типа, разделенных  
запятыми>; <имя типа>; и т. д.
```

Если переменные будут принимать только целые значения, то для них целесообразно использовать тип Integer (англ. *integer* - целый). Переменные этого типа могут принимать целые значения от  $-2\,147\,483\,648$  ( $-2^{32}$ ) до  $2\,147\,483\,647$  ( $2^{32} - 1$ ), для каждого значения выделяется 4 байт памяти.

Для преобразования текстового представления целого числа в само целое число используется функция StrToInt (англ. *string to integer* - строка в целое число), а для обратного преобразования - функция IntToStr.

Выше текста составленной процедуры находится еще одна строка var. На рисунке 2.29 в этой строке указано имя одной переменной Form1 типа TForm. Эта переменная соответствует форме, которая используется в проекте. В этой строке указываются имена и типы тех переменных, которые

являются доступными в любой части программы. Такие переменные называются глобальными. В отличие от глобальных переменных, те переменные, имена и типы которых объявлены в строке заголовка процедуры и в строке `var` в тексте определенной процедуры, доступны только командам этой процедуры. Такие переменные называются локальными. О локальных и глобальных переменных поговорим подробнее в следующих пунктах.

Объектам, которые мы размещаем на форме, также соответствуют переменные, имена которых совпадают с именами соответствующих объектов, например `Button1`, `Edit1`. Каждая из этих переменных тоже имеет свой тип, например переменная `Button1` имеет тип `TButton`, а переменная `Edit1` - тип `TEdit`. Имена и типы этих переменных среда разработки вносит в текст проекта автоматически в раздел `type` (англ. *type* - тип) (рис. 2.29).

Переменная может получить значение в результате выполнения команды присваивания. Общий вид команды присваивания в Delphi таков: `<имя переменной> := <выражение>`. Выражение может состоять из значения (константы: число, текст в одинарных кавычках и др.), из имени переменной или содержать несколько значений и имен переменных, соединенных знаками арифметических операций, функции, скобки.

Обозначение арифметических действий и правила записи выражений в Delphi похожи на аналогичные правила, которые рассматривались при изучении табличного процессора Excel. Но есть и определенные особенности:

- 1) при вводе в текст проекта десятичных дробей целую и дробную части нужно разделять точкой;
- 2) при вводе в ходе исполнения проекта в поле десятичной дроби делитель целой и дробной частей зависит от настроек Windows (в русской версии Windows XP по умолчанию установлена запятая);
- 3) ни в тексте проекта, ни при вводе в поле нельзя использовать обыкновенные дроби;
- 4) в Delphi нет операции возведения в степень; для возведения в степень используется специальная функция, которую мы рассмотрим позже, а пока степень с натуральным показателем можно заменять произведением;
- 5) приоритет операций во всех случаях определяется правилами математики.





### Проверьте себя


1. Что такое обработчик события? Как связать с определенным событием его обработчик?
2. Что такое процедура? Опишите структуру процедуры - обработчика события.
3. Приведите общий вид команды, которой можно изменить значение свойства элемента управления. Приведите примеры таких команд.
4. Как обозначаются синтаксические ошибки в среде разработки? Приведите примеры.
5. Какие элементы управления можно использовать для ввода и вывода данных?
6. Как вывести сообщение в отдельное специальное окно?
7. Для чего предназначены функции `StrToFloat`, `FloatToStr`, `StrToInt`, `IntToStr`?
8. Что определяет тип переменной? Какие типы переменных вы знаете?

- 9°. Какие значения могут принимать переменные типа **Real** и типа **Integer**? Какой размер участка памяти выделяется для хранения значений переменных каждого из этих типов?
- 10°. Как описываются типы переменных, которые используются в процедуре; во всей программе?
- 11°. Какие переменные называются локальными, а какие - глобальными?
- 12°. Какой общий вид команды присваивания в **Delphi**?

 **Выполните задания**

1. Запишите команды присваивания переменной  $x$  значений приведенных выражений:

a) $3x^2 - 5y^3 + 7z;$	 г) $\frac{34 - 5x^3 + yz}{-24 - 4,76y^2};$
б) $(x + y + z) : 3;$	д) $\frac{2x^2 - 5}{3 - 4y^2} - 7zy;$
в) $\frac{7x - 12y^3}{45 - 2z^2};$	 е) $7,23xyz - \frac{7x^2 - 15y}{4z - 8,23y^3} - x.$


2. Откройте новый проект. Измените заголовок формы на *Я изменяю значения свойств формы*. Разместите на форме кнопку. Измените заголовок кнопки на *Изменить*. Измените имя формы на **MyForm1**. Создайте для события **OnClick** кнопки заготовку обработчика этого события. Введите в процедуру команды, которые изменят цвет окна на **clPurple**, уменьшат ширину окна на 100 пикселей и увеличат его высоту на 200 пикселей. Выполните проект. Выберите кнопку. Убедитесь, что изменились размеры и цвет окна. Создайте в собственной папке папку **Проект 2.5.2** и сохраните в ней проект.
3. Создайте проект для умножения двух действительных чисел. Установите желтый цвет фона и синий цвет текста для вывода результата. Предусмотрите вывод результата еще и в окно для сообщений. Выполните проект несколько раз для разных чисел, следите за правильностью результата. Создайте в собственной папке папку **Проект 2.5.3** и сохраните в ней проект.
-  4. Откройте новый проект. Установите заголовок формы *Доступный/Недоступный*. Разместите на форме две кнопки. Для первой установите такие значения свойств: **Caption** — *Кнопка 1*, **Enabled** — *True*. Для второй соответственно: **Caption** — *Кнопка 2*, **Enabled** — *False*. Создайте для события **OnClick** первой кнопки заготовку обработчика этого события и введите в него такие команды:

```
Button2.Enabled := True;
Button1.Enabled := False;
```

Создайте для события **OnClick** второй кнопки заготовку обработчика этого события и введите в него такие команды:

```
Button1.Enabled := True;
Button2.Enabled := False;
```

Проанализируйте тексты этих процедур и сделайте вывод, что будет происходить во время исполнения проекта. Выполните проект. По очереди выбирайте доступную кнопку и наблюдайте за результатами. Создайте в собственной папке папку **Проект 2.5.4** и сохраните в ней проект.

-  5. Создайте проект, в котором событие **OnMouseMove** кнопки вызовет изменение ее размеров, заголовка и положения в окне. Создайте в собственной папке папку **Проект 2.5.5** и сохраните в ней проект.

6\*. Создайте проект, в котором событие OnClick поля вызовет изменение его размеров, цвета его фона и появление в поле текста с вашей фамилией и именем символами красного цвета. Создайте в собственной папке папку **Проект 2.5.6** и сохраните в ней проект.

7\*. Создайте проект, который по длине, ширине и высоте вашей комнаты будет вычислять площадь пола и площадь поверхности стен. Создайте в собственной папке папку **Проект 2.5.7** и сохраните в ней проект.

8\*. Создайте проект для нахождения суммы, разности и произведения двух произвольных действительных чисел. Создайте в собственной папке папку **Проект 2.5.8** и сохраните в ней проект.

9\*. Создайте проект, в котором расположите на форме три кнопки. Установите такие значения их свойств, чтобы в начале выполнения проекта первая кнопка была видима и доступна, вторая - видима и недоступна, третья — невидима и недоступна. Для каждой кнопки создайте процедуру для события OnClick, чтобы:

- 1) выбор первой кнопки делал ее невидимой и недоступной, вторую кнопку - видимой и доступной, третью кнопку - видимой и недоступной;
- 2) выбор второй кнопки делал ее невидимой и недоступной, третью кнопку - видимой и доступной, первую кнопку - видимой и недоступной;
- 3) выбор третьей кнопки делал ее невидимой и недоступной, первую кнопку - видимой и доступной, вторую кнопку - видимой и недоступной.

Создайте в собственной папке папку **Проект 2.5.9** и сохраните в ней проект.



### Практическая работа № 6. Ввод и вывод данных. Работа с переменными

**Внимание!** Во время работы с компьютером соблюдайте правила безопасности и санитарно-гигиенические нормы.

1. Откройте среду Turbo Delphi 2006 и создайте новый проект.
2. Установите для формы цвет фона желтый, размеры 1000 на 600 пикселей, текст заголовка *Практическая 6*.
3. Разместите на форме две кнопки, три поля и две надписи.
4. Задайте на первой кнопке текст *Перемещение*, на второй - *Вычисление*, у полей и надписей - *пустой текст*.
5. Разместите на форме возле каждого поля и каждой надписи дополнительные надписи с текстами, которые будут пояснять их назначение.
6. Составьте обработчик события OnClick первой кнопки, выполнение которого приведет к перемещению формы на 50 пикселей вправо и на 30 пикселей вниз, каждого поля - на 20 пикселей влево и на 30 пикселей вверх и сделает их фон красным.
7. Выберите первую кнопку и убедитесь в правильности результатов.
8. Составьте обработчик события OnClick второй кнопки, выполнение которого приведет к нахождению объема и площади полной поверхности прямоугольного параллелепипеда, считая, что в поля будут введены длины его ребер.
9. Выберите вторую кнопку и убедитесь в правильности результатов.
10. Добавьте ко второму обработчику события команду вывода сообщения об объеме параллелепипеда в окно сообщений. Повторно выберите вторую кнопку и убедитесь в правильности результатов.
11. Создайте в собственной папке папку *Практическая 6* и сохраните в ней проект.



## 2.6. Отладка программ



1. Для чего предназначены компиляторы? В чем особенности их работы?
2. Какие ошибки в программном коде обнаруживаются на этапе компиляции? В чем они заключаются? Как они отображаются в Turbo Delphi 2006? Где отображается их описание?
3. Какие ошибки называются логическими? Как можно обнаружить имеющиеся логические ошибки?
4. Как открыть ранее созданный проект?

### Тестирование программы. Комментарии

Вы уже знаете, что в процессе разработки проекта могут возникать синтаксические и логические ошибки. Синтаксические ошибки, которые не заметил автор проекта, находит компилятор на этапе компиляции во время запуска проекта на исполнение. Нужно исправить все эти ошибки и опять запустить на исполнение новую версию проекта.

Логические ошибки в проекте обнаружить значительно сложнее, чем синтаксические. Ведь проект исполняется, выдает результаты, но они не соответствуют условию задачи, для решения которой создавался данный проект.

При рассмотрении этапов решения задач с использованием компьютера уже пояснялось, что одним из способов выявления логической ошибки является создание тестовых наборов входных данных с заранее известными результатами и исполнение созданного проекта для этих наборов данных. После чего нужно сравнить полученные результаты с ожидаемыми и найти команды в программном коде, выполнение которых является причиной появления логических ошибок. Процесс поиска логических ошибок в тексте программы с использованием тестовых наборов входных данных называют тестированием программы.

Объясним этот способ на таком примере. Пусть нужно составить проект

$$y = \frac{2x - 5}{x^2 + 1}$$

для вычисления значений функции  $y = \frac{2x - 5}{x^2 + 1}$ . Часто ученики, записывая команду присваивания для вычисления значения соответствующего выражения  $y := (2*x - 5)/(x*x + 1)$ , забывают взять знаменатель (а то и числитель) в скобки, то есть вводят, например, команду  $y := (2*x - 5)/x*x + 1$  или  $y := 2*x - b/x*x + 1$ . Такую логическую ошибку легко заметить, применив удачную подборку тестов. Так, во время тестирования для  $x = 0$  станет очевидным, что проект не выдает ожидаемого правильного результата  $-5$ , а прерывает свое исполнение, открывая окно с сообщением *division by zero* (англ. *division by zero* - деление на ноль). После этого нужно найти команду, которая содержит действие деления и может стать причиной такой ошибки.

*Обращаем ваше внимание, что для несложных программ, которые вы пока еще учитесь составлять, подобрать систему тестовых примеров не очень сложно. Это дело усложняется, если нужно протестировать большую по количеству команд и более сложную по своей логической структуре программу, а тем более, профессиональные проекты, которые содержат сотни,*

*а то и тысячи разнообразных команд.*

Помогают в поиске логических ошибок комментарии, которые можно включать в текст программы.



*Комментарий* — это текст, который игнорируется компилятором и включается в текст программы с целью облегчения понимания структуры программного кода.

Чтобы компилятор отличил команды от комментариев, в Turbo Delphi 2006 комментарии нужно брать в фигурные скобки, например {Это команда нахождения частного}. Если текст комментария короткий и занимает не больше одной строки, то можно вместо скобок поставить перед ним две косые черточки, например // Это команда нахождения частного. Иногда комментарии вместо фигурных скобок заключают в круглые скобки со звездочками, например (\*Это команда нахождения частного\*). Никакой комментарий не может размещаться внутри команды программы.

```
procedure TForm1.Button1Click(Sender: TObject);
30 var x, y, z: Real;
begin
    // Команды ввода
    x := StrToFloat(Edit1.Text);
    y := StrToFloat(Edit2.Text);
    z := x + y; // Команда вычисления результата
    // Команды для вывода
    Label1.Font.Color := clRed;
    Label1.Caption := FloatToStr(z);
    (Дальше идет команда вывода результата
40 в специальное окно сообщений)
    ShowMessage('Сумма ' + FloatToStr(z));
end;
43 end.
```

Рис. 2.32. Текст программы с комментариями

На рисунке 2.32 представлены примеры использования комментариев в тексте программы.

Если текст программы большой, то для поиска места логической ошибки иногда удобно тестировать программу не полностью, а частями. Если правильность работы некоторых частей программы не вызывает сомнений и выполнение этих частей не является необходимым для выполнения других, то эти части программы комментируют, то есть заключают в фигурные скобки.

Компилятор пропускает при компиляции закомментированные фрагменты программы и компилирует только все остальные фрагменты. Это дает возможность пользователю сосредоточить свое внимание на поиске логических ошибок в тех частях, где возможны логические ошибки, а также экономить время на тестировании программы.

## Пошаговое исполнение программ

Еще одним способом выявления в программе логических ошибок является ее пошаговое исполнение, или трассировка. Шаг исполнения программы — это строка программы. В строке могут быть записаны одна или несколько команд программы. Подпрограмму, как отдельную часть программы, можно выполнить как один шаг, а можно выполнить по строкам — это зависит от режима пошагового исполнения.

Для выполнения одного шага программы, если необходимо и подпрограмму выполнить по строкам, нужно нажать клавишу F7 или выполнить *Run => Trace Into* (англ. *trace into* - трассировать внутрь). А для выполнения одного шага программы, если можно подпрограмму выполнить полностью за один шаг, нужно нажать клавишу F8 или выполнить *Run => Step Over* (англ. *step over* - шаг поверху). Еще раз отметим, что подпрограмму можно выполнить как один шаг только в том случае, если мы уверены, что она не содержит логические ошибки.

Рассмотрим, как происходит пошаговое исполнение программы на примере программы нахождения суммы двух произвольных действительных чисел, рассмотренной в предыдущем подпункте (см. рис. 2.32).

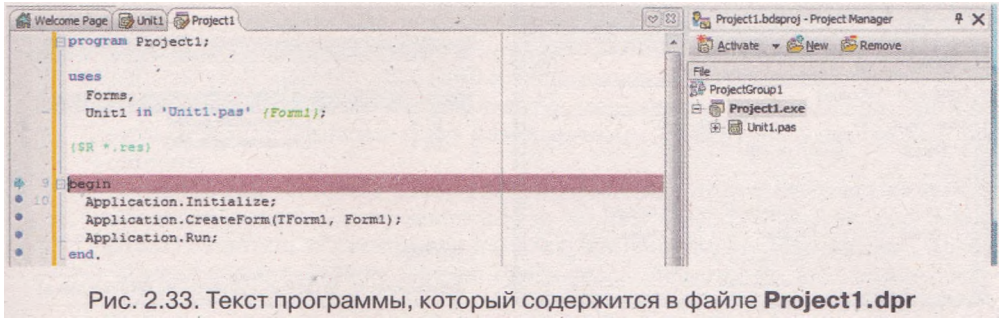


Рис. 2.33. Текст программы, который содержится в файле **Project1.dpr**

После первого нажатия клавиши F7 происходит компиляция проекта, и если проект не содержит синтаксических ошибок, то начинается его исполнение. Сначала открывается вкладка с текстом программы, который вошел в файл с расширением имени dpr (англ. *delphi project* - проект Делфи). Для рассматриваемого проекта это файл Project1.dpr. На рисунке 2.33 приведены команды этой части программы и ее начало подсвечивается цветной полоской. При каждом следующем нажатии клавиши F7 будут выполняться команды следующей строки. В результате выполнения представленных на рисунке 2.33 трех команд откроется окно проекта с размещенными на нем элементами управления (если окно не открылось, то нужно выбрать его кнопку, которая появилась на Панели задач).

После этого нужно ввести в поля соответствующие числа и выбрать кнопку *Сложить*. В итоге открывается вкладка с текстом программы, который вошел в файл с расширением имени pas (англ. *pascal* - паскаль). Для рассматриваемого проекта это файл Unit1.pas. На рисунке 2.34 представлены команды этой части программы, и ее начало подсвечивается цветной полоской. При каждом следующем нажатии клавиши F7 будут выполняться команды следующей строки. В результате последовательного нажатия клавиши F7 выполнятся все приведенные команды, в результате чего в надпись и в окно сообщений будет выведен результат. Понятно, что все комментарии при выполнении будут пропущены.

Для того чтобы прервать режим пошагового исполнения, нужно нажать комбинацию клавиш Ctrl+F2.

Ясно, что рассмотренное выше пошаговое исполнение программы лишь продемонстрировало сам процесс и ничего нам не дало для выяснения того места программы, которое приводит к логической ошибке, если такая существует. Ведь мы не видели, как изменяются значения переменных в ходе такого исполнения. А именно это может нам подсказать, какие команды являются причиной логической ошибки.

Чтобы во время пошагового исполнения программы следить за изменением значений переменных, нужно открыть окно Watch List (англ. *watch list* - список просмотра) и поместить в это окно имена переменных, за значениями которых мы хотим наблюдать в ходе пошагового исполнения программы.

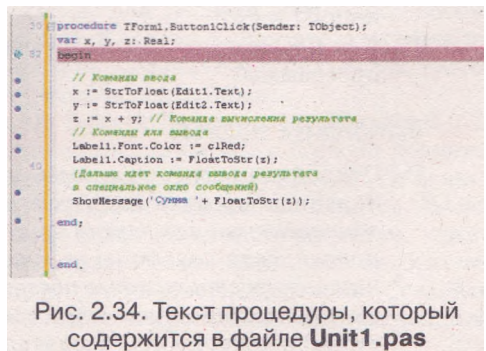


Рис. 2.34. Текст процедуры, который содержится в файле **Unit1.pas**

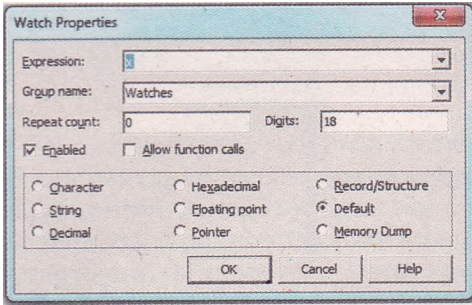


Рис. 2.35. Окно Watch Properties

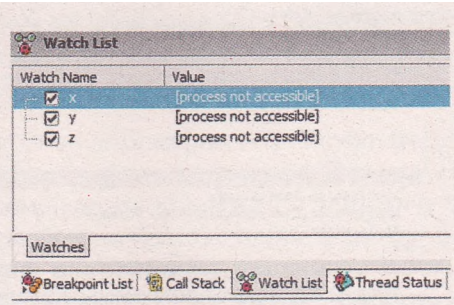


Рис. 2.36. Окно Watch List

Для того чтобы поместить имя переменной в окно Watch List, нужно нажать комбинацию клавиш **Ctrl+F5** или выполнить *Run => Add Watch* (англ. *add watch* - добавить просмотр). В итоге сначала открывается окно Watch Properties (англ. *watch properties* - свойства просмотра) (рис. 2.35). В первое поле этого окна нужно ввести имя переменной и выбрать кнопку **OK**.

После этого открывается окно Watch List (рис. 2.36) с занесенным в него именем переменной. Таким образом можно в это окно занести имена всех тех переменных, за значениями которых целесообразно наблюдать во время пошагового исполнения программы.

Во время пошагового исполнения программы значения этих переменных будут изменяться и отображаться в окне Watch List. Внимательно наблюдая за этими изменениями и сравнивая эти значения с ожидаемыми, можно заметить, в какой момент исполнения программы значение определенной переменной становится неожиданным. Именно эта команда и может быть причиной логической ошибки.

### Проверьте себя

- 1°. В чем заключается процесс тестирования программы?
- 2°. Поясните, для чего выполняется тестирование программы.
- 3°. Что такое комментарий в программе? Каким образом можно включить комментарий в текст программы?
- 4\*. Поясните, для чего в программу включают комментарии.
- 5°. Для чего программы исполняют пошагово?
- 6°. Поясните, в чем заключается процесс пошагового исполнения программы.
- 7°. Прокомментируйте ход пошагового исполнения программы.
- 8°. Чем отличаются пошаговое исполнение программ с использованием клавиш **F7** и **F8**?
- 9\*. Как просмотреть значения переменных во время пошагового исполнения программы?

### Выполните задания

1. Откройте проект, например из папки Тема 2\Задания 2.6\Проект 2.6.1. Заданием проекта является поиск решения задачи: *Два поезда выехали одновременно навстречу друг другу. Найдите, через какое время они встретятся, если известны расстояние между поездами в момент начала движения и скорость каждого поезда.* Выполнение обработчика события **OnClick** кнопки должно привести к выводу искомого времени, если значения расстояния и скоростей вводятся в поля. Для заданных начальных данных



вычислите ожидаемые результаты: а) расстояние - 1000 км, скорости - 120 км/ч и 130 км/ч; б) расстояние - 600 км, скорости - 80 км/ч и 120 км/ч; в) расстояние - 1200 км, скорости - 115 км/ч и 85 км/ч.

Проверьте, совпадают ли результаты, полученные в ходе исполнения проекта, с вычисленными вами результатами. Поясните, в чем может быть ошибка.



- 2•. Создайте три тестовых набора входных данных для тестирования проекта, предназначенного для решения задачи: *Два поезда выехали одновременно с одной станции в противоположных направлениях. Найдите расстояние, на котором они будут друг от друга через заданное время, если известна скорость каждого поезда.* Вычислите ожидаемые результаты по составленным тестам. Откройте проект, например из папки **Тема 2\Задания 2.6\Проект 2.6.2**. Протестируйте его на ваших тестовых данных. Сделайте вывод, содержит ли программа логические ошибки. Если да, то в чем может быть ошибка?
- 3\*. Откройте проект, например из папки **Тема 2\Задания 2.6\Проект 2.6.3**. Просмотрите программный код и на основе анализа комментариев сделайте предположение, какую задачу решает этот проект. Опираясь на ваше предположение, составьте три тестовых набора входных данных для тестирования проекта. Вычислите ожидаемые результаты. Протестируйте проект на ваших тестовых данных. Сделайте вывод, правильным ли было ваше предположение.
- 4•. Создайте проект для вычисления длины окружности и площади круга по значению радиуса, которое пользователь вводит в поле. Поставьте комментарии для обозначения строк, в которых вычисляются искомые значения. Создайте в собственной папке папку **Проект 2.6.4** и сохраните в ней проект.
- 5°. Откройте проект, например из папки **Тема 2\Задания 2.6\Проект 2.6.5**. Исполните программу в пошаговом режиме. Обратите внимание на отображение хода исполнения программы.
- 6•. Откройте проект, например из папки **Тема 2\Задания 2.6\Проект 2.6.6**. В ходе пошагового исполнения программы проследите за значениями переменных **S1** и **S2**.
- 7•. Ученики для праздника купили конфеты и печенье. Известны цены конфет и печенья и количество купленных сладостей каждого вида. Создайте проект для нахождения полной стоимости покупки. Создайте в собственной папке папку **Проект 2.6.7** и сохраните в ней проект. Исполните проект в пошаговом режиме, проследите за значениями переменных, которые содержат значение стоимости каждого товара.
- 8•. Откройте проект, например из папки **Тема 2\Задания 2.6\Проект 2.6.8**. В ходе пошагового исполнения программы проследите за значениями переменных  $x$ ,  $a$  и  $S$ . Измените проект так, чтобы в результате исполнения переменная  $S$  содержала сумму  $x + x^2 + x^4$  для введенного значения  $x$ . Добавьте к тексту программы комментарий, в котором укажите, какие изменения были внесены в программный код. Создайте в собственной папке папку **Проект 2.6.8** и сохраните в ней измененный проект.
- 9\*. Откройте проект, например из папки **Тема 2\Задания 2.6\Проект 2.6.9**. Этот проект предназначен для определения, на сколько больше времени потратил один из поездов на путь между городами, чем другой, по заданному расстоянию между городами и скоростям каждого из поездов. Выясните, содержит ли проект логические ошибки. Если да, определите, какие команды являются их причинами, и исправьте их. Запишите наборы тестовых данных, на которых вы обнаружили логические ошибки. Создайте в собственной папке папку **Проект 2.6.9** и сохраните в ней измененный проект.



## Практическая работа № 7. Отладка программ

**Внимание!** Во время работы, с компьютером соблюдайте правила безопасности и санитарно-гигиенические нормы.

1. Откройте Turbo Delphi 2006.
2. Создайте проект для определения, во сколько раз произведение двух заданных целых чисел больше суммы этих чисел.
3. Создайте три тестовых набора входных данных для тестирования проекта. Вычислите ожидаемые результаты. Протестируйте проект на ваших тестовых данных.
4. Поставьте комментарии для обозначения строк, в которых происходит вычисление суммы и произведения.
5. Исполните проект в пошаговом режиме. Проследите за значениями суммы и произведения. Запишите, какие сообщения отображаются в столбце Value для указанных переменных перед началом пошагового исполнения проекта, в ходе исполнения для одного из наборов начальных данных и после завершения исполнения.
6. Создайте в собственной папке папку Практическая 7 и сохраните в ней проект

### 2.7. Основные понятия математической логики



1. Что такое логическое выражение? Какие значения оно может принимать?
2. Опишите общий вид, количество аргументов и результаты функций AND, OR, NOT табличного процессора.
3. Какие свойства переменной вы знаете?
4. Что определяет тип переменной? Какие числовые типы вам известны?

В предыдущих пунктах мы рассматривали проекты, которые реализовывали линейные алгоритмы. Но для решения большинства задач линейных алгоритмов недостаточно. В проектах, которые реализуют другие виды алгоритмов, например алгоритмы с ветвлениями, используют логические выражения. Рассмотрим логические выражения и операции над ними.

#### Высказывания. Логические константы. Логические операции

Логика (греч. *Λόγος* - слово, смысл, мысль, язык) - наука о формах, методах и законах интеллектуальной познавательной деятельности, правильного мышления, о способах рассуждений.

Один из разделов логики - математическая логика - является наукой о законах математического мышления. К основным понятиям математической логики относится понятие высказывание. Под высказыванием понимают повествовательное предложение, о котором можно однозначно сказать, правильное (истинное) оно или неправильное (ложное). Высказывания обозначают большими буквами английского алфавита, например  $A = \langle \text{«Киев - столица Украины»} \rangle$ ,  $B = \langle \text{«}2 + 2 = 5 \rangle$ .

Основным свойством высказывания является его истинность, другие свойства считаются несущественными. Значения истинности высказывания обозначают 1 или true, если высказывание истинно, 0 или false, если высказывание ложно. Например, высказывание  $A = \langle \text{«Киев - столица Украины»} \rangle$  является истинным, то есть значение его свойства истинность равно true. Высказывание  $C = \langle \text{«}3 > 5 \rangle$  является ложным, то есть значение его свойства истинность равно false.



Значения true и false называются *логическими константами*.

Из заданных высказываний можно получить новые - составные, используя логические операции: отрицание, конъюнкцию (лат. *conjunctio* - объединение), дизъюнкцию (лат. *disjunctio* - разъединение) и др.



*Отрицанием* называется операция получения из высказывания  $A$  такого составного высказывания  $\bar{A}$  (читается «не  $A$ »), которое истинно тогда, когда  $A$  ложно, и ложно тогда, когда  $A$  истинно.

Использованию операции отрицания в русском языке соответствует употребление частицы «не» перед глаголом. Например, отрицанием высказывания  $A =$  «Существует наибольшее простое число» есть высказывание  $\bar{A} =$  «Не существует наибольшего простого числа».

Высказывание и его отрицание не могут быть одновременно истинными или одновременно ложными.

Для определения результатов выполнения логических операций в математической логике используют таблицы истинности. В таблице истинности для каждого возможного набора значений свойства истинность высказываний приводится соответствующее значение этого свойства для высказывания, которое является результатом выполнения логической операции над ними.

По определению, отрицание выполняется над одним высказыванием, которое может быть либо истинным, либо ложным. Поэтому таблица истинности операции отрицания имеет такой вид:

$A$	$\bar{A}$
0	1
1	0



*Конъюнкцией* двух высказываний  $A$  и  $B$  называется операция получения такого составного высказывания  $A \wedge B$  (читается « $A$  и  $B$ »), которое истинно тогда и только тогда, когда оба высказывания  $A$  и  $B$  истинные.

Использованию операции конъюнкции в русском языке соответствует употребление союза «и». Например, высказывание  $B =$  «Число 27 кратно 3 и число 27 кратно 9» является конъюнкцией двух высказываний: «Число 27 кратно 3» и «Число 27 кратно 9».

По определению, конъюнкция выполняется над двумя высказываниями. Каждое из них может быть либо истинным, либо ложным. Поэтому таблица истинности операции конъюнкции имеет такой вид:

$A$	$B$	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

Операцию конъюнкции также называют логическим умножением.



**Дизъюнкцией** двух высказываний  $A$  и  $B$  называется операция получения такого составного высказывания  $A \vee B$  (читается «*А или В*»), которое истинно тогда и только тогда, когда истинно хотя бы одно из высказываний  $A$  или  $B$ .

Использованию операции дизъюнкции в русском языке соответствует употребление союза «или». Например, высказывание  $C = «21 < 21»$  является дизъюнкцией двух высказываний: « $21 = 21$ » или « $21 < 21$ ».

По определению, дизъюнкция выполняется над двумя высказываниями. Каждое из них может быть либо истинным, либо ложным. Поэтому таблица истинности операции дизъюнкции имеет такой вид:

$A$	$B$	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Операцию дизъюнкции также называют логическим сложением.

При решении многих задач встречаются высказывания с переменными, в которых используются знаки сравнения:  $>$  (больше),  $<$  (меньше),  $=$  (равно),  $<>$  (не равно),  $>=$  (больше или равно),  $<=$  (меньше или равно), например  $x > 5$ ,  $y < z$ . Такие высказывания могут быть истинными при одних значениях переменных и ложными при других.

Высказывание  $x < 0$  является простым, а высказывание  $3 < x < 5$  - составным. Последнее можно представить как конъюнкцию двух простых высказываний:  $(x > 3) \wedge (x < 5)$ .

Например, чтобы определить, лежит ли точка с координатами  $(x; y)$  в первой координатной четверти, нужно определить истинность составного высказывания  $(x > 0)$  и  $(y > 0)$ . Для точки с координатами  $(3; -5)$  будем иметь  $(3 > 0)$  и  $(-5 > 0)$ . Поскольку первое утверждение истинно, а второе — ложно, то их конъюнкция ложна. То есть эта точка не лежит в первой четверти.

## Логические переменные. Логические выражения и их таблицы истинности. Логические формулы

Алгеброй логики (булевой логикой, алгеброй высказываний) называют раздел математической логики, в котором рассматриваются общие свойства выражений, составленных из высказываний с использованием логических операций.



Основные положения алгебры логики были сформулированы английским математиком и философом **Джорджем Булем** (1815-1864) (рис. 2.37). В 1854 году вышел его основной труд «Исследование законов мысли, на которых основаны математические теории логики и вероятности». Эта книга в наше время причислена к математической классике. Созданная им **булева логика** стала основным математическим инструментом для создания компьютеров.

В алгебре логики используется понятие логическая переменная.

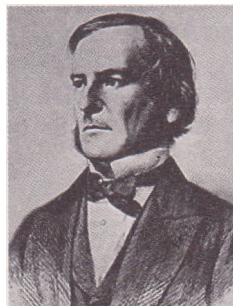


Рис. 2.37.  
Джордж Буль





**Логическая переменная** — это переменная, которая может принимать только значения `true` или `false`.

Заданием алгебры логики является определение истинности логических выражений — выражений, которые состоят из логических констант, логических переменных, логических операций, скобок, например  $(A \vee B) \wedge (A \vee B)$ .

Понятно, что значением логического выражения может быть только `true` или `false`.

Для логических операций, рассмотренных выше, определен такой приоритет операций: отрицание, конъюнкция, дизъюнкция. Для изменения этого порядка выполнения логических операций используют скобки.

Для вычисления значения истинности логического выражения можно использовать таблицы истинности.

Для построения и заполнения таблицы истинности нужно:

1. Вычислить количество возможных наборов значений логических переменных. Если выражение содержит  $n$  разных логических переменных, то возможных наборов значений этих переменных будет  $2^n$ . Это число определяет количество строк в таблице истинности.
2. Определить количество логических операций в логическом выражении. Это количество плюс количество логических переменных определяет количество столбцов в таблице.
3. Заполнить первые  $n$  столбцов всеми возможными наборами значений логических переменных.
4. Заполнить каждый следующий столбец значениями, полученными - при выполнении очередной логической операции. Очередность устанавливается согласно названному приоритету операций.

В последнем столбце таблицы будут получены все возможные значения истинности заданного логического выражения.

Например, выражение  $A \vee B \wedge A$  содержит две логические переменные, т. е.  $n = 2$ . Следовательно, всего существует 4 набора возможных значений этих переменных ( $2^n = 2^2 = 4$ ). Выражение содержит 3 логические операции: отрицание, дизъюнкция и конъюнкция. Таким образом, таблица истинности будет состоять из 4 строк и 5 столбцов. Первой, согласно приоритету, будет выполняться операция отрицания, второй - конъюнкции, последней - дизъюнкции, в которой нужно использовать результаты первых двух операций.

Получим такую таблицу истинности:

A	B	1	2	3
		$\bar{A}$	$B \wedge A$	$\bar{A} \vee (B \wedge A)$
0	0	1	0	1
0	1	1	0	1
1	0	0	0	0
1	1	0	1	1

Анализируя последний столбец таблицы истинности, делаем вывод, что выражение будет иметь значение `false` только в случае, когда логические переменные имеют такие значения:  $A = true$ ,  $B = false$ . Во всех остальных случаях значением логического выражения будет `true`.



Два логических выражения называются *равносильными*, если они принимают одинаковые значения при одних и тех же наборах значений логических переменных, которые входят в эти выражения.

Равносильность двух логических выражений образует *логическую формулу*.

Приведем несколько интересных и полезных логических формул:

1.  $\overline{\overline{A}} = A$  – формула двойного отрицания;
2.  $A \wedge (A \vee B) = A$  – формула поглощения;
3.  $\overline{A \vee B} = \overline{A} \wedge \overline{B}$  – формула отрицания дизъюнкции (закон де Моргана).

Для доказательства этих равносильностей можно составить и сравнить таблицы истинности логических выражений в правой и левой частях. Предлагаем вам сделать это самостоятельно.



**Август де Морган** (1806-1871) (рис. 2.38) – шотландский математик и логик, профессор математики Лондонского университетского колледжа, первый президент Лондонского математического общества. Результаты своих исследований по логике получил независимо от Джорджа Буля и изложил в 1847 году.



Рис. 2.38. Август де Морган



### Проверьте себя

- 1°. Что изучает наука логика? Что изучает математическая логика?
- 2°. Что такое высказывание? Приведите примеры истинных и ложных высказываний.
- 3\*. Какие предложения не являются высказываниями?
- 4°. Что называется логической константой?
- 5°. Как образуются составные высказывания? Приведите примеры.
- 6°. Что такое отрицание? Приведите примеры.
- 7°. Для чего используют таблицы истинности?
- 8°. Охарактеризуйте таблицу истинности отрицания.
- 9°. Что такое конъюнкция? Приведите примеры. Постройте таблицу истинности.
- 10°. Что такое дизъюнкция? Приведите примеры. Постройте таблицу истинности.
- 11°. Какая переменная называется логической?
- 12\*. Чем логическая переменная отличается от логического высказывания?
- 13°. Из чего состоят логические выражения? Какие значения они могут принимать?
- 14°. Назовите приоритет логических операций.
- 15°. Приведите алгоритм построения и заполнения таблицы истинности логического выражения.
- 16°. Какие логические выражения называются равносильными? Что такое логическая формула?



### Выполните задания

1. Постройте отрицания высказываний и выясните их истинность:
  - а) Число 3 является делителем числа 545.
  - б) Автомобиль не имеет права ехать на красный свет.
  - в) Существуют параллелограммы с прямыми углами.
  - г) Уравнение  $2x^2 - 3x + 1 = 0$  имеет целый корень.

- 2\*. Среди приведенных составных высказываний найдите конъюнкции и дизъюнкции и определите их истинность:
- Число 36 кратно 4 и кратно 6.
  - $17 < 42 < 18$ .
  - Число 2 простое или четное.
  - $\triangle ABC$  является остроугольным, прямоугольным или тупоугольным.
  - Диагонали любого параллелограмма перпендикулярны и точкой пересечения делятся пополам.
  - $7^2 = 49$  и  $(-7)^2 = 49$ .
  - Если треугольник равнобедренный, то он равносторонний.
  - $21 \leq 21$ .
  - $21 \leq 18$ .

Конъюнкция	
Дизъюнкция	
Ни конъюнкция, ни дизъюнкция	

- 3\*. Определите истинность или ложность составных высказываний для приведенных значений переменных:
- $x > 0$  и  $y > 0$  или  $x < 0$  и  $y < 0$ ;
  - $x > 0$  и не  $y < 0$  или  $x < 0$  и  $y > 0$ .
- Значения переменных:  
 а)  $x = 5, y = 8$ ; б)  $x = 5, y = -8$ ; в)  $x = -b, y = 8$ ; г)  $x = -5, y = -8$ .

- 4\*. Пусть  $A = \text{«Иванов изучает, английский язык»}$ ,  $B = \text{«Иванов имеет оценку 8 по информатике»}$ . Сформулируйте высказывания:

- а)  $A \wedge B$ ; б)  $\overline{A} \vee B$ ; в)  $A \vee \overline{A} \wedge B$ ; г)  $\overline{A} \vee \overline{B} \wedge A$ .

- 5\*. Укажите порядок выполнения операций в логических выражениях:

- а)  $\overline{A} \vee \overline{A} \wedge B$ ; в)  $(A \vee \overline{B}) \wedge (A \vee B)$ ;  
 б)  $A \vee B \wedge A$ ; г)  $A \wedge B \vee C \wedge D$ .

- 6\*. Постройте таблицы истинности логических выражений:

- а)  $A \vee \overline{A} \wedge B$ ; б)  $A \vee \overline{A} \wedge B$ ;  
 в)  $A \wedge B \vee \overline{B}$ ; г)  $(A \vee \overline{B}) \wedge (\overline{A} \vee \overline{B})$ ;  
 д)  $\overline{A} \vee \overline{B}$ ;  
 е)  $A \wedge \overline{B} \vee \overline{A} \wedge B$ .

- 7\*. Докажите логические формулы:

- а)  $A \wedge A = A$ ;  
 б)  $\overline{A \wedge B} = \overline{A} \vee \overline{B}$ ;  
 в)  $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$ ;  
 г)  $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$ ;  
 д)  $A \wedge (A \vee B) = A$ .

## 2.8. Ветвление в алгоритмах и программах



- Что такое алгоритм? Назовите основные блоки блок-схемы алгоритма и объясните их назначение.
- Какие алгоритмы (фрагменты алгоритмов) называются линейными?
- Какие выражения называются логическими? Какие значения они могут принимать?
- Назовите логические операции, приведите определение каждой из них.
- Назовите логические функции табличного процессора Excel 2007. Чему равны их значения в зависимости от значений аргументов?

### Алгоритмы с ветвлением

В предыдущих пунктах было рассмотрено несколько линейных алгоритмов, в частности алгоритмы для решения задач на вычисление значений

## Глава 2

арифметического выражения для исполнителя, который умеет выполнять арифметические операции. Рассмотрим пример задачи, алгоритм решения которой не является линейным.

**Задача 1.** Вычислить значение выражения  $(a - b) / (c - d)$ , где  $a, b, c, d$  - действительные числа.

Обратим внимание на то, что значение этого выражения можно вычислить не для любого набора значений переменных  $a, b, c, d$ . То есть если значение разности  $c - d$  равно нулю, то значение выражения  $(a - b) / (c - d)$  вычислить нельзя, а если не равно, то можно.

Это значит, что система команд исполнителя должна содержать команду сравнения двух чисел, например такую: « $s = t$ ?», где  $s$  и  $t$  - либо числа, либо переменные или выражения, которые принимают определенные числовые значения. Такая команда является примером команды проверки условия. Результатом выполнения команды проверки условия может быть либо истина (условие выполняется), либо ложь (условие не выполняется).

С другой стороны, вы уже знаете, что сравнение « $s = t$ ?» можно рассматривать как высказывание или как простое логическое выражение, которое будет принимать значение true или false в зависимости от конкретных значений переменных  $s$  и  $t$ . И тогда команду проверки условия

можно интерпретировать как команду вычисления значения логического выражения.

Алгоритм решения задачи 1 будет выглядеть так:

1. Ввести значения переменных  $a, b, c, d$ .
2.  $x := c - d$ .
3. Вычислить значение логического выражения  $x = 0$ .
4. Если вычисленное значение логического выражения true, то вывести сообщение: «Выражение значения не имеет: деление на нуль», после чего выполнить команду 8, если false, то выполнить команду 5.
5.  $y := a - b$ .
6.  $z := y / x$ .
7. Вывести значение переменной  $z$ .
8. Закончить выполнение алгоритма.

Команды 1-3 приведенного алгоритма будут выполняться при любом наборе значений переменных  $a, b, c, d$ . Последующее выполнение этого алгоритма будет зависеть от значения логического выражения, вычисленного в команде 3. Если это значение false, то будут выполняться команды 5-8, а если true, то будут выполняться команда вывода

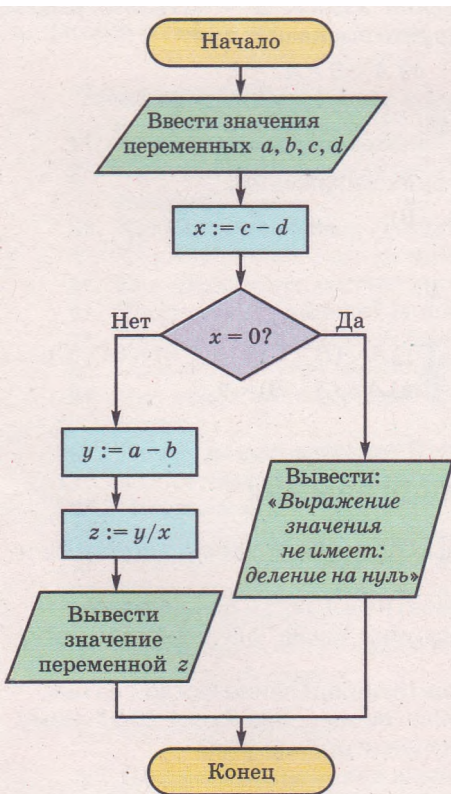



Рис. 2.39. Блок-схема алгоритма вычисления значения выражения  $(a - b) / (c - d)$



сообщения: «Выражение значения не имеет: деление на нуль» и команда 8.

В блок-схеме алгоритма команды проверки условия или вычисления значения логического выражения обозначаются блоком Решение . Поскольку результатом выполнения этих команд может быть либо true, либо false, то из этого блока есть два выхода. Выход Да означает, что результатом проверки условия является true, а выход Нет — что результатом проверки условия является false.

Рассмотрим фрагмент блок-схемы алгоритма (рис. 2.39) от блока Решение до блока Терминатор (не включая этот блок).



Характерной чертой этого фрагмента алгоритма является то, что при каждом его выполнении некоторые команды будут выполняться, причем каждая по одному разу, а некоторые — выполняться не будут. Это зависит от результата выполнения команды проверки условия (команды вычисления значения логического выражения).

Такой фрагмент алгоритма называется ветвлением.

Исполним приведенный алгоритм для двух наборов данных. Эти наборы подберем так, чтобы проиллюстрировать ход выполнения алгоритма для каждого из двух возможных результатов вычисления значения логического выражения.

Команда	Результат выполнения
<i>Исполнение для первого набора данных</i>	
Ввести значения переменных $a, b, c, d$	$a = 5, b = 6, c = -3, d = 5$
$x := c - d$	$x = -3 - 5 = -8$
Вычислить значение логического выражения $x = 0$	$(-8 = 0) = \text{false}$
$y := a - b$	$y = 5 - 6 = -1$
$z := y/x$	$z = -1/(-8) = 0,125$
Вывести значение переменной $z$	$z = 0,125$
<i>Исполнение для второго набора данных</i>	
Ввести значения переменных $a, b, c, d$	$a = 12,3, b = -1, c = 8,2, d = 8,2$
$x := c - d$	$x = 8,2 - 8,2 = 0$
Вычислить значение логического выражения $x = 0$	$(0 = 0) = \text{true}$
Вывести сообщение: «Выражение значения не имеет: деление на нуль»	Сообщение: «Выражение значения не имеет: деление на нуль»

*Обращаем ваше внимание:*

- приведенный алгоритм содержит как ветвление, так и линейные фрагменты;
- в ветвлениях можно использовать как простые логические выражения, так и составные.

В алгоритмах используют ветвление двух видов: полное ветвление (рис. 2.40) и неполное ветвление (рис. 2.41).

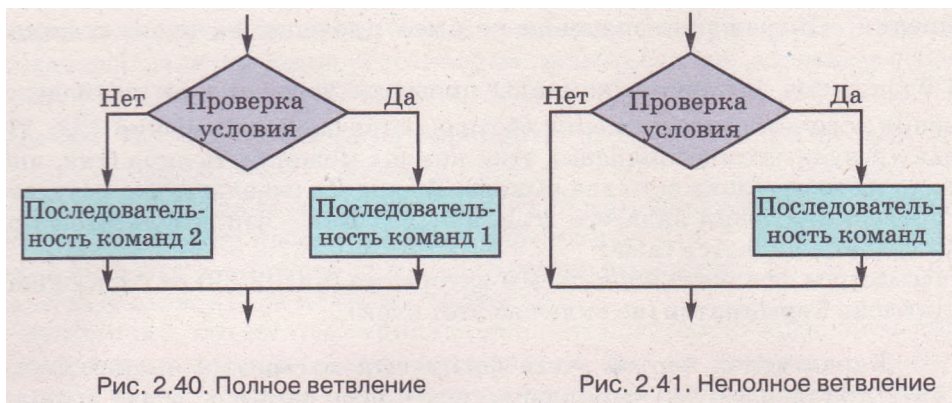


Рис. 2.40. Полное ветвление

Рис. 2.41. Неполное ветвление

Выполнение полного ветвления происходит так: исполнитель выполняет команду проверки условия (команду вычисления значения логического выражения); если результат выполнения этой команды true, то исполнитель выполняет последовательность команд 1, после чего переходит к выполнению первой команды следующего фрагмента алгоритма; если же результат выполнения этой команды false, то исполнитель выполняет последовательность команд 2, после чего также переходит к выполнению первой команды следующего фрагмента алгоритма.

Выполнение неполного ветвления отличается от выполнения полного ветвления тем, что если результат выполнения команды проверки условия false, исполнитель сразу переходит к выполнению первой команды следующего фрагмента алгоритма.

Внутри ветвления могут находиться как линейные фрагменты алгоритма, так и другие ветвления. Приведем пример алгоритма с ветвлением в ветвлении.

**Задача 2.** Даны два числа. Определить, равны ли они. Если нет, то какое из них больше?

Блок-схема алгоритма решения этой задачи представлена на рисунке 2.42.

Приведем пример еще одной задачи, алгоритм решения которой содержит ветвление.

**Задача 3.** Есть девять одинаковых на вид монет, одна из которых фальшивая и легче других. Двумя взвешиваниями на весах без гирь определить фальшивую монету.

Составим алгоритм для исполнителя с такой системой команд:

1. Взять указанную кучку монет.
2. Разделить указанную кучку монет на три равные кучки.
3. Положить на весы указанные кучки монет.
4. Проверить условие «Весы в равновесии?».
5. Определить при взвешивании, какая из кучек монет легче.
6. Сообщить результат.

Для исполнителя с такой системой команд алгоритм решения задачи 3 таков:

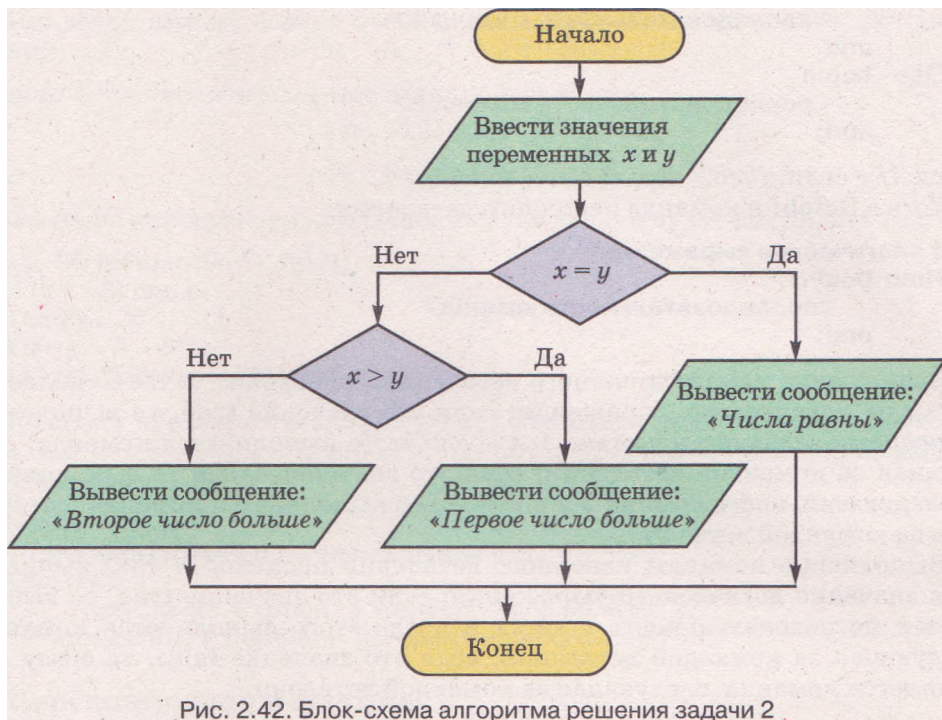


Рис. 2.42. Блок-схема алгоритма решения задачи 2.

1. Взять данную кучку из девяти монет.
2. Разделить взятую кучку монет на три равные кучки.
3. Положить на весы первую и вторую кучки монет.
4. Проверить условие «Весы в равновесии?».
5. Если истина, то взять третью кучку монет, если ложь, то взять более легкую кучку.
6. Разделить взятую кучку монет на три равные кучки.
7. Положить на весы первую и вторую кучки монет.
8. Проверить условие «Весы в равновесии?».
9. Если истина, то сообщить: «Фальшивой является монета, которая не положена на весы», если ложь - сообщить: «Фальшивой является более легкая монета».

*Обращаем ваше внимание:* в алгоритме решения задачи 3 выполнение или невыполнение тех или иных команд алгоритма зависит от того, в какую кучку попадет фальшивая монета. Например, если во время первого взвешивания весы не будут в равновесии, то команда «Взять третью кучку монет» не будет выполнена. Но если монеты распределить на кучки по-другому, то может случиться так, что при первом взвешивании весы будут в равновесии, и тогда не будет выполнена команда «Взять более легкую кучку монет».

## Ветвление в Delphi

Команда полного ветвления в Delphi имеет такой общий вид:

```

If <логическое выражение>
Then begin

```

```
    < последовательность команд 1 >
end
Else begin
    < последовательность команд 2 >
end;
```

(англ. *if* - если, *then* - тогда, *else* - иначе).

Есть в Delphi и команда неполного ветвления:

```
If <логическое выражение>
Then begin
    <последовательность команд>
end;
```

Выполнение команды полного ветвления происходит так: вычисляется значение логического выражения; если это значение true, то выполняется последовательность команд 1 и после этого выполняется команда, следующая за командой ветвления; если это значение false, то выполняется последовательность команд 2 и после этого выполняется команда, следующая за командой ветвления.

Выполнение команды неполного ветвления происходит так: вычисляется значение логического выражения; если это значение true, то выполняется последовательность команд и после этого выполняется команда, следующая за командой ветвления; если это значение false, то сразу выполняется команда, следующая за командой ветвления.

*Обращаем ваше внимание:*

- если после ключевых слов Then и Else следуют только по одной команде, то операторные скобки begin и end можно не ставить;
- перед Else ставить точку с запятой нельзя.

Если записать в Delphi фрагмент программы решения задачи 1 (блок-схема соответствующего алгоритма представлена на рис. 2.39), то он будет выглядеть так:

```
a := StrToFloat (Edit1.Text);
b := StrToFloat (Edit2.Text);
c := StrToFloat (Edit3.Text);
d := StrToFloat (Edit4.Text);
x := c-d;
If x = 0
Then Label1.Caption := 'Выражение значения не имеет: деление на нуль'
Else begin
    y := a-b;
    z := y/x;
    Label1.Caption := FloatToStr (z);
end;
```

*Обращаем ваше внимание:* вычисление значения выражения можно было бы записать не двумя командами, а одной:  $z := (a-b)/x$ .

Логическое выражение в команде ветвления может быть простым или составным. Составные логические выражения образуются из простых путем использования рассмотренных в предыдущем пункте операций конъюнкции (and), дизъюнкции (or), отрицания (not). Порядок выполнения этих операций в Delphi такой же, как и в математической логике.



Приведем пример задачи с использованием составных логических выражений.

Задача 4. Вычислить значение заданной функции

$$y = \begin{cases} 2x - 12, & \text{если } -3 < x < 5, \\ 7 - 8x & \text{для всех других } x. \end{cases}$$



Фрагмент программы для решения этой задачи будет выглядеть так:

```
x := StrToFloat (Edit1 .Text);
If (x > -3) and (x < 5)
Then y := 2*x- 12
Else y := 7 -8*x;
Label1 .Caption := FloatToStr (y);
```

Команда ветвления в Delphi может включать другую команду ветвления. Вот как будет выглядеть команда ветвления для алгоритма, приведенного на рисунке 2.42:

```
If x = y
Then Label1 .Caption := 'Числа равны'
Else If x > y
Then Label1 .Caption := 'Первое число больше'
Else Label1 .Caption := 'Второе число больше';
```

## Переключатели и флажки

Для реализации ветвления в проекте можно использовать такие элементы управления, как переключатели и флажки. Работая с текстовыми и графическими редакторами, редактором презентаций, другими программами, вы неоднократно видели, что при выборе переключателя внутри появляется метка , а при выборе флажка - метка .

Напомним, что из группы переключателей может быть выбран только один, а в группе флажков метка может быть установлена либо на одном из них, либо на нескольких, либо ни на одном. Переключатели предназначены для того, чтобы в ходе выполнения проекта выбрать один из возможных вариантов. Например, установить для определенного свойства объекта одно значение из некоторого определенного набора значений. Флажки предназначены для того, чтобы в ходе выполнения проекта выбрать либо один из возможных вариантов, либо несколько. Например, установить значение одного или нескольких свойств объекта.

Для размещения на форме переключателя нужно установить указатель на строку TRadioButton (англ. *radio button* - круглая кнопка) окна Tool Palette и дважды щелкнуть левую кнопку мыши, а для размещения флажка - установить указатель на строку TCheckBox (англ. *check box* - коробка для проверки) и также дважды щелкнуть левую кнопку мыши.

Переключатель и флажок имеют большинство свойств, которые мы рассматривали для формы, поля, кнопки и надписи, и еще некоторые, среди которых рассмотрим свойство Checked (англ. *checked* - выбранный). Оно может принимать одно из двух значений *True* или *False* и определяет, установлена метка в данном объекте или нет (выбран этот объект или нет). Поэтому значение этого свойства можно использовать в команде ветвления вместо логического выражения.

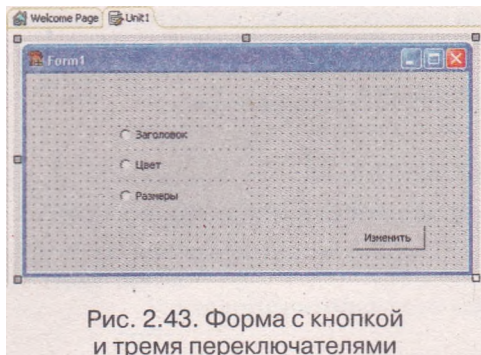


Рис. 2.43. Форма с кнопкой и тремя переключателями

Рассмотрим, как можно использовать переключатели для реализации ветвления в проекте.

Пусть, например, нужно в ходе исполнения проекта изменять значение только одного из таких свойств формы: либо заголовок, либо цвет, либо размеры.

Для этого разместим на форме кнопку и три переключателя (рис. 2.43). При размещении переключателей на форме для каждого из них по умолчанию

устанавливается значение свойства `Checked` - `False`. До запуска проекта изменим в первом переключателе значение этого свойства на `True`. После этого на форме он будет выглядеть выбранным (рис. 2.43). Таким он будет и после запуска проекта на исполнение.

Создадим процедуру для обработки события `OnClick` кнопки и в ее текст поместим такую команду:

```
If RadioButton1.Checked
Then Form1.Caption := 'Заголовок изменен'
Else If RadioButton2.Checked
Then Form1.Color := cRed
Else begin
    Form1.Width := 600;
    Form1.Height := 300;
end;
```

Начинаться выполнение этой команды будет с определения значения свойства `RadioButton1.Checked`.

Если при наступлении события `OnClick` кнопки выбранным будет первый переключатель, то `RadioButton1.Checked = true`. Поэтому выполнится команда изменения заголовка формы, а другие части всей этой команды выполняться не будут.

Если при наступлении события `OnClick` кнопки выбранным будет второй переключатель, то `RadioButton1.Checked = false`. Поэтому будут выполняться команды после первого ключевого слова `Else`, то есть внутренняя команда ветвления. Во время ее выполнения определяется значение свойства `RadioButton2.Checked`. Поскольку выбран второй переключатель, то это значение равно `true`. Поэтому выполнится команда изменения цвета формы, а другие части команды выполняться не будут.

Если же при наступлении события `OnClick` кнопки выбранным будет третий переключатель, то `RadioButton1.Checked = false`. Поэтому будут выполняться команды после первого ключевого слова `Else`, то есть внутренняя команда ветвления. При ее выполнении определяется значение свойства `RadioButton2.Checked`. Поскольку второй переключатель не выбран, то его значение `false`. Тогда будут выполняться команды после второго ключевого слова `Else`, то есть две команды изменения размеров формы.

Если перед запуском проекта на выполнение ни у одного переключателя не изменять значение свойства `Checked`, установленное по умолчанию, то при запуске проекта ни один из них не будет выбранным. Если и во время

исполнения проекта не выбрать ни один из переключателей и выбрать кнопку, то изменятся размеры формы (подумайте самостоятельно, почему произойдут именно такие изменения). Чтобы этого не произошло, нужно либо обязательно выбирать один из переключателей, либо после второго ключевого слова Else вставить еще одну команду ветвления:

```

If RadioButton1.Checked
Then Form1.Caption := 'Заголовок изменен'
Else If RadioButton2.Checked
    Then Form1.Color := cIRed
    Else If RadioButton3.Checked
        Then begin
            Form1.Width :=600;
            Form1.Height := 300;
        end;
    
```

Поясните, как будет выполняться эта команда, если ни один из переключателей не будет выбран.

Рассмотрим теперь, как можно использовать флажки для реализации ветвления.

Пусть, например, нам нужно в ходе выполнения проекта изменять значение либо только одного из свойств формы (или заголовок, или цвет, или размеры), либо нескольких из них, либо не изменять значения ни одного из них.

Для этого разместим на форме кнопку и три флажка (рис. 2.44). При размещении флажков на форме для каждого из них по умолчанию автоматически устанавливается значение свойства `Checked` - `False`. Поскольку при выполнении проекта может быть не установлена метка ни одного флажка, то целесообразно оставить такие начальные значения свойства `Checked` всех флажков.

Создадим процедуру для обработки события `OnClick` кнопки и в ее текст поместим такие три последовательные (не вложенные друг в друга) команды ветвления:

```

If CheckBox1.Checked
Then Form1.Caption := 'Заголовок изменен';
If CheckBox2.Checked
Then Form1.Color := cIRed;
If CheckBox3.Checked
Then begin
    Form1.Width := 500;
    Form1.Height := 200;
end;
    
```

Каждая из этих команд независимо друг от друга будет определять, выбран ли соответствующий флажок, и в зависимости от этого каждая из них будет или не будет выполнять соответствующие команды.

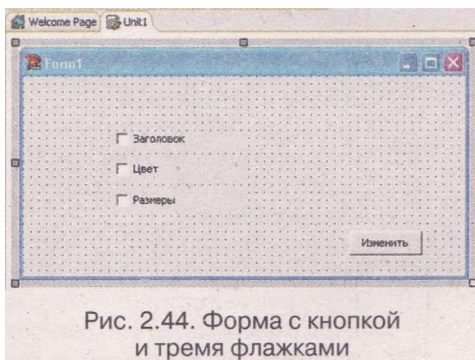


Рис. 2.44. Форма с кнопкой и тремя флажками

Из рассмотренных выше примеров можно сделать такие выводы:

- переключатели используются тогда, когда нужно выбрать один и только один вариант действий из нескольких возможных;
- для обработки состояния переключателей в процедуру включают команды ветвления, которые вложены друг в друга;
- флажки используются тогда, когда может быть выбран либо один, либо несколько, либо не выбран ни один вариант действий из нескольких возможных;
- для обработки состояния флажков в процедуру включают последовательные команды ветвления, независимые друг от друга.

### Проверьте себя

- 1°. Какие значения могут быть результатом выполнения команды проверки условия (вычисления значения логического выражения)?
- 2°. Как изображается команда проверки условия в блок-схеме алгоритма?
- 3°. Назовите характерные особенности ветвления.
- 4°. Изобразите блок-схемы полного и неполного ветвления. Опишите особенности их выполнения.
- 5\*. Приведите примеры правил русского языка, математики, других предметов, которые содержат ветвление.
- 6\*. Приведите примеры жизненных ситуаций, которые можно описать алгоритмом с ветвлением.
- 7°. Приведите общий вид команд полного и неполного ветвления в Delphi.
- 8°. Поясните выполнение команд полного и неполного ветвления в Delphi.
- 9°. Какие операции используют для получения составных логических выражений в Delphi?
- 10°. Для чего используют флажки и переключатели?
- 11°. Что определяет свойство Checked для переключателей и флажков?
- 12°. Поясните отличия использования команд ветвления при составлении проектов с флажками и переключателями.
- 13°. Поясните ход выполнения фрагментов программ:

а) **If** <логическое выражение 1 >  
**Then begin**  
     <команды1>  
     **end**  
**Else If** <логическое выражение 2>  
     **Then begin**  
         <команды 2>  
         **end**  
     **Else begin**  
         <команды 3>  
         **end;**

б) **If** <логическое выражение 1 >  
**Then begin**  
     <команды 1 >  
     **end;**  
**If** <логическое выражение 2>  
     **Then begin**  
         <команды 2>  
         **end;**

### Выполните задания

1. Составьте блок-схему алгоритма вычисления значения выражения  $(a + b) - c : a$ . Исполните его при разных значениях  $a, b, c$ . Подберите числа так, чтобы каждый раз исполнение алгоритма происходило по-разному.
2. Составьте блок-схему алгоритма вычисления значения выражения  $(a + b) - c : (a - 2b)$ . Исполните его при разных значениях  $a, b, c$ . Подберите числа так, чтобы каждый раз исполнение алгоритма происходило по-разному.





3. Составьте блок-схему алгоритма нахождения  $x$  из уравнения  $ax = b$ . Исполните его при: а)  $a = 2, b = -8$ ; б)  $a = 20, b = 0$ ; в)  $a = 0, b = 12$ ; г)  $a = 0, b = 0$ .
- 4\*. Составьте блок-схему алгоритма решения квадратного уравнения. Исполните его при разных значениях коэффициентов уравнения.
- 5\*. Дано некоторое количество одинаковых на вид монет. Известно, что среди них есть одна фальшивая и она тяжелее настоящей. Составьте алгоритм выявления фальшивой монеты наименьшим количеством взвешиваний на чашечных весах без гирь, если количество монет равно: а) 3; б) 9; в) 27; г) 30.
- 6\*. Создайте проект, который определяет меньшее число среди трех разных чисел. Создайте в собственной папке папку Проект 2.8.6 и сохраните в ней проект.
- 7\*. Создайте проект, который по градусной мере двух углов треугольника вычисляет градусную меру третьего угла и определяет вид этого треугольника по его углам (остроугольный, прямоугольный или тупоугольный). Создайте в собственной папке папку Проект 2.8.7 и сохраните в ней проект.
- 8\*. Создайте проект, в котором кнопка будет перемещаться на 20 пикселей вправо при наведении на нее указателя. После того как кнопка исчезнет за правой границей окна, она должна появиться возле его левой границы. Создайте в собственной папке папку Проект 2.8.8 и сохраните в ней проект.
- 9\*. Создайте проект, в котором можно вычислить значение функции

$$y = \begin{cases} 15 - 3x, & \text{если } -1 < x < 4, \\ 6 + 4x & \text{для всех других } x. \end{cases}$$

Создайте в собственной папке папку Проект 2.8.9 и сохраните в ней проект.

- 10\*. Создайте проект, в котором можно вычислить либо сумму, либо разность, либо произведение, либо частное двух произвольных действительных чисел в зависимости от выбранного переключателя. Создайте в собственной папке папку Проект 2.8.10 и сохраните в ней проект.
- 11\*. Создайте проект, в котором можно вычислить сумму, разность, произведение, частное двух произвольных действительных чисел в зависимости от установленных меток флажков. Создайте в собственной папке папку Проект 2.8.11 и сохраните в ней проект.

## Практическая работа № 8. Составление программ с ветвлениями

**Внимание!** Во время работы с компьютером соблюдайте правила безопасности и санитарно-гигиенические нормы.

- Откройте среду разработки Turbo Delphi 2006 и создайте новый проект.
- Разместите на форме элементы управления для решения задачи: *Известны длины сторон двух прямоугольников. Вычислить площади прямоугольников и сравнить их. Определить, являются ли прямоугольники квадратами. Выяснить, поместится ли один прямоугольник внутрь другого. Сообщить полученные результаты.* В проекте:
  - составьте обработчик события OnClick первой кнопки для вычисления площадей прямоугольников и вывода сообщения о том, у какого прямоугольника площадь больше;
  - составьте обработчик события OnClick второй кнопки, выполнение которого в зависимости от установленных флажков приведет к определению того, является ли соответствующий прямоугольник квадратом;
  - составьте обработчик события OnClick третьей кнопки, выполнение которого приведет к определению, поместится ли первый прямоугольник внутрь второго или второй прямоугольник внутрь первого в зависимости от выбранного переключателя;

- 4) выполните составленные процедуры и убедитесь, что результаты их работы правильные.
3. Создайте в собственной папке папку Практическая 8 и сохраните в ней проект.

## 2.9. Циклы в алгоритмах и в программах



1. Какие процессы называются циклическими? Приведите примеры.
2. Опишите характерные свойства линейных алгоритмов (фрагментов алгоритмов), алгоритмов с ветвлением.
3. Как выглядит и как выполняется команда полного ветвления в Delphi?
4. Как выглядит и как выполняется команда неполного ветвления в Delphi?
5. Как и для чего используются переключатели и флажки?

### Циклы в алгоритмах

В линейных алгоритмах и в алгоритмах с ветвлением каждая команда алгоритма могла быть выполнена не более одного раза. Но для решения многих задач нужно составлять алгоритмы, команды которых могут быть выполнены более одного раза. Рассмотрим примеры таких задач.

**Задача 1.** Есть бочка и ведро. Используя ведро, наполнить бочку водой из колодца.

Рассмотрим исполнителя с такой системой команд:

1. Наполнить ведро водой.
2. Вылить воду из ведра в бочку.
3. Проверить условие «Бочка неполная?».

Поскольку из условия задачи неизвестно, есть ли в бочке вода, исполнитель должен сначала проверить условие «Бочка неполная?». Если результат этой проверки true, то он должен наполнить ведро водой, вылить ее из ведра в бочку и опять проверить условие «Бочка неполная?». И так до тех пор, пока результат проверки этого условия станет false. После этого можно закончить исполнение алгоритма.

Приведем алгоритм решения этой задачи для рассмотренного исполнителя в словесной форме и в виде блок-схемы (рис. 2.45).

1. Проверить условие «Бочка неполная?».
2. Если результат выполнения предыдущей команды true, то выполнить команду 3, если false, то закончить исполнение алгоритма.
3. Наполнить ведро водой.
4. Вылить воду из ведра в бочку.
5. Перейти к выполнению команды 1.

В этом алгоритме команды 3-5 могут быть выполнены более одного раза.

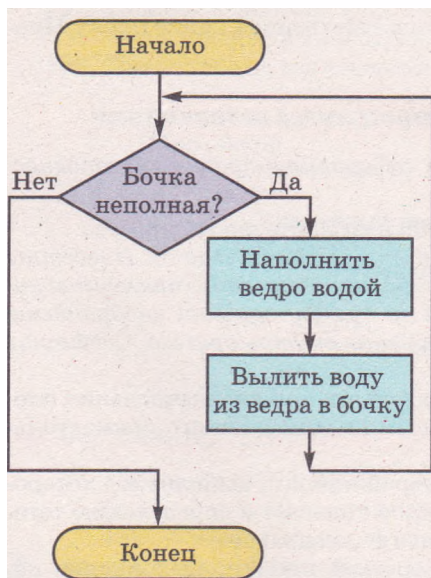


Рис. 2.45. Блок-схема алгоритма решения задачи 1

Очередное выполнение этих команд зависит от результата проверки условия в команде 1. Если этот результат true, то команды 3-5 выполняются еще раз, если же false, то эти команды больше не будут выполняться.

Обращаем ваше внимание: команды 3-5 именно могут быть выполнены более одного раза, а не обязательно выполняются более одного раза. Ведь возможно, что после первого же выливания воды из ведра в бочку она наполнится и выполнение алгоритма закончится. Кроме того, если бочка с самого начала была полной, то эти команды не выполняются ни разу.

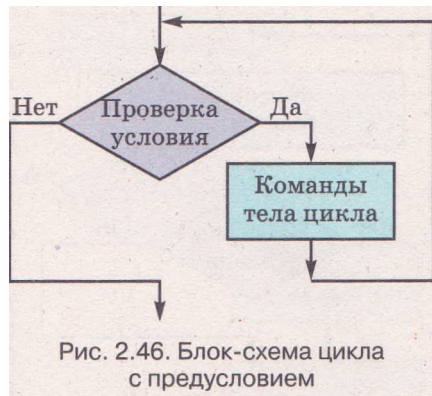


Рис. 2.46. Блок-схема цикла с предусловием



Фрагмент алгоритма, состоящий из команд, которые могут быть выполнены более одного раза, называется *циклом*. Алгоритмы, содержащие циклы, называются *алгоритмами с циклами*.

В приведенном алгоритме цикл состоит из трех команд: команды проверки условия и двух команд, которые образуют *тело цикла*.

Рассмотренный выше цикл называется *циклом с предусловием*, потому что условие проверяется перед началом выполнения команд тела цикла.

Общий вид цикла с предусловием представлен на рисунке 2.46. Выполнение такого цикла происходит следующим образом: исполнитель выполняет команду проверки условия (вычисления значения логического выражения); если результат выполнения этой команды true, то исполнитель выполняет команды тела цикла, после чего опять выполняет команду проверки условия (вычисления значения логического выражения); если же результат выполнения команды проверки условия (вычисления значения логического выражения) false, то исполнитель переходит к выполнению первой команды следующего фрагмента алгоритма.

Если бы в условии задачи 1 было известно, что бочка пустая, то исполнителю не нужно было бы сразу проверять условие «Бочка неполная?». Он должен был бы хотя бы один раз наполнить ведро водой, перелить воду из ведра в бочку и только после этого проверить условие «Бочка неполная?» (или условие «Бочка полная?», если она входит в систему его команд).

Блок-схема алгоритма решения такой задачи с использованием условия «Бочка полная?» представлена на рисунке 2.47.

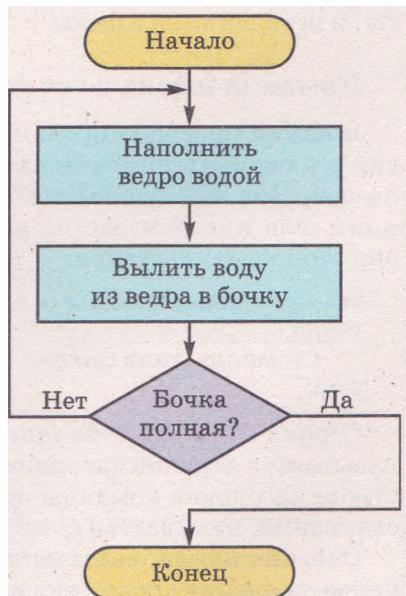


Рис. 2.47. Блок-схема алгоритма решения модифицированной задачи 1

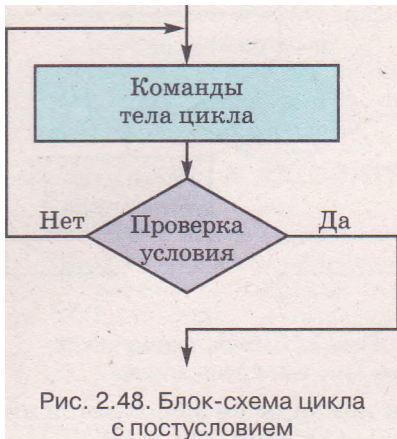


Рис. 2.48. Блок-схема цикла с постусловием

При выполнении приведенного алгоритма команды тела цикла обязательно будут выполняться хотя бы один раз, потому что команда проверки условия выполняется после выполнения команд тела цикла. Такой цикл называется циклом с постусловием.

Общий вид блок-схемы цикла с постусловием представлен на рисунке 2.48.

Выполнение такого цикла происходит так: исполнитель выполняет команды тела цикла, после чего выполняет команду проверки условия (вычисления значения логического выражения); если результат выполнения этой команды `false`, то исполнитель опять выполняет команды тела цикла; если

же `true`, то исполнитель переходит к выполнению первой команды следующего фрагмента алгоритма.

*Обращаем ваше внимание:* если в алгоритме, блок-схема которого представлена на рисунке 2.47, использовать условие «Бочка неполная?», то выполнение цикла продолжалось бы при результате `true` выполнения команды проверки условия и прекращалось бы при результате `false`.

Таким образом, мы рассмотрели три базовых структуры алгоритмов: линейные фрагменты (следование), ветвления и циклы (повторения). Доказано, что используя только эти три структуры, можно составить алгоритм решения любой задачи, если он существует.

Заметим, что большинство алгоритмов содержат и линейные фрагменты, и ветвления, и циклы.

## Команда цикла со счетчиком в Delphi

В языке программирования Delphi есть несколько команд, которые могут реализовать цикл. Одна из них - команда цикла со счетчиком. Ее целесообразно использовать в тех случаях, когда количество повторений команд тела цикла известно до начала выполнения команды цикла. Общий вид этой команды таков:

```
for <имя переменной> := <выражение 1 > to <выражение 2> do  
begin  
    <команды тела цикла>  
end;
```

Строка `for ... to ... do` (англ. *for* - для, *to* - до, *do* - делать, выполнять) называется строкой заголовка команды цикла со счетчиком. Переменная в строке заголовка команды цикла со счетчиком, стоящая перед знаком присваивания, называется счетчиком цикла.

Счетчик цикла, выражение 1 и выражение 2 должны принимать только целые значения. Если тело цикла состоит только из одной команды, операторные скобки `begin` и `end` можно не ставить.

Выполняется команда цикла со счетчиком так:

1. Присвоить счетчику цикла (например, переменной `i`) значение выражение `1`.



2. Вычислить значение логического выражения  $i < \langle \text{выражение 2} \rangle$ .
3. Если значение логического выражения, вычисленного в команде 2, true, то выполнить команды тела цикла и перейти к выполнению команды 4, иначе выполнить команду, следующую за командой цикла.
4. Увеличить значение счетчика цикла на 1.
5. Перейти к выполнению команды 2.

Блок-схема выполнения команды цикла со счетчиком представлена на рисунке 2.49.

Для иллюстрации выполнения команды цикла со счетчиком рассмотрим фрагмент программы, в котором вычисляется сумма  $1! + 2! + 3! + 4!$  (напоминаем, что  $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$ ):

```

var a, s, i: Integer;
begin
  s := 0; a := 1;
  for i := 1 to 4 do
  begin
    a := a*i;
    s := s + a;
  end;
  Labell.Caption := IntToStr(s);
end;
    
```

Выполним этот фрагмент программы.

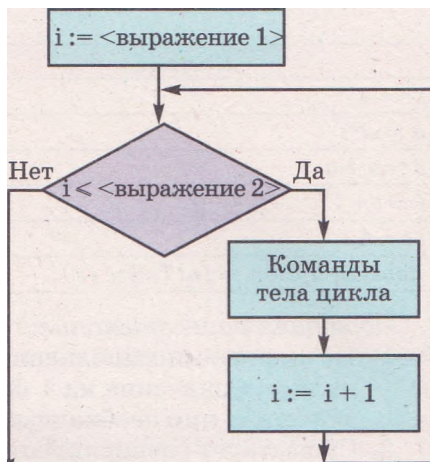


Рис. 2.49. Блок-схема выполнения команды цикла со счетчиком

Команда	Результат выполнения
$s := 0$	$s = 0$
$a := 1$	$a = 1$
$i := 1$	$i = 1$
$i \leq 4$	$(1 \leq 4) = true$
$a := a*i$	$a = 1*1 = 1$
$s := s + a$	$s = 0 + 1 = 1$
$i := i + 1$	$i = 2$
$i \leq 4$	$(2 \leq 4) = true$
$a := a*i$	$a = 1*2 = 2$
$s := s + a$	$s = 1 + 2 = 3$
$i := i + 1$	$i = 3$
$i \leq 4$	$(3 \leq 4) = true$
$a := a*i$	$a = 2*3 = 6$
$s := s + a$	$s = 3 + 6 = 9$
$i := i + 1$	$i = 4$

Команда	Результат выполнения
$i \leq 4$	$(4 \leq 4) = true$
$a := a * i$	$a = 6 * 4 = 24$
$s := s + a$	$s = 9 + 24 = 33$
$i := i + 1$	$i = 5$
$i \leq 4$	$(5 \leq 4) = false$
$Label1.Caption := IntToStr (s)$	$Label1.Caption = 33$

Обращаем ваше внимание.

1. По окончании выполнения команды цикла со счетчиком счетчик цикла имеет значение на 1 больше, чем значение <выражение 2>, и это значение, при необходимости, можно использовать дальше.
2. Существует разновидность команды цикла со счетчиком, который изменяется в обратном порядке:

```
for <имя переменной> := <выражение 1 > downto <выражение 2> do
begin
  <команды тела цикла>
end;
```

(англ. *down* - вниз), отличающаяся от предыдущей только тем, что после каждого выполнения команд тела цикла значение счетчика цикла не увеличивается на 1, а уменьшается на 1.

### Команды цикла с предусловием и цикла с постусловием в Delphi

Если количество повторений команд тела цикла до начала выполнения команды цикла неизвестно, нужно использовать команду цикла с предусловием или команду цикла с постусловием.

Общий вид команды цикла с предусловием таков:

```
while <Логическое выражение> do
begin
  <команды тела цикла>
end;
```

(англ. *while* - пока). Если тело цикла состоит только из одной команды, операторные скобки `begin` и `end` можно не ставить.

Выполняется команда цикла с предусловием так:

1. Вычислить значение логического выражения.
2. Если это значение `true`, то выполнить команды тела цикла и перейти к команде 1, а если `false`, то выполнить команду, следующую за командой цикла.

Для иллюстрации выполнения команды цикла с предусловием рассмотрим фрагмент программы для решения следующей задачи.

**Задача 2.** Вычислить сумму положительных членов арифметической прогрессии с положительным первым членом и отрицательной разностью.

Договоримся вводить первый член прогрессии в поле `Edit1`, а ее разность — в поле `Edit2`.

```

var a, d, s: Real;
begin
  a := StrToFloat (Edit1.Text);
  d := StrToFloat (Edit2.Text);
  s := 0;
  while a > 0 do
    begin
      s := s + a;
      a := a + d;
    end;
  Label1.Caption := FloatToStr (s);
end;

```

Выполним этот фрагмент программы для некоторого набора значений *a* и *d*.

Команда	Результат выполнения
<i>a</i> := StrToFloat (Edit1.Text)	<i>a</i> = 7
<i>d</i> := StrToFloat (Edit2.Text)	<i>d</i> = -3
<i>s</i> := 0	<i>s</i> = 0
<i>a</i> > 0	(7 > 0) = true
<i>s</i> := <i>s</i> + <i>a</i>	<i>s</i> = 0 + 7 = 7
<i>a</i> := <i>a</i> + <i>d</i>	<i>a</i> = 7 + (-3) = 4
<i>a</i> > 0	(4 > 0) = true
<i>s</i> := <i>s</i> + <i>a</i>	<i>s</i> = 7 + 4 = 11
<i>a</i> := <i>a</i> + <i>d</i>	<i>a</i> = 4 + (-3) = 1
<i>a</i> > 0	(1 > 0) = true
<i>s</i> := <i>s</i> + <i>a</i>	<i>s</i> = 11 + 1 = 12
<i>a</i> := <i>a</i> + <i>d</i>	<i>a</i> = 1 + (-3) = -2
<i>a</i> > 0	(-2 > 0) = false
Label1.Caption := FloatToStr ( <i>s</i> )	Label1.Caption = 12

Общий вид команды цикла с постусловием таков:

```

repeat
  <команды тела цикла>
until <логическое выражение>;

```

(англ. *repeat* - повторить, *until* - пока, не раньше чем).

Выполняется команда цикла с постусловием так:

1. Выполнить команды тела цикла.
2. Вычислить значение логического выражения.
3. Если это значение false, то выполнить команду 1, а если true, то выполнить команду, следующую за командой цикла.

Команду цикла с постусловием целесообразно использовать в тех случаях, когда команды тела цикла должны выполняться хотя бы один раз.

Проиллюстрируем, как будет выглядеть фрагмент программы для решения предыдущей задачи с использованием цикла с постусловием:

```

vara, d, s: Real;
begin
  a := StrToFloat (Edit1.Text);
  d := StrToFloat (Edit2.Text);

```

```

s := 0;
repeat
  s := s + a;
  a := a + d;
until a <= 0;
Label1.Caption := FloatToStr (s);
end;

```

Выполним этот фрагмент программы для некоторого набора значений  $a$  и  $d$ .

Команда	Результат выполнения
$a := \text{StrToFloat}(\text{Edit1.Text})$	$a = 7$
$d := \text{StrToFloat}(\text{Edit2.Text})$	$d = -3$
$s := 0$	$s = 0$
$s := s + a$	$s = 0 + 7 = 7$
$a := a + d$	$a = 7 + (-3) = 4$
$a \leq 0$	$(4 \leq 0) = \text{false}$
$s := s + a$	$s = 7 + 4 = 11$
$a := a + d$	$a = 4 + (-3) = 1$
$a \leq 0$	$(1 \leq 0) = \text{false}$
$s := s + a$	$s = 11 + 1 = 12$
$a := a + d$	$a = 1 + (-3) = -2$
$a \leq 0$	$(-2 \leq 0) = \text{true}$
$\text{Label1.Caption} := \text{FloatToStr}(s)$	$\text{Label1.Caption} = 12$

Обращаем ваше внимание:

1. В команде цикла с постусловием операторные скобки не используют независимо от количества команд в теле цикла.
2. Если цикл с предусловием заменить на цикл с постусловием или наоборот, то логическое выражение одного является отрицанием логического выражения другого.

## Примеры программ с использованием циклов

Рассмотрим несколько интересных и полезных для последующего изучения темы задач, в алгоритмах решения которых используются циклы.

При решении многих задач целесообразно использовать еще две арифметические операции: нахождение неполного частного и остатка от деления целого числа на натуральное. Напомним, что для любого целого числа  $m$  и натурального числа  $n$  существует единственная пара целых чисел  $q$  и  $r$  ( $0 \leq r < n$ ), таких что  $m = nq + r$ . Число  $q$  называется неполным частным, а число  $r$  - остатком. Для нахождения неполного частного в Delphi используется операция `div` (англ. *divide* - разделить), а для нахождения остатка - `mod` (англ. *modulo* - остаток от деления). Например,

$23 \text{ div } 5 = 4, 28 \text{ div } 4 = 7, 2 \text{ div } 3 = 0,$   
 $23 \text{ mod } 5 = 3, 28 \text{ mod } 4 = 0, 2 \text{ mod } 3 = 2.$

Задача 3. Дано натуральное число  $n$ , которое больше 1. Выяснить, является ли это число простым.

Напомним, что простым называется натуральное число, которое имеет ровно два делителя. Поэтому можно перебрать все натуральные числа от 1



до данного числа и подсчитать количество делителей данного числа. Если это количество равно двум, то данное число простое, если больше - не простое. Соответствующий фрагмент программы будет выглядеть так:

```

var i, k, n: Integer;
begin
  n := StrToInt (Edit1 .Text);
  k := 0; // Количество делителей числа n
  for i := 1 to n do
    if n mod i = 0 // Проверка, является ли число i делителем числа n
    then k := k + 1; {Увеличение на 1 количества делителей числа n, если число i является его делителем}
    if k = 2
    then Label1 .Caption := 'простое'
    else Label1 .Caption := 'не простое';
end;
```

Но время выполнения программы для решения этой задачи можно существенно уменьшить, если учесть такие свойства натуральных чисел:

1. Любое натуральное число, большее 1, всегда имеет два делителя (единицу и само это число). Поэтому простым будет такое натуральное число, которое не будет иметь других делителей.
2. Среди натуральных чисел только одно четное число является простым (2), все остальные простые числа - нечетные.
3. Если не учитывать само число, то у натурального числа нет делителей, которые превышают арифметический квадратный корень из этого числа.

Если использовать указанные свойства, то соответствующий фрагмент программы может быть таким:

```

var i, k, n: Integer; f: Boolean;
begin
  n := StrToInt (Edit1 .Text);
  f := true; {Будем пока считать число n простым, ведь делителей у него пока еще не нашлось}
  if (n mod 2 = 0) and (n > 2)
  then f := false // Если число n четное и не равно 2, то оно не простое
  else
    begin
      k := 3; {Если число нечетное, то будем искать его делители, начиная с числа 3}
      while (k <= sqrt (n)) and f do {Искать делители числа будем среди чисел, которые не превышают арифметический квадратный корень из числа n, и пока такой делитель не нашется}
        if n mod k = 0 // Проверка, является ли число k делителем числа n
        then f := false
        else k := k + 2; {Если k не является делителем n, то следующий возможный делитель - следующее нечетное число}
      end;
    iff
      then Label1 .Caption := 'простое'
      else Label1 .Caption := 'не простое';
    end;
```

В приведенном фрагменте программы использована логическая переменная f. Ее значение будет определять, является число n простым или

нет: true - простое, false - не простое. Тип логической переменной в Delphi обозначается Boolean, в честь Джорджа Буля. Для вычисления арифметического квадратного корня использована стандартная функция sqrt (англ. *square root* - квадратный корень).

**Задача 4.** Найти наибольший общий делитель (НОД) двух данных натуральных чисел  $a$  и  $b$  ( $a > b$ ).

В курсе математики 6-го класса вы научились находить НОД чисел, раскладывая их на простые множители. Можно составить программу, в которой реализуется этот метод нахождения НОД.

Но более простой оказывается программа, которая реализует другой метод нахождения НОД, базирующийся на таком математическом утверждении: если  $a > b$ , то  $\text{НОД}(a, b) = \text{НОД}(b, r)$ , где  $r$  — остаток от деления  $a$  на  $b$ . Идея этого метода заключается в том, что последовательно заменяются числа, для которых нужно найти НОД: большее из них заменяется на меньшее, а меньшее — на остаток от деления большего числа на меньшее. Заканчивается этот процесс замены тогда, когда остаток от деления становится равным нулю. Тогда НОД равен последнему отличающемуся от 0 остатку от деления.

Например,

$$\text{НОД}(80, 12) = \text{НОД}(12, 8) = \text{НОД}(8, 4) = \text{НОД}(4, 0) = 4,$$

$$\text{НОД}(125, 54) = \text{НОД}(54, 17) = \text{НОД}(17, 3) = \text{НОД}(3, 2) = \text{НОД}(2, 1) = \text{НОД}(1, 0) = 1.$$

Этот метод нахождения НОД называется алгоритмом Евклида.

Ниже приведен фрагмент программы, в котором находится НОД двух чисел по алгоритму Евклида.

```
var a, b, r: Integer;
begin
  a := StrToInt(Edit1.Text);
  b := StrToInt(Edit2.Text);
  r := a mod b;
  while r <> 0 do
    begin
      a := b;
      b := r;
      r := a mod b;
    end;
  Labell.Caption := IntToStr(b);
end;
```

Обращаем ваше внимание, что приведенный фрагмент программы работает правильно и в тех случаях, когда  $a < b$ . Попробуйте самостоятельно выяснить, почему.



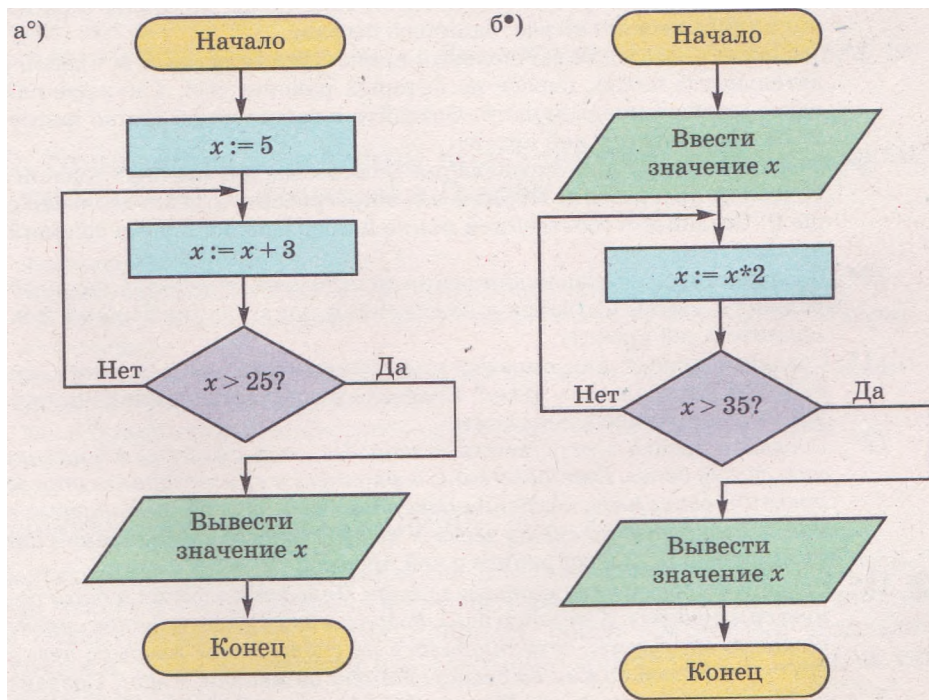
### Проверьте себя

- 1°. Что такое цикл в алгоритме?
- 2\*. Приведите блок-схему цикла с предусловием. Поясните выполнение этого цикла. Приведите примеры циклов с предусловием.
- 3\*. Приведите блок-схему цикла с постусловием. Поясните выполнение этого цикла. Приведите примеры циклов с постусловием.

4. Могут ли команды тела цикла не выполняться ни разу? Поясните свой ответ. Приведите примеры.
- 5\*. Может ли выполнение цикла никогда не закончиться? Поясните свой ответ. Приведите примеры.
6. Поясните отличия в исполнении основных алгоритмических структур: следование, ветвление, цикл.
- 7°. Приведите общий вид команды цикла со счетчиком в Delphi.
8. Поясните выполнение команды цикла со счетчиком в Delphi.
- 9°. Приведите общий вид команды цикла с предусловием и команды цикла с постусловием в Delphi.
10. Поясните выполнение команды цикла с предусловием и команды цикла с постусловием в Delphi.
- 11\*. Поясните отличия в использовании команд цикла с параметром и цикла с предусловием в Delphi.
- 12\*. Поясните отличия в использовании команд цикла с предусловием и цикла с постусловием в Delphi.

 **Выполните задания**

1. Выполните алгоритм:



2. Выполните в таблице фрагмент программы и выясните, какими будут значения переменных после его завершения:

а) `p := 1; a := 2;
for i := 1 to 5 do
begin
a := 3*a + 1;
p := p*a;
end;`

б) `k := 1; a := 12;
while a < 100 do
begin
a := 2*a - 4;
k := k + 1;
end;`

в) `s := 0; a := 1; k := 1;
repeat
s := s + a;
k := k + 1;
a := k*k;
until a > 50;`



3. Выполните фрагмент программы и выясните, какими будут значения переменных после его завершения:

<p>а) <b>p := 1; a := 8;</b>  <b>for i := 1 to 4 do</b>  <b>begin</b>  <b>  a:=2*a-5;</b>  <b>  p:=p*a;</b>  <b>end;</b></p>	<p>б) <b>к:=1;a:=100;</b> в) <b>s := 0; a := 5428;</b>  <b>while a &gt; 10 do</b>  <b>begin</b>  <b>  a := a div 2;</b>  <b>  k:= k + 1;</b>  <b>end;</b></p>	<p><b>repeat</b>  <b>  c := a mod 10;</b>  <b>  s:= s + c;</b>  <b>  a := a div 10;</b>  <b>until a &lt; 1;</b></p>
--	---	---

4#. Выполните в таблице фрагмент программы для задачи 3, приведенной в тексте пункта.



5. Выполните в таблице фрагмент программы для задачи 4, приведенной в тексте пункта.

6°. Составьте блок-схему алгоритма нахождения суммы первых пяти членов последовательности, первый член которой  $x = 7$ , а каждый следующий вычисляется на основе предыдущего значения по формуле  $x := 2*x + 3$ . Исполните алгоритм.



7. Составьте блок-схему алгоритма нахождения количества положительных членов арифметической прогрессии с известным первым членом и разностью -5. Исполните алгоритм для трех разных значений первого члена прогрессии. Подберите эти значения так, чтобы команды тела цикла выполнились несколько раз; один раз; ни разу.

8. Создайте проект для вычисления произведения первых  $N$  членов последовательности чисел, первое из которых равняется  $x$ , а каждое следующее вдвое меньше предыдущего. Создайте в собственной папке папку Проект 2.9.8 и сохраните в ней проект.

9. Создайте проект для определения количества двузначных членов арифметической прогрессии. Первый член прогрессии больше **10**, разность больше 0. Создайте в собственной папке папку Проект 2.9.9 и сохраните в ней проект.

10. Создайте проект для нахождения количества делителей заданного натурального числа. Создайте в собственной папке папку Проект 2.9.10 и сохраните в ней проект.

11\*. Создайте проект для определения количества простых чисел среди первых **100** натуральных чисел. Создайте в собственной папке папку Проект 2.9.11 и сохраните в ней проект.

12. Создайте проект для решения задачи: *Человек положил в банк определенную сумму денег. Ежегодно банк добавляет к сумме заранее определенный процент от суммы, сохранившейся на счету после предыдущего года. Какая сумма будет на счету через  $N$  лет?* Создайте в собственной папке папку Проект 2.9.12 и сохраните в ней проект.

13. Создайте проект для решения задачи: *Человек положил в банк определенную сумму денег. Ежегодно банк добавляет к сумме заранее определенный процент от суммы, сохранившейся на счету после предыдущего года. Через сколько лет сумма на счету станет не меньше  $S$  грн.?* Создайте в папке Мои документы папку Проект 2.9.13 и сохраните в ней проект.

14\*. Создайте проект для вычисления суммы цифр заданного натурального числа, количество цифр которого не больше 9. Создайте в собственной папке папку Проект 2.9.14 и сохраните в ней проект.

## Практическая работа № 9. Программирование циклических вычислений

Внимание! Во время работы с компьютером соблюдайте правила безопасности и санитарно-гигиенические нормы.

1. Откройте среду визуального проектирования Turbo Delphi 2006.



2. Создайте проект для решения задачи: *Перед началом наводнения уровень воды в реке составлял  $N$  м. Во время наводнения каждый час уровень воды повышался на  $P$  % от уровня предыдущего часа. Каким будет уровень воды через  $N$  часов после начала наводнения? Через сколько часов после начала наводнения уровень воды будет не меньше, чем  $K$  м?*
- Разместите на форме поля для ввода начальных данных, надписи с текстами, которые будут объяснять их назначения, и три кнопки.
  - Установите на первой кнопке надпись *Вопрос 1*, на второй - *Вопрос 2*, на третьей - *Очистить*, в полях - *пустой текст*.
  - Составьте обработчик события `OnClick` первой кнопки, выполнение которого приведет к выводу в окно сообщения ответа на первый вопрос задачи.
  - Выполните составленную процедуру и убедитесь, что результаты ее работы правильны.
  - Составьте обработчик события `OnClick` второй кнопки, выполнение которого приведет к нахождению ответа на второй вопрос задачи и выводу его в отдельную надпись.
  - Выполните составленную процедуру и убедитесь, что результаты ее работы правильны.
  - Составьте обработчик события `OnClick` третьей кнопки, выполнение которого приведет к очистке текста в полях и надписи с ответом на второй вопрос задачи.
3. Создайте в собственной папке папку *Практическая 9* и сохраните в ней проект.

## 2.10. Процедуры и функции пользователя и стандартные процедуры и функции в Delphi



- Что такое процедура?
- Что такое обработчик события? Как создать обработчик события?
- Какой общий вид процедур-обработчиков событий, которые вы создавали в Delphi?
- Какие переменные называются локальными, а какие - глобальными? Объясните отличия между ними.
- Как в Delphi получить текстовое представление значения числовой переменной и наоборот?

### Процедуры пользователя в Delphi

В предыдущих пунктах вы создавали и использовали процедуры, которые были связаны с определенными событиями и были методами определенных объектов. Заготовки этих процедур среда разработки создавала автоматически, а вы дополняли эти заготовки командами и строкой `var` объявления переменных.

При создании проектов можно включать в них процедуры, не связанные с событиями. Такие процедуры называются процедурами пользователя. Их чаще всего используют в случаях, когда нужно несколько раз при разных значениях переменных выполнять одни и те же операции: вычислять площадь и периметр треугольников или других геометрических фигур, выбирать большее из двух чисел, определять, простое число или нет и др. Или когда задачу можно разделить на отдельные подзадачи, каждая из которых решается по отдельному алгоритму, то есть при использовании метода пошаговой детализации.

## Глава 2

Процедура пользователя, как и другие процедуры - это отдельная часть программы, которая имеет свое имя и может быть вызвана для выполнения в любой другой части программы.

Процедуры пользователя могут быть:

без аргументов и без результатов;

с аргументами и без результатов;

без аргументов и с результатами;

с аргументами и с результатами.



*Аргументы процедуры* — это переменные, которые получают свои значения в команде вызова процедуры перед началом выполнения команд процедуры.

*Результаты процедуры* — это переменные, которые передают свои значения из процедуры определенным переменным в команде вызова процедуры.

Чаще всего используют процедуры последнего типа. Общий вид этих процедур в Delphi таков:

```
procedure <имя> (<списки аргументов с их типами>; var <списки результатов с их типами>);  
var (<списки локальных переменных процедуры с их типами>);  
begin  
  <команды>  
end;
```

Общий вид команды вызова процедуры с аргументами и результатами таков:

```
<имя> (<списки выражений, значения которых передаются аргументам процедуры>, <списки переменных, которым передают свои значения результаты процедуры>);
```

*Обращаем ваше внимание:* в команде вызова процедуры порядок выражений, значения которых передаются аргументам процедуры, и переменных, которым передают свои значения результаты процедуры, должен строго отвечать порядку записи аргументов и результатов процедуры в строке ее заголовка.

Оформим как процедуру фрагмент программы, который вычисляет НОД двух натуральных чисел. У этой процедуры два аргумента - два натуральных числа, НОД которых нужно найти, и один результат - их НОД. Эта процедура будет выглядеть так:

```
procedure MCD (a, b: Integer; var c: Integer);  
var r: Integer;  
begin  
  r := a mod b;  
  while r <> 0 do  
    begin  
      a := b;  
      b := r;  
      r := a mod b;  
    end;  
  c := b;  
end;
```

Вызывать на выполнение эту процедуру для получения значения НОД двух конкретных натуральных чисел можно, например, такой командой: `MCD (35, 77, x)`. Выполняться эта команда будет так:

1. Переменные  $a$  и  $b$  (аргументы процедуры) получают значения 35 и 77 соответственно.
2. Выполняются команды процедуры.
3. Переменная  $c$  (результат процедуры) передает свое значение переменной  $x$ , указанной в команде вызова процедуры.

Следовательно, в результате выполнения указанной команды вызова процедуры переменная  $x$  получит значение 7.

В команде вызова этой процедуры в скобках на первых двух местах, соответствующих аргументам процедуры, могут стоять не только константы, но и переменные, которые на момент выполнения команды вызова имеют определенные натуральные значения. То есть при выполнении команды `MCD (k, p, x)` аргументы процедуры  $a$  и  $b$  получают значения переменных  $k$  и  $p$  соответственно. На этих местах могут стоять также арифметические выражения, значения которых будут вычислены в начале выполнения команды вызова процедуры и эти значения будут переданы аргументам процедуры, например `MCD (2*k+3, 3*p-1, x)`.

*Обращаем ваше внимание:*

1. Аргументы и результаты процедуры могут быть одними и теми же переменными, то есть допустима процедура с заголовком `Procedure One (var a, b: Integer)`. У такой процедуры переменные  $a$  и  $b$  являются одновременно и аргументами, и результатами. Команда вызова этой процедуры будет выглядеть так: `One (x, y)`.
2. Процедура может среди своих команд иметь команду вызова другой процедуры.



Переменные, указанные в заголовке процедуры, называются *формальными параметрами процедуры*.

Переменные, указанные в команде вызова процедуры, называются *фактическими параметрами процедуры*.

Проиллюстрируем удобство использования процедур пользователя (в частности, процедуры нахождения НОД двух натуральных чисел) на примере такой задачи.

**Задача 1.** Найти сумму двух обыкновенных дробей.

В приведенном тексте проекта (рис. 2.50) процедура вызывается дважды. Первый раз для нахождения НОД знаменателей дробей. Этот НОД в следующей команде используется для нахождения наименьшего общего знаменателя дробей ( $\text{НОК}(x, y) = x*y / \text{НОД}(x, y)$ ). Второй раз - для нахождения НОД числителя и знаменателя дроби-суммы с целью ее последующего сокращения. Без использования процедуры текст этого проекта увеличился бы, ухудшилась бы его структурированность.

Результат выполнения сложения двух обыкновенных дробей можно увидеть на рисунке 2.51.

```

{$R *.dfm}

procedure MCD (a, b: Integer; Var c: Integer);
Var r: Integer;
Begin
  r := a mod b;
  While r <> 0 do
  Begin
    a := b;
    b := r;
    r := a mod b;
  End;
  c := b;
End;

procedure TForm1.Button1Click(Sender: TObject);
var nom1, nom2, den1, den2, den, nom, k, z: Integer;
begin
  nom1 := StrToInt (Edit1.Text);
  den1 := StrToInt (Edit2.Text);
  nom2 := StrToInt (Edit3.Text);
  den2 := StrToInt (Edit4.Text);
  MCD(den1, den2, z);
  den := den1*den2 div z;
  nom := nom1*den div den1 + nom2*den div den2;
  MCD(den,nom, k);
  If k <> 1
  Then
  Begin
    den := den div k;
    nom := nom div k;
  End;
  Label5.Caption := IntToStr(nom) + '/' + IntToStr(den);
end;
end.

```

Рис. 2.50. Программа сложения двух обыкновенных дробей с использованием процедуры MCD

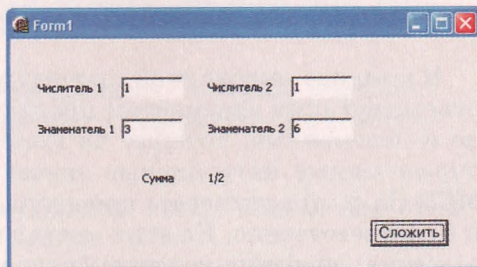


Рис. 2.51. Выполнение программы сложения двух обыкновенных дробей

## Функции пользователя в Delphi

Как упоминалось выше, процедура может не иметь результатов, а может иметь один или более результатов. Если процедура имеет один результат, то удобнее ее оформить и вызвать как функцию пользователя.

Функция пользователя, как и процедура пользователя — это отдельная часть программы, которая имеет свое имя и может быть вызвана для выполнения в любой другой части программы.

Отличается функция от процедуры таким:

1. Строка заголовка функции выглядит так:

**function <имя> (<списки аргументов с их типами>): <тип результата>**

2. Значение, которое функция должна вернуть (передать) вызывающей ее команде, в тексте самой функции должно быть присвоено переменной, имя которой совпадает с именем функции.
3. Функция может быть вызвана либо в правой части команды присваивания, либо в команде вывода.

```

{$R *.dfm}

Function MCD (a, b: Integer): Integer;
Var r: Integer;
Begin
  r := a mod b;
  While r <> 0 do
  Begin
    a := b;
    b := r;
    r := a mod b;
  End;
  MCD := b;
End;

procedure TForm1.Button1Click(Sender: TObject);
Var nom1, nom2, den1, den2, den, nom, k: Integer;
Begin
  nom1 := StrToInt (Edit1.Text);
  den1 := StrToInt (Edit2.Text);
  nom2 := StrToInt (Edit3.Text);
  den2 := StrToInt (Edit4.Text);
  den := den1*den2 div MCD(den1, den2);
  nom := nom1*den div den1 + nom2*den div den2;
  k := MCD(den,nom);
  If k <> 1
  Then
  Begin
    den := den div k;
    nom := nom div k;
  End;
  Label5.Caption := IntToStr(nom) + '/' + IntToStr(den);
End;
end.

```

Рис. 2.52. Программа сложения двух обыкновенных дробей с использованием функции MCD



Если вычисление НОД оформить как функцию, то программа будет выглядеть так, как показано на рисунке 2.52.

Обратите внимание не только на отличия в оформлении процедуры и функции, но и на отличия в их вызове и использовании в программах, приведенных на рисунках 2.50 и 2.52 (табл. 2.4):

Таблица 2.4. Отличия в вызове процедуры и функции

Использование процедуры MCD	Использование функции MCD
MCD (den1, den2, z); den := den1*den2/z;	den := den1*den2/MCD (den1, den2);
MCD (den, nom, k);	k := MCD (den, nom);

Процедуры пользователя и функции пользователя называются подпрограммами пользователя.

### Стандартные процедуры и функции в Delphi

В языке программирования Delphi можно использовать большой набор стандартных процедур и функций, предназначенных для выполнения разнообразных операций над числами, текстами, элементами управления, графическими объектами, файлами и др.

Некоторые из них вы уже использовали, например функцию StrToInt для перевода текста в целое число, функцию FloatToStr для перевода действительного числа в текст, функцию sqrt для вычисления арифметического квадратного корня из действительного числа.

Рассмотрим еще несколько стандартных математических функций и их предназначение (табл. 2.5).

Таблица 2.5. Примеры стандартных математических функций в Delphi

Функция	Количество аргументов и их типы	Результат	Тип результата
Abs	1, целый или действительный	Модуль числа	Тот же, что и у аргумента
Arctan	1, действительный	Арктангенс числа (в радианах)	Действительный
Cos	1, действительный	Косинус числа	Действительный
Exp	1, действительный	Экспонента числа ( $e^x$ )	Действительный
Frac	1, действительный	Дробная часть числа	Действительный
Int	1, действительный	Целая часть числа	Действительный
Ln	1, действительный	Натуральный логарифм числа	Действительный
Pi	0	Число $\pi$	Действительный
Power	2, действительные	I аргумент в степени II аргумент	Действительный
Random	0	Случайное число от 0 до 1	Действительный
Random	1, целый	Случайное целое число от 0 до значение аргумента - 1	Целый
Sin	1, действительный	Синус числа	Действительный
Sqr	1, действительный	Квадрат числа	Действительный

Обращаем ваше внимание:

1. При вызове функции без аргументов нужно указывать только ее имя, например  $x := P_i$ .
2. Перед первым использованием функции **Random** нужно включить в текст проекта команду вызова процедуры **Randomize** без аргументов и без результатов, которая запустит генератор случайных чисел.
3. Для использования функции **Power** нужно в строку **Uses** текста проекта прибавить текст **Math**. После этого к проекту подключится дополнительная библиотека стандартных процедур и функций и станет возможным использование еще целого ряда математических и других процедур и функций, с которыми вы можете ознакомиться в Справке.



## Проверьте себя

- 1°. Что такое процедуры пользователя? Какими они могут быть?
- 2°. В каких случаях целесообразно использовать процедуры пользователя? В чем заключаются преимущества их использования?
- 3°. Каков общий вид процедур пользователя с аргументами и с результатами?
- 4°. Что такое аргументы процедуры; результаты процедуры?
- 5°. Какой вид имеет команда вызова процедуры с аргументами и результатами?
- 6°. Что такое формальные параметры процедуры и фактические параметры процедуры? Объясните отличия между ними.
- 7°. Что такое функция пользователя?
- 8°. Чем функция пользователя отличается от процедуры пользователя?
- 9\*. Каков общий вид функции?
- 10°. Какие вы знаете стандартные процедуры и функции в Delphi? Охарактеризуйте их.
- 11\*. Найдите в Справке информацию о процедурах без аргументов и без результатов; с аргументами и без результатов. Приведите примеры их использования.



## Выполните задания

- 1°. Запишите заголовок процедуры с именем **stepen**, аргументами которой являются переменная **a** действительного типа и переменная **b** целого типа, а результатом - переменная **c** действительного типа, и пример команды ее вызова.
- 2°. Запишите процедуру с именем **simp**, которая выполняет проверку, является ли данное натуральное число простым и возвращает как результат значение переменной **f** логического типа - признак того, является ли число простым, и значение переменной **k** целого типа - количество делителей данного числа.
- 3°. Запишите процедуру с именем **square**, которая вычисляет площадь треугольника по известным длинам его сторон, значения которых передаются процедуре как аргументы. Для вычисления площади используйте формулу Герона.
- 4°. Создайте проект, в котором для двух заданных на плоскости точек с целыми координатами определяется, лежат ли они в одной координатной четверти. Используйте процедуру для определения координатной четверти, в которой размещена точка с заданными координатами. Создайте в собственной папке папку Проект 2.10.4 и сохраните в ней проект.



- 5°. Запишите заголовок функции с именем **length**, аргументами которой являются четыре переменные целого типа, а результат имеет действительный тип, и пример команды ее вызова.
- 6°. Запишите функцию, которая вычисляет длину отрезка по известным координатам его концов. Запишите команду вызова этой функции для вычисления длины отрезка  $a$  с координатами концов (0; 0) и (20; 30).
- 7°. Запишите функцию, в которой вычисляется значение факториала заданного натурального числа  $n$  ( $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$ ). Запишите команду использования этой функции для вычисления суммы факториалов первых пяти натуральных чисел.
- 8°. Создайте проект, в котором можно найти максимальное из четырех заданных разных целых чисел. Используйте функцию для нахождения большего из двух заданных целых чисел. Создайте в собственной папке папку **Проект 2.10.8** и сохраните в ней проект.
- 9°. Создайте проект, в котором для двух заданных целых чисел определяется, какое из них имеет большую сумму цифр. Используйте функцию для нахождения суммы цифр заданного целого числа. Создайте в собственной папке папку **Проект 2.10.9** и сохраните в ней проект.
- 10°. Создайте проект, в котором для двух заданных целых чисел определяется, какое из них имеет большую сумму делителей. Используйте функцию для нахождения суммы делителей заданного целого числа. Создайте в папке **Мои документы** папку **Проект 2.10.10** и сохраните в ней проект.
- 11\*. Создайте проект, в котором для заданного целого числа определяется, является ли одновременно простым само число и число, записанное теми же цифрами, записанными в обратном порядке. Используйте функцию для определения того, является ли число простым. Создайте в собственной папке папку **Проект 2.10.11** и сохраните в ней проект.
- 12°. Создайте проект, в котором вычисляется сумма  $\sqrt{1} + \sqrt{2} + \sqrt{3} + \sqrt{4} + \sqrt{5} + \dots + \sqrt{n}$ . Создайте в собственной папке папку **Проект 2.10.12** и сохраните в ней проект.

## Практическая работа № 10. Использование подпрограмм

**Внимание!** Во время работы, с компьютером соблюдайте правила безопасности и санитарно-гигиенические нормы.

- Откройте среду разработки **Turbo Delphi 2006**.
- Создайте проект, в котором двумя способами вычисляется площадь выпуклого четырехугольника:
  - по известным длинам сторон  $AB$ ,  $BC$ ,  $CD$ ,  $DA$  и длине диагонали  $AC$ . Используйте процедуру вычисления площади треугольника по формуле Герона. Составьте обработчик события **OnClick** первой кнопки, выполнение которого приведет к выводу решения задачи в окно сообщения;
  - по известным координатам вершин четырехугольника, поданным в порядке обхода. Используйте функцию вычисления длины отрезка по координатам его концов. Составьте обработчик события **OnClick** второй кнопки, выполнение которого приведет к выводу решения задачи в отдельную надпись.
- Создайте в собственной папке папку **Практическая 10** и сохраните в ней проект.

## 2.11. Одномерные массивы



1. Что определяет тип переменной? Какие типы переменных вам известны?
2. Какой общий вид команд ветвления в **Delphi**? Как они выполняются?
3. Какой общий вид команды цикла со счетчиком в **Delphi**? Как она выполняется?
4. Какой общий вид команды цикла с предусловием в **Delphi**? Как она выполняется?
5. Где встречаются упорядоченные наборы данных? В чем удобство их использования?

### Понятие одномерного массива

В предыдущих пунктах рассматривали задачи, в которых использовалось небольшое количество переменных. Но существует много задач, в которых нужно обрабатывать значения достаточно большого количества переменных.

Например, если нужно составить программу для нахождения средней массы 100 учеников 11-х классов, то очевидно нужно использовать 100 переменных, значением каждой из которых будет масса соответствующего одиннадцатиклассника, найти сумму значений этих переменных и разделить ее на 100.

Если использовать 100 переменных с произвольными именами, например  $a, b, c, \dots, z, aa, ab, ac, \dots, az, ba, bb, bc, \dots, bz, ca, cb, cc, \dots, cv$  (как в электронных таблицах), то команда нахождения суммы их значений будет содержать в правой своей части 100 слагаемых или нужно включить в текст программы 100 команд, которые по очереди прибавляют к значению переменной *summa* значения массы следующего одиннадцатиклассника:

```
summa := a;  
summa := summa + b;  
summa := summa + c;
```

```
summa := summa + cv;
```

Все это делает текст программы громоздким и неудобным, ввиду того, что количество учеников может быть еще большим или вообще произвольным.

Для решения задач, в которых обрабатываются значения большого количества переменных, можно пойти таким путем: рассматривать все эти переменные как элементы некоторого упорядоченного набора переменных. Такому набору переменных можно присвоить имя, а каждая отдельная переменная набора будет определяться именем набора и своим порядковым номером в этом наборе.



Упорядоченный набор переменных одного типа называется *массивом*. Каждая переменная, входящая в массив, называется *элементом массива*.

Массив, в котором каждый элемент определяется только одним порядковым номером, называется *одномерным*.

Имя каждого элемента одномерного массива состоит из имени массива и взятого в квадратные скобки порядкового номера этого элемента в массиве.



Так, для примера, рассмотренного выше, вместо 100 отдельных переменных стоит рассматривать массив, состоящий из 100 элементов. Если привоить массиву имя  $a$ , то имена переменных - элементов этого массива будут такие:  $a[1]$ ,  $a[2]$ ,  $a[3]$ , ...,  $a[100]$ . Каждая из этих переменных будет иметь тип Real.

Тогда, чтобы найти сумму значений всех этих 100 переменных, достаточно записать такие команды:

```
summa := 0;
for i := 1 to 100 do
summa := summa + a[i];
```

Обращаем ваше внимание, что номер элемента массива может задаваться как числом, так и значением выражения, которое имеет целочисленное значение.

Выполнение этого фрагмента программы будет происходить так:

Команда	Результат выполнения
$summa := 0$	$summa = 0$
$i := 1$	$i = 1$
$i \leq 100$	$(1 \leq 100) = true$
$summa := summa + a[i]$	$summa = 0 + a[1] = a[1]$
$i := i + 1$	$i = 2$
$i \leq 100$	$(2 \leq 100) = true$
$summa := summa + a[i]$	$summa = a[1] + a[2]$
$i := i + 1$	$i = 3$
$i \leq 100$	$(3 \leq 100) = true$
$summa := summa + a[i]$	$summa = a[1] + a[2] + a[3]$
...	...
$i := i + 1$	$i = 100$
$i \leq 100$	$(100 \leq 100) = true$
$summa := summa + a[i]$	$summa = a[1] + a[2] + a[3] + a[4] + \dots + a[100]$
$i := i + 1$	$i = 101$
$i \leq 100$	$(101 \leq 100) = false$

В результате значение переменной  $summa$  будет равно сумме значений всех 100 элементов одномерного массива.

В тексте программы на языке Delphi в строке var переменная  $a$  типа одномерный массив объявляется так: var a: array [1..100] of Real (англ. array - массив). В квадратных скобках указывается диапазон номеров элементов массива: от 1 до 100. По этой записи компилятор выделяет в памяти компьютера 100 участков под каждую из 100 переменных - элементов массива. Размер каждого участка зависит от типа элементов массива.

Обращаем ваше внимание, что иногда удобно нумеровать элементы массива, начиная не с первого номера. Так, например, для переменных, значения которых - количество населения Украины в период с 1990 по 2011 год можно использовать массив a: array [1990..2011] of Integer. Вообще, номера элементов одномерного массива могут быть элементами произвольного диапазона целых чисел.

## Ввод и вывод значений элементов одномерного массива

Для ввода и вывода значений элементов одномерного массива можно воспользоваться многострочным полем (компонент типа **TMemo**), который можно разместить на форме так же, как и другие компоненты (рис. 2.53).

Среди свойств этого компонента есть много таких, которые имели рассмотренные нами ранее компоненты: **Name**, **Color**, **Font**, **Enabled**, **Visible**, **Height**, **Width**, **Top**, **Left** и др. Среди свойств, которых не имели предыдущие компоненты, рассмотрим следующие:

- **ScrollBars** (англ. *scroll* - прокручивание, *bars* - прямоугольные полосы) - значение этого свойства определяет наличие или отсутствие на этом компоненте полос прокрутки (горизонтальной - *ssHorizontal*, вертикальной - *ssVertical*, обеих - *ssBoth*) для просмотра текста, выходящего за пределы компонента;
- **ReadOnly** (англ. *read* - читать, *only* - только) - значение этого свойства определяет возможность редактирования строк компонента: *False* - можно редактировать, *True* - нельзя (только для чтения);
- **Lines** (англ. *lines* - линии, строки) - значением этого свойства является набор значений, расположенных в строках этого компонента; каждая строка имеет номер, нумерация строк начинается с 0;
- **Lines.Count** (англ. *count* - подсчет) - значение этого свойства равно количеству заполненных данными строк компонента; отметим, что это свойство отсутствует в списке свойств окна **Object Inspector**.

Рассмотрим задачу, в которой проиллюстрируем, как ввести значения элементов одномерного массива из многострочного поля и как вывести значения элементов одномерного массива в многострочное поле.

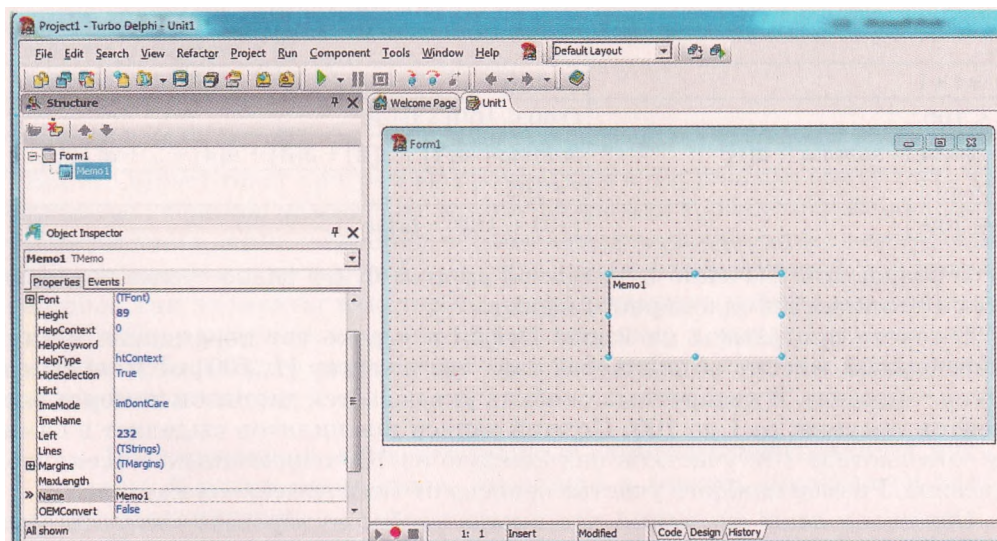


Рис. 2.53. Многострочное поле (компонент типа **TMemo**) и его свойства

**Задача 1.** Сформировать одномерный массив из десяти действительных чисел, которые вводятся в строки многострочного поля, вывести

эти числа в квадрат и вывести эти квадраты в другое многострочное поле.

Разместим на форме два многострочных поля с именами *Mem01* и *Mem02*, расположим их на форме рядом друг с другом и установим для них значения свойств: *Height = 150*, *ScrollBars = ssVertical*, *ReadOnly = False*. Разместим на форме также кнопку, установим для нее *Caption = 'В квадрат'* и создадим процедуру, связанную с событием *OnClick* для этой кнопки.

На рисунке 2.54 представлены текст этой процедуры и окно ее выполнения. В тексте процедуры использованы два стандартных метода, связанных с многострочным полем: процедура *Clear* (англ. *clear* - очищать), очищающая все строки этого компонента, и процедура *Append* (англ. *append* - добавлять), добавляющая к многострочному полю новую строку и выводящая в него значение параметра, указанного в скобках.

*Обращаем ваше внимание:* в приведенном на рисунке 2.54 примере процедуры решения задачи 1 ввод значений элементов, возведение их в квадрат и вывод новых значений элементов массива организованы тремя отдельными циклами только для лучшего понимания. Очевидно, что все эти операции можно было бы организовать в одном цикле.

После запуска проекта на исполнение в открывшемся окне в первом многострочном поле (на рис. 2.54 слева) нужно удалить текст *Mem01*, находящийся там по умолчанию, и ввести в каждую строку по одному числу, которые и будут значениями соответствующих элементов массива. После этого нужно выбрать кнопку, и в результате выполнения процедуры из чисел в строках первого многострочного поля будет сформирован одномерный массив, потом значения элементов этого, массива будут возведены в квадрат, и новые значения будут выведены в строки второго многострочного поля.

Можно ввести числа в многострочное поле до запуска проекта, изменяя значение свойства *Lines* этого поля. Это удобно делать, например, на этапе отладки программы.

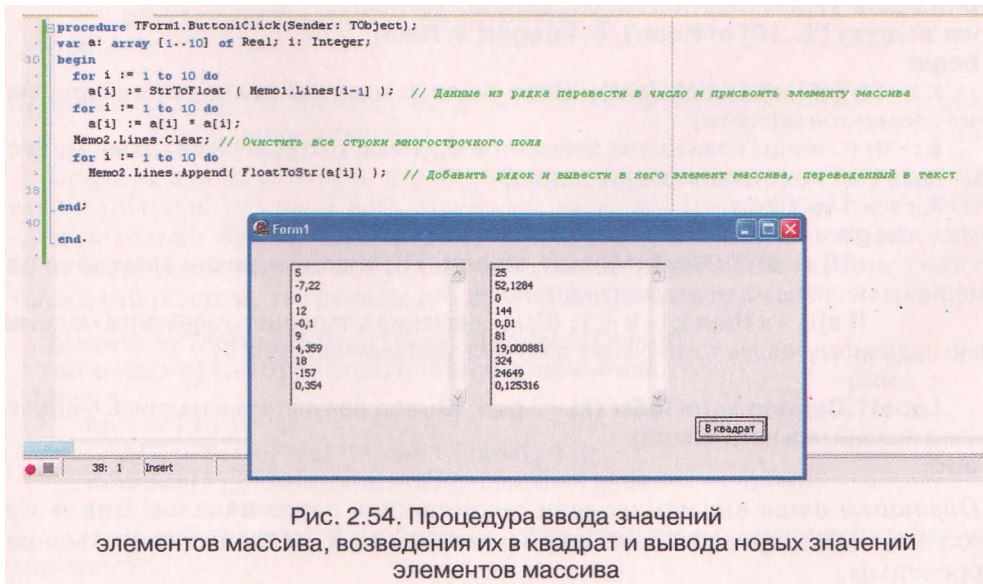


Рис. 2.54. Процедура ввода значений элементов массива, возведения их в квадрат и вывода новых значений элементов массива

Для этого нужно:

1. Выбрать многострочное поле.
2. Выбрать кнопку в конце строки Lines окна Object Inspector.
3. В открывшемся окне String List Editor (англ. *string* - строка текста, *list* - список, *editor* - редактор) удалить текст первой строки и ввести данные десять чисел, нажимая после каждого клавишу Enter.
4. Выбрать кнопку ОК.

Тогда при сохранении проекта эти значения также будут сохранены и автоматически появятся в этом поле после запуска проекта на исполнение.

## Некоторые задачи обработки одномерных массивов

Рассмотрим несколько задач на обработку значений элементов одномерного массива. Во всех задачах, рассмотренных ниже, будем вводить значения элементов массива из многострочного поля, а выводить результат в многострочное поле, если результатом являются значения элементов одномерного массива, или в надпись; будем создавать процедуры, связанные с событием OnClick для кнопки. При необходимости вы сможете аналогично создать процедуры, связанные с другими событиями или с другими объектами.

Будем рассматривать все массивы из десяти чисел. Если количество элементов массива другое, то нужно будет внести в тексты процедур соответствующие изменения.

**Задача 2.** Определить, сколько раз встречается данное действительное число среди значений элементов данного массива из десяти действительных чисел.

Очевидно, что для решения этой задачи нужно просмотреть последовательно значения всех элементов массива, сравнить каждое из них с данным числом, и если некоторое из них равно этому числу, то увеличить счетчик таких элементов массива на 1.

```

procedure TForm1.Button1 Click (Sender: TObject);
vara: array [1..10] of Real; i, k: Integer; x: Real;
begin
    x := StrToFloat (Edit1.Text); {Ввод числа, с которым нужно сравнивать значения элементов массива}
    k := 0; {Счетчик количества элементов массива, которые равны данному числу x - пока еще не встретилось ни одного}
    for i := 1 to 10 do
        begin
            a[i] := StrToFloat (Memol.Lines[i-1]); {Ввод значений элементов одномерного массива из многострочного поля}
            if a[i] = x then k := k + 1; {Если значение очередного элемента массива равно заданному числу, то значение счетчика увеличивается на 1}
        end;
    Label1.Caption := IntToStr (k) + ' раз' {Вывод результата в надпись с добавлением пояснительного текста}
end;

```

*Обращаем ваше внимание:* если не включить перед началом цикла команду  $k := 0$ , то первое выполнение команды  $k := k + 1$  может оказаться некорректным.



**Задача 3.** Определить, встречается ли данное действительное число среди значений элементов данного массива из десяти действительных чисел.

Можно было бы решать эту задачу аналогично предыдущей: определить, сколько элементов массива равны данному числу, и если это количество 0, то данного числа в массиве нет, а если больше 0, то есть. Но такой метод решения является нерациональным. Представьте себе, что данное число является значением уже первого элемента массива. Тогда все остальные элементы массива просматривать не нужно, а предыдущий метод решения предусматривает обязательный просмотр всех элементов массива.

Для реализации рационального метода решения этой задачи используем переменную логического типа *f* и присвоим ей начальное значение *false* (данное число в массиве пока еще не встретилось). Будем сравнивать последовательно значения элементов массива с данным числом и если один из них равен данному числу, то изменим значение переменной *f* на *true* и прервем выполнение цикла командой *break* (англ. *break* - прерывать). Вывод результата реализуем в зависимости от значения переменной *f*.

```

procedure TForm1.Button1Click (Sender: TObject);
var a: array [1 ..10] of Real; i: Integer; x: Real; f: Boolean;
begin
    for i := 1 to 10 do
        a[i] := StrToFloat (Memol.Lines[i-1]);
        x := StrToFloat (Edit1.Text);
        f := false; // Данное число в массиве пока еще не встречалось
    for i := 1 to 10 do
        if a[i] = x
        then begin
            f :=true; //Данное число в массиве встретилось
            break; {Прерываем выполнение цикла, потому что число встре-
тилось в массиве}
        end;
    iff
        then Label1.Caption := 'встречается'
        else Label1.Caption := 'не встречается';
end;

```

**Задача 4.** В данном массиве из десяти действительных чисел определить наибольшее значение.

Сначала будем считать, что значение первого элемента массива и является наибольшим среди всех значений элементов массива. После чего последовательно будем просматривать все остальные элементы массива, и если встретится значение больше того, которое мы на данный момент считаем наибольшим, то сделаем это значение наибольшим.

```

procedure TForm1.Button1Click (Sender: TObject);
var a: array [1 ..10] of Real; i: Integer; max: Real;
begin
    for i := 1 to 10 do
        a[i] := StrToFloat (Memol.Lines[i-1]);
        max :=a[1]; //Считаем, что наибольший элемент - первый
    for i := 2 to 10 do
        if a[i] > max then max := a[i]; {Заменяем значение наибольшего
элемента}

```

```

Label1.Caption := FloatToStr (max) + ' - наибольшее значение';
end;

```

Задача 5. В данном массиве из десяти разных действительных чисел определить наибольшее и наименьшее значения и поменять их местами.

Поскольку нужно поменять местами наибольшее и наименьшее значения среди значений элементов массива, то нужно определить не только сами эти значения, а еще и порядковый номер элемента массива, значение которого наибольшее, и порядковый номер элемента массива, значение которого наименьшее. Попробуем определить все это, просматривая элементы массива только один раз.

Сначала будем считать, что значение первого элемента массива - и наибольшее, и наименьшее, а номер наибольшего и наименьшего элемента массива 1. После чего последовательно будем просматривать все остальные элементы массива. Если значение очередного элемента массива больше, того значения, которое мы на тот момент считаем наибольшим, то заменяем наибольшее значение среди уже просмотренных элементов массива и номер наибольшего элемента. Если нет, то может быть это значение меньше, чем то, которое мы пока что считаем наименьшим. И если это так, то заменяем наименьшее значение среди уже просмотренных элементов массива и номер наименьшего элемента.

```

procedure TForm1.Button1Click (Sender: TObject);
var a: array [1..10] of Real; i, nmax, nmin: Integer; max, min: Real;
begin
  for i := 1 to 10 do
    a[i] := StrToFloat (Memol.Lines[i-1]);
    max := a[1]; nmax := 1; {Значение первого элемента принимается за наибольшее}
    min := a[1]; nmin := 1; {Значение первого элемента принимается за наименьшее}
    for i := 2 to 10 do
      if a[i] > max {Сравнение значения очередного элемента с наибольшим на этот момент}
      then
        begin
          max := a[i]; // Заменяем значение наибольшего элемента
          nmax := i; // Заменяем номер наибольшего элемента
        end
      else
        if a[i] < min {Сравнение значения очередного элемента с наименьшим на этот момент}
        then
          begin
            min := a[i]; // Заменяем значение наименьшего элемента
            nmin := i; // Заменяем номер наименьшего элемента
          end;
          a[nmax] := min; a[nmin] := max; {Обмен значениями наибольшего и наименьшего элементов}
        for i := 1 to 10 do
          Memol.Lines.Append (FloatToStr(a[i]); {Вывод значений элементов массива с переставленными наибольшим и наименьшим значениями}
        end;

```

**Задача 6.** Определить элементы данного массива из десяти действительных чисел, значения которых больше данного действительного числа.

Очевидно, что как и в задаче 2, нужно просмотреть последовательно значения всех элементов массива и выяснить, какие из них больше, чем заданное число. Возникает вопрос, где хранить значения этих элементов массива. Ведь в зависимости от значений элементов массива и от данного числа, количество таких элементов может быть разным: от 0 до 10.

Было бы целесообразно заносить значения искомых элементов в другой массив, но нам предварительно неизвестно количество его элементов.

В языке программирования Delphi есть возможность использовать массивы, количество элементов которых предварительно неизвестно и может изменяться в ходе выполнения проекта. Такие массивы называются динамическими. В отличие от них, массивы, количество элементов которых известно предварительно и не изменяется в ходе выполнения проекта, называются статическими.

*Обращаем ваше внимание, что нумерация элементов динамического массива всегда начинается с 0.*

В строке var динамический массив b, элементами которого будут действительные числа, описывается так: var b: array of Real;

```

procedure TForm1.Button1Click (Sender: TObject);
var: array [1 ..10] of Real; b: array of Real; i, k: Integer; x: Real;
begin
    for i := 1 to 10 do
        a[i] := StrToFloat (Memo1.Lines[i-1 ]);
    x := StrToFloat (Edit1.Text);
    k := 0; {Счетчик количества элементов массива b (сначала в массиве b нет
элементов)}
    for i := 1 to 10 do
        if a[i] > x
            then begin
                k := k+1; {Увеличение на 1 значения счетчика количества
элементов массива b}
                SetLength (b, k); {Установка нового размера (количества
элементов) динамического массива b}
                b[k-1] := a[i]; {Занесение соответствующего элемента массива
a в массив b}
            end;
    Memo2.Lines.Clear;
    if k = 0
    then Memo2.Lines[0] := 'Таких чисел не существует'
    else
        for i := 1 to k do
            Memo2.Lines.Append (FloatToStr(b[i-1]));
    end;

```

**SetLength** (англ. *set* - устанавливать, *length* - длина).

Используя динамические массивы, можно создавать программы для произвольного количества чисел, введенных в строки многострочного поля. В таких случаях нужно ввести данные из этого многострочного поля в динамический массив a, например так:

## Глава 2

```
k := Memol.Lines.Count; // Определяется количество заполненных строк поля
SetLength (a, k); {Устанавливается соответствующее количество элементов
динамического массива a и выделяется для его элементов соответствующий участ-
ток памяти}
For i := 0 to k-1 do a[i] := StrToFloat (Memol.Lines[i]); {Ввод данных из строк
поля в массив}
```

Дальше данные из динамического массива можно обрабатывать различными методами.

### Сортировка одномерного массива и поиск данного числа в упорядоченном массиве

Много программ на обработку значений элементов одномерного массива будут работать значительно быстрее, если значения его элементов будут упорядочены по возрастанию или по убыванию. Это касается прежде всего задач поиска тех значений элементов массива, которые удовлетворяют определенным условиям.

Существует более десяти различных методов сортировки одномерного массива. Одни из них выполняются быстрее, другие - медленнее, одни - сложнее по своей логической структуре, другие - проще. Мы рассмотрим один из методов сортировки одномерного массива - метод выбора.

Поясним суть этого метода на примере. Пусть нам нужно упорядочить по возрастанию такой одномерный массив из шести элементов (табл. 2.6, строка 0):

Таблица 2.6. Упорядочение массива методом выбора

№	$\alpha[1]$	$\alpha[2]$	$\alpha[3]$	$\alpha[4]$	$\alpha[5]$	$\alpha[6]$
0	23	15	7	2	6	4
1	2	15	7	23	6	4
2	2	4	7	23	6	15
3	2	4	6	23	7	15
4	2	4	6	7	23	15
5	2	4	6	7	15	23

На первом шаге определим значение наименьшего элемента во всем массиве ( $a[4] = 2$ ) и обменяем его со значением первого элемента. Получаем массив в строке 1, в котором наименьший элемент занял свое место. На втором шаге определим значение наименьшего элемента среди всех элементов массива, кроме первого, ( $a[6] = 4$ ) и обменяем его со значением второго элемента. Получаем массив в строке 2, в котором первые 2 элемента заняли свои места. На третьем шаге определим значение наименьшего элемента среди всех элементов массива, кроме первых двух, ( $a[5] = 6$ ) и обменяем его со значением третьего элемента. Получаем массив в строке 3, в котором первые 3 элемента заняли свои места. Повторив аналогичные действия еще 2 раза, получаем массив, упорядоченный по возрастанию.

*Обращаем ваше внимание:* хотя массив имеет шесть элементов, достаточно 5 раз найти наименьшее значение элементов из еще не упорядоченной части массива и обменять его местами со значением первого из еще не упорядоченной части массива элемента. На последнем шаге не только 5-й,



но и 6-й элемент массива занимают свои места в упорядоченной части массива, и таким образом весь массив становится упорядоченным.

Из пояснений сути этого метода становится понятным смысл его названия: ведь на каждом шагу выбирается наименьший элемент среди еще неупорядоченной части массива, и его значение обменивается со значением элемента, первого среди еще неупорядоченной части массива.

**Процедура, реализующая этот метод, выглядит так:**

```

procedure TForm1.Button2Click (Sender: TObject);
var a: array [1..10] of integer; i, j, min, nmin: integer;
begin
  for i := 1 to 10 do
    a[i] := StrToInt (Memo1.Lines[i-1]);
  for i := 1 to 9 do
    begin
      min := a[i]; nmin := i;
      for j := i+1 to 10 do
        if a[j] < min then
          begin
            min := a[j]; nmin := j;
          end;
      a[nmin] := a[i];
      a[i] := min;
    end;
  Memo2.Lines.Clear;
  for i := 1 to 10 do
    Memo2.Lines.Append (IntToStr (a[i]))
  end;
end;

```

**Продемонстрируем тот факт, что поиск нужного значения среди значений элементов массива (задача 3) будет происходить значительно быстрее, если массив упорядочен.**

Если массив неупорядочен, то задачу можно решить методом, рассмотренным выше. При реализации этого метода, если данного числа в массиве нет, то для выяснения этого факта нужно будет сравнить данное число со значениями всех элементов массива. И если количество элементов массива большое, например 1000, то и число сравнений (а следовательно, и время исполнения проекта) будет соответствующим.

Если же массив упорядочен, то можно выяснить, есть ли данное число в массиве, иным способом, значительно более эффективным. Поясним его на примере. Пусть имеем упорядоченный по возрастанию массив из десяти чисел: 2, 5, 8, 12, 13, 16, 17, 20, 22, 30 и некоторое данное число  $x$ . Сравним это число со значением элемента массива, который находится посередине массива (с числом 13). Если данное число  $x$  равно 13, то оно в массиве есть, если нет, то выясним, больше ли данное число чем 13 ( $x > 13$ ). Если да, то его нужно искать только в правой половине массива, если нет, то только в левой половине массива. Таким образом, область поиска сужается вдвое. На следующем шаге делаем так же: сравниваем данное число со значением элемента массива, который находится посередине той части массива, которая осталась для поиска. Опять либо данное число равно значению этого элемента массива, либо оставляем для поиска или левую, или правую половину оставшейся части массива, то есть область поиска опять сужается вдвое.

Такой метод поиска эффективнее, чем предыдущий, потому что значительно быстрее приводит к результату, особенно для больших  $N$  (максимум за  $\lceil \log_2 N \rceil + 1$  шагов, где  $N$  - количество элементов в массиве, а квадратными скобками здесь обозначена целая часть числа).

Такой метод поиска заданного числа в одномерном массиве называется методом половинного деления (бинарного поиска).

Соответствующая процедура, в которой реализуется этот метод для упорядоченного массива из десяти целых чисел, выглядит так:

```

procedure TForm1.Button1Click(Sender: TObject);
var a: array [1..10] of Integer; i, x, left, right, m: Integer; f: Boolean;
begin
  for i := 1 to 10 do
    a[i] := StrToInt (Memol.Lines[i-1]);
    x := StrToInt (Edit1.Text);
    left := 1; {Начальный номер элемента той части массива, где будет происходить поиск}
    right := 10; {Конечный номер элемента той части массива, где будет происходить поиск}
    f := false; // Заданное число в массиве пока еще не найдено
    while (left <= right) and not f do
      begin
        m := (left + right) div 2; {Номер элемента посередине той части массива, где будет продолжаться поиск}
        if x > a[m]
          then left := m+1 {Изменяется начальный номер элемента той части массива, где будет происходить поиск}
        else if x < a[m]
          then right := m-1 {Изменяется последний номер элемента той части массива, где будет происходить поиск}
        else f := true; // Число в массиве нашлось
      end;
    iff
      then Edit1.Text := 'Число в массиве есть'
      else Edit1.Text := 'Числа в массиве нет';
end;

```

Метод половинного деления использовали в своих логических рассуждениях, классификациях, методах решения задач еще древние греки. Они называли этот метод дихотомией (греч.  $\delta\iota\chi\eta\mu\acute{\alpha}$  - пополам,  $\tau\omicron\mu\eta$  - деление). Может и вы использовали этот метод в игре «Отгадай задуманное число»?



### Проверьте себя

- 1°. Что такое одномерный массив?
- 2°. Из чего состоит имя элемента массива?
- 3°. Что может использоваться как номер элемента массива?
- 4\*. При составлении проектов для решения каких задач удобно использовать массивы?
- 5°. Как объявить переменную типа *одномерный массив* в строке *var*?
- 6°. Каким может быть диапазон номеров элементов массива?
- 7°. Назовите и охарактеризуйте известные вам свойства многострочного поля.
- 8°. Назовите и поясните известные вам стандартные методы многострочного поля.

9. Как ввести числа в многострочное поле до запуска проекта?
10. Опишите разные способы определения, встречается ли заданное число среди значений элементов массива.
11. Опишите последовательность действий для определения наибольшего значения среди значений элементов массива.
- 12°. Какой массив называется динамическим? Как объявляется динамический массив в строке **var**?
13. Поясните, в чем заключаются отличия между статическим и динамическим массивами.
14. Поясните суть метода выбора при сортировке массива.
15. Поясните суть метода половинного деления (бинарного поиска) данного числа в одномерном массиве.
- 16\*. Поясните, в чем преимущество поиска заданного элемента в упорядоченном массиве по сравнению с неупорядоченным. Приведите примеры такого поиска в вашей учебной деятельности.



### Выполните задания

- 1°. Составьте таблицу выполнения фрагмента программы и определите значение переменной **dob** после его завершения для массива, значениями элементов которого являются числа 2; -3,5; 1,2; 0,25; -4:

```
dob:= 1;
for i := 1 to 5 do
dob := dob * a[i];
```



- 2°. Составьте таблицу выполнения фрагмента программы и определите значение переменной **sum** после его завершения для массива, значениями элементов которого являются числа 2; 0; 23; -14:

```
sum := 0;
for i := 1 to 4 do
if a[i] mod 2 = 0 then sum := sum + a[i]*a[i];
```

3. Запишите строку объявления переменных типа одномерный массив:
  - а) массив переменных целого типа с именем **x** и диапазоном номеров от 1 до 50;
  - б) массив переменных действительного типа с именем **mas** и диапазоном номеров от 10 до 35;
  - в) массив переменных целого типа с именем **tab** и диапазоном номеров от -7 до 12.
- 4°. Создайте проект, в котором нужно сформировать массив из десяти целых чисел, находящихся в строках многострочного поля, найти среднее арифметическое значений элементов массива и вывести результат в надпись. Создайте в собственной папке папку **Проект 2.11.4** и сохраните в ней проект.
5. Создайте проект, в котором нужно сформировать массив из двенадцати действительных чисел, находящихся в строках многострочного поля, увеличить значение каждого элемента массива в 3 раза и вывести новые значения в другое многострочное поле. Создайте в собственной папке папку **Проект 2.11.5** и сохраните в ней проект.
6. Создайте проект, в котором нужно сформировать массив из девяти целых чисел, находящихся в строках многострочного поля, определить, встречаются ли среди значений элементов этого массива числа, кратные числу 3, и вывести результат в надпись. Создайте в собственной папке папку **Проект 2.11.6** и сохраните в ней проект.
7. Создайте проект, в котором нужно сформировать массив из десяти действительных чисел, находящихся в строках многострочного поля, определить наименьшее среди значений элементов этого массива и поменять его

местами с первым элементом массива. Полученный массив выведите в другое многострочное поле. Создайте в собственной папке папку **Проект 2.11.7** и сохраните в ней проект.



8. Создайте проект, в котором нужно сформировать массив из девяти действительных чисел, находящихся в строках многострочного поля, определить наибольшее среди значений элементов этого массива и обменять его местами со значением элемента массива, расположенного посередине массива. Полученный массив выведите в другое многострочное поле. Создайте в папке **Мои документы** папку **Проект 2.11.8** и сохраните в ней проект.
- 9\*. Создайте проект, в котором нужно сформировать динамический массив из действительных чисел, которые находятся в строках многострочного поля, определить среднее арифметическое значение элементов этого массива и найти элементы, которые больше среднего арифметического значения. Для хранения результатов используйте динамический массив. Создайте в собственной папке папку **Проект 2.11.9** и сохраните в ней проект.
- 10\*. Создайте проект, в котором нужно сформировать массив из действительных чисел, находящихся в строках многострочного поля, отсортировать его и найти количество разных значений элементов массива. Создайте в собственной папке папку **Проект 2.11.10** и сохраните в ней проект.



### Практическая работа № 11. Обработка одномерных массивов

**Внимание!** Во время работы с компьютером соблюдайте правила безопасности и санитарно-гигиенические нормы.

1. Откройте среду разработки **Turbo Delphi 2006**.
2. Создайте проект для решения задачи: *Известна средняя температура каждого дня недели. Определите среднюю температуру за всю неделю. Выясните, сколько раз в неделю температура была выше нуля. Отсортируйте значение температур по убыванию и выясните, была ли одинаковой температура несколько дней в неделе.* В проекте:
  - а) разместите на форме необходимые элементы управления: многострочное поле, надписи, кнопки;
  - б) составьте обработчик события **OnClick** первой кнопки, выполнение которого приведет к выводу в надпись средней температуры за неделю;
  - в) составьте обработчик события **OnClick** второй кнопки, выполнение которого приведет к определению, сколько раз в неделю температура была выше нуля, и выводу результата в надпись;
  - г) составьте обработчик события **OnClick** третьей кнопки, выполнение которого приведет к сортировке значений температур и определению, есть ли в массиве одинаковые значения, выводу отсортированного массива в многострочное поле, сообщения о наличии одинаковых элементов в надпись;
  - д) выполните составленные процедуры и убедитесь, что результаты их работы правильны.
3. Создайте в собственной папке папку **Практическая 11** и сохраните в ней проект.



# Глава 3

## БАЗЫ ДАННЫХ.

### СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

В этой главе вы узнаете о:

- базах данных, компьютерных базах данных, их классификации и назначении;
- моделях данных, их классификации;
- реляционной модели данных, ее особенностях и основных объектах;
- связях в базах данных, их классификации;
- модели «сущность-связь», правилах ее построения и отображения на базу данных;
- системах управления базами данных, их назначении;
- системе управления базами данных **Access 2007** и возможностях, которые она предоставляет пользователю при работе с базами данных;
- создании, редактировании и форматировании таблиц, форм, запросов и отчетов базы данных в среде СУБД **Access 2007**;
- поиске, сортировке и фильтрации данных средствами СУБД **Access 2007**.

#### 3.1. Базы данных. Модель «сущность-связь»



1. Что такое информационная система? Назовите составляющие информационной системы и их назначение.
2. Что такое модель объекта? Какие разновидности моделей вы знаете?
3. Что такое предметная область? Приведите примеры предметных областей.

#### Понятие о базах данных

Необходимость поиска нужной информации у человека возникает постоянно, независимо от сферы его профессиональных интересов: от какой платформы отправляется поезд на Хмельницкий, как приготовить вареники с вишнями, какое строение имеет молекула воды, сколько дней длилась Вторая мировая война, справедлива ли примета о черной кошке, какая частота переменного электрического тока в бытовой электрической сети, какое окончание имеют существительные третьего склонения в именительном падеже единственного числа и др. Ответы на часть этих вопросов человек может получить из своей памяти, для получения других необходимо обратиться к информационной системе железнодорожного вокзала, просмотреть кулинарную книгу, справочник по химии, физике или электротехнике, пособие по правописанию и т. п. Для облегчения поиска нужной информации человечество придумало много средств - универсальные и предметные энциклопедии, справочники и словари, справочные бюро и информационные таблицы и др.

Объемы накопленных человечеством сообщений непрерывно растут. Так, во время раскопок древнего города шумеров Ур было найдено свыше 20 тысяч глиняных табличек со сведениями об обычаях давнего народа, его легендах и событиях, которые происходили более 5 тысяч лет назад. Знаменитая

Александрийская библиотека, которая была основана в Египте в III в. до нашей эры, по разным источникам, содержала от 100 до 700 тысяч рукописей. Сегодняшние библиотеки поражают объемами разнообразных данных. Наибольшей в мире считается Британская библиотека в Лондоне, которая насчитывает свыше 150 млн единиц хранения, а наибольшая библиотека нашей страны - Национальная библиотека Украины имени В.И. Вернадского в Киеве - насчитывает свыше 15 млн единиц хранения.

Ученые утверждают, что хранение больших объемов данных оправдано только при условии, если поиск нужных данных осуществляется быстро и подаются они в доступной для понимания форме. Эти условия обеспечивают современные технологии хранения данных. Основой этих технологий являются компьютеризованные базы данных (БД).



**База данных** — это упорядоченный по определенным правилам набор взаимосвязанных данных.



Первая в Украине компьютерная база данных была разработана в ходе работ по проектированию и эксплуатации электронной вычислительной машины «Киев» (1959 г.). ЭВМ была разработана для вычислительного центра Академии наук УССР Л.Н. Дашевским, Е.Л. Ющенко, Е.А. Шкарабарой, С.Б. Погребинским под научным руководством Б.В. Гниденко и В.М. Глушкова.

**Виктору Михайловичу Глушкову** (1923-1982)

(рис. 3.1) принадлежит цикл трудов по использованию ЭВМ в управлении не только в пределах одного предприятия или организации, а всей страны. Это он разработал проект **Общegosударственной автоматизированной системы сбора и обработки информации** для учета, планирования и управления экономикой страны. Основу этой системы должны были составлять базы данных, или, как их называл В.М. Глушков, **банки данных**. В 1975 г. он говорил: «Уже в ближайшие десятилетия в ряде высокоразвитых стран будут созданы так называемые *национальные банки данных*». Создание подобных банков данных, как и прогнозировал В.М. Глушков, в корне изменило работу и научного работника, и инженера, и журналиста.



Рис. 3.1. Виктор Михайлович Глушков

### Понятие сущности, модель «сущность—связь»

В начале проектирования базы данных, как правило, разрабатывается модель предметной области, для которой создается эта БД. В ней указываются типы объектов, сведения о которых будут включены в базу данных, и связи между ними. Для наглядности такую модель можно представить в графическом виде.

Рассмотрим пример модели предметной области Информатика в школе (рис. 3.2).



Рис. 3.2. Модель предметной области Информатика в школе

В прямоугольниках записаны названия типов объектов (групп объектов) предметной области, стрелки указывают на связи между ними, а надписи возле стрелок описывают эти связи. Учитель предварительно *работает с Учебником информатики* и организует процесс обучения учеников (*обучает*). Он же оценивает уровень учебных достижений учеников и *выставляет* Оценки. Ученики *учатся* под руководством учителя и самостоятельно, имея Учебник информатики. Они также *принадлежат* к определенному Классу и *получают* Оценки.

Тип объекта предметной области называют сущностью. Сущностями рассмотренной выше предметной области являются: Учитель, Оценка, Ученик, Учебник информатики, Класс. Сущности, по сути, являются совокупностью однотипных объектов. Например, сущность Классы может состоять из объектов 11-А, 11-Б, 11-В и т. п. Отдельные объекты сущности называют экземплярами сущности. Сущность имеет как минимум один экземпляр. Например, сущность Учебник информатики имеет только один экземпляр сущности, если все ученики школы учатся, используя учебник информатики одного авторского коллектива.



*Сущность предметной области* — это тип реального или мнимого объекта предметной области.

В дальнейшем словесное описание связей между сущностями заменяют их условными обозначениями соответствующего типа. За множественностью выделяют такие типы связей:

- один к одному (обозначается как 1:1), когда одному экземпляру одной сущности соответствует один экземпляр другой сущности. Для приведенного примера такая связь может быть установлена между сущностями Учитель и Учебник информатики, если каждый учитель использует только один учебник, который не используют другие учителя;
- один ко многим (обозначается как 1:оо, или 1:М, где М - от англ. Many - много), когда одному экземпляру одной сущности может соответствовать несколько экземпляров другой сущности. Такая связь возможна между сущностями Учитель и Ученик, когда каждый учитель информатики обучает учеников одного или нескольких классов и при этом нет учителей, которые обучают одних и тех же учеников;
- много к одному (обозначается как оо:1, или М:1), когда нескольким экземплярам одной сущности соответствует один экземпляр другой сущности. Этот тип связи является противоположным к связи один ко многим. Например, если рассмотреть связь между сущностью Ученик и сущностью Класс, то тип связи будет *много к одному*, а если рассматривать связь между сущностями Класс и Ученик, то тип связи будет *один ко многим*;
- много ко многим (обозначается как оо:оо, или М:М), когда нескольким экземплярам одной сущности могут соответствовать несколько экземпляров другой сущности. Например, связь между сущностями Ученик и Оценка - многие ученики получают много разных оценок.

Связи между сущностями классифицируют также за *полнотой*. В соответствии с этой классификацией выделяют связи, в которых: каждый экземпляр одной сущности *обязательно связан* с одним или несколькими экземплярами другой сущности. Например, связь между сущностями Класс и Ученик предусматривает, что каждый ученик при-



надлежит к определенному классу и каждый класс состоит из определенной группы учеников;

- каждый экземпляр одной сущности *не обязательно связан* хотя бы с одним экземпляром другой сущности. Например, между сущностями Ученик и Компьютер может быть установлена связь, которая предусматривает, что не каждый ученик имеет собственный компьютер.

Рассмотренную нами модель предметной области называют моделью «сущность—связь», или ER-моделью, или ER-диаграммой (англ. *Entity-Relationship* - сущность (объект)-связь (отношение)). При создании ER-модели используются специальные обозначения типов сущности, свойств экземпляров сущностей, связей и др. Набор таких условных обозначений называют нотацией (англ. *notation* - обозначение, изображение условными знаками). В таблице 3.1 представлены основные элементы одной из нотаций.

Таблица 3.1. Основные элементы графического представления модели «сущность—связь»

Условное обозначение	Название фигуры	Что обозначает
Название	Прямоугольник	Сущность предметной области
Название	Овал	Свойство экземпляров сущности
Название	Ромб	Связь
—————	Сплошная линия	Линия обязательной связи
- - - - -	Штриховая линия	Линия необязательной связи

Построим модель «сущность—связь» для предметной области Кинотеатр, в которой рассмотрим две сущности: Зритель и Билет. Каждый зритель, который пришел в кинотеатр, имеет билет. Двух одинаковых билетов, как и двух одинаковых зрителей, не существует (рис. 3.3).

В сущности Зритель для данной модели рассматриваются два свойства: *Фамилия* и *Имя*; а в сущности Билет - пять: *Дата*, *Время*, *Ряд*, *Место* и *Номер билета*. Между сущностями Зритель и Билет существует связь - *Имеет*. За множественностью это связь один к одному. На это указывают

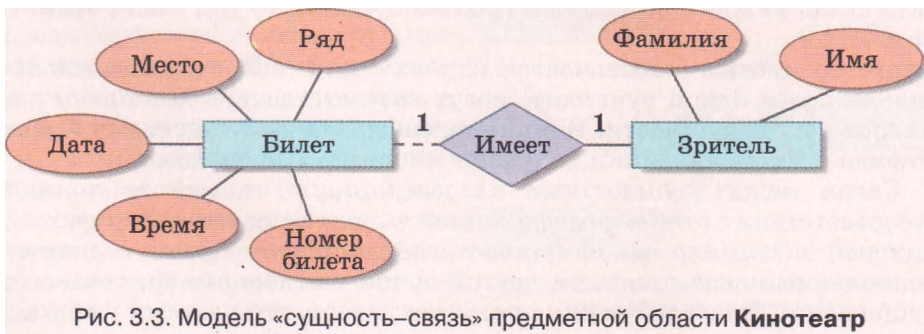


Рис. 3.3. Модель «сущность—связь» предметной области Кинотеатр



обозначения над линиями связи возле прямоугольников сущностей. За полнотой эта связь является обязательной для сущности Зритель (для того чтобы быть зрителем, человек должен иметь билет) и необязательной для сущности Билет (поскольку не все билеты могут быть проданы, тогда они останутся в кассе и не попадут к зрителю).

### Проверьте себя

- 1°. Для чего нужно упорядоченное хранение данных?
- 2°. Какие требования к хранению данных выдвигаются современным информатизированным обществом?
- 3°. Что такое база данных? Где используются базы данных?
- 4°. Объясните, что такое сущность.
- 5°. Что такое экземпляр сущности? Приведите примеры.
- 6°. Назовите типы связей между сущностями предметной области.
- 7°. Опишите особенности связи «один к одному» между сущностями предметной области.
- 8°. Опишите особенности связи «один ко многим» и «много к одному» между сущностями предметной области, приведите примеры таких связей.
- 9°. Опишите особенности связи «много ко многим» между сущностями предметной области.
- 10°. На что указывает тип связи при их классификации за полнотой?
- 11°. Что такое модель «сущность-связь»? Что она описывает?
- 12°. Как называется набор условных обозначений при графическом представлении модели «сущность-связь»?
- 13°. Опишите обозначения, которые используются при графическом представлении модели «сущность-связь».
- 14°. Какова роль В.М. Глушкова в развитии направлений информатики, связанных с использованием баз данных?

### Выполните задания

1. Рассмотрите диаграмму «сущность-связь» (рис. 3.4). Определите:
  - а) количество и название сущностей, изображенных на диаграмме;
  - б) свойства каждой сущности;



Рис. 3.4

в) виды связей для случаев:

- есть несколько видов товара и есть несколько складов, каждый товар хранится на одном складе и на каждом складе хранится только один вид товара;
- есть несколько видов товара и есть несколько складов, на каждом складе есть все виды товаров;
- есть только один вид товара и он хранится на одном складе;
- есть несколько видов товара и все они хранятся на одном складе;
- есть один вид товара и он хранится на нескольких складах.

2. Рассмотрите диаграмму «сущность-связь» (рис. 3.5). Определите:

- а) количество и название сущностей, изображенных на диаграмме;
- б) свойства каждой сущности;
- в) виды связей для случаев:
- есть только один магазин и он продает автомобили одной модели;
  - есть несколько магазинов и все они продают автомобили одной модели;
  - есть несколько магазинов и есть несколько моделей автомобилей, в каждом магазине есть автомобили всех моделей;
  - есть один магазин и он продает автомобили нескольких моделей.

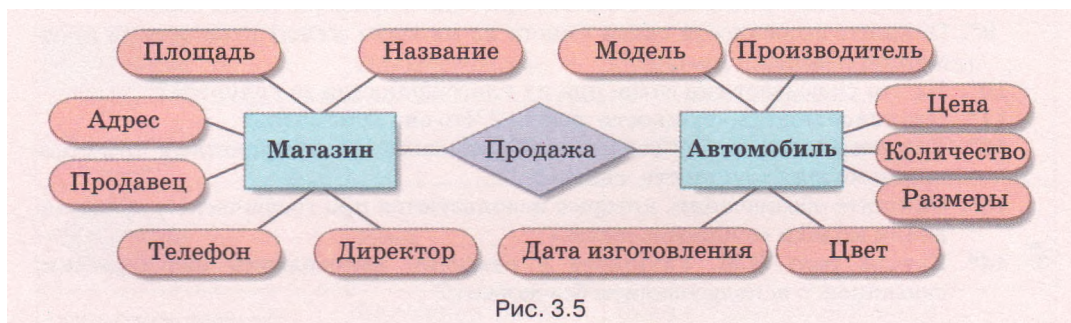


Рис. 3.5

3. Разработайте модель «сущность-связь» для создания таких баз данных:

- а) телефонный справочник учеников класса; справочник должен содержать данные о номерах стационарного и мобильного телефонов ученика, его фамилию, имя, адрес;
- б) домашняя аудиотека, которая должна содержать данные о номере диска в домашней аудиотеке, о названии диска, об исполнителе, о дате выпуска диска, об общей длительности музыкальных произведений на диске;
- в) электронная книга учета поступления товара в магазин, которая должна содержать данные о номере накладной, по которой получен товар, о названии фирмы-поставщика, о наименовании и модели товара, о количестве товара и его цене, о дате поступления в магазин.
- г) электронный классный журнал - перечень необходимых данных для этого журнала подготовьте самостоятельно.



4. Подготовьте сообщение по таким темам:

- а) Использование баз данных в первых украинских ЭВМ.
- б) Использование баз данных в первых автоматизированных системах управления (АСУ) предприятием, разработанных Институтом кибернетики НАН Украины.
- в) Использование баз данных в современных коммуникационных устройствах.

## 3.2. Модели данных. Основные понятия реляционной базы данных



1. Что такое база данных? Для чего создаются базы данных?
2. Опишите структуру файловой системы операционной системы Windows XP.
3. Как осуществляется поиск нужных данных в операционной системе Windows XP, в Интернете?
4. Что такое модель «сущность-связь»? Для чего она используется?

### Понятие о моделях данных. Виды моделей данных

Основным назначением базы данных является гарантированное систематизированное хранение значительных объемов данных и предоставление доступа к ним пользователю или компьютерной программе. Поскольку база данных является достаточно сложным по своей структуре объектом, то перед созданием она предварительно проектируется - создается *модель базы данных*. При создании модели базы данных используют ту или иную модель данных. *Модель данных* имеет три составляющие:

1. *Структура данных* определяет способ организации данных, множество возможных типов данных (целые числа, действительные числа, текстовые данные, мультимедийные данные, массивы чисел или текстов и др.) и набор операций, которые можно выполнить над данными конкретного типа.
2. *Средства обработки данных* определяют набор команд (аналогично системе команд исполнителя алгоритма), которые обеспечивают обработку данных в зависимости от способа их организации.
3. *Ограничение целостности* определяет требования для обеспечения правильности данных в любой момент времени. Различают требования, которые налагаются соответствующим видом модели данных, и требования, которые устанавливает пользователь.

Самыми распространенными являются такие виды моделей данных: иерархическая, сетевая, реляционная и объектно-реляционная.

**Иерархическая** (греч. *ιεραρχία* - священная власть) модель данных определяет организацию данных об объектах в виде дерева. В иерархической модели структура данных предусматривает, что у каждого объекта есть только один объект высшего уровня, которому он подчинен (родительский), и могут быть несколько подчиненных (потомков). Исключение составляет только наивысший по иерархии объект - у него нет родительского объекта.

Примером иерархической организации данных является файловая структура, которая используется при размещении данных на дисках, например FAT32. В указанной файловой структуре на диске есть один основной родительский объект (корневая папка), который содержит данные о подчиненных объектах - папках и файлах. В свою очередь эти объекты могут содержать разнообразные данные, в том числе и о подчиненных им папках и файлах.

Еще один пример иерархической организации данных представлен на рисунке 3.6.

Обращаем ваше внимание, что каждый родительский объект в совокупности с подчиненными объектами (потомками) можно рассматривать как отдельное дерево.

Для обработки данных в иерархической модели данных используется такой набор команд:

- найти указанное дерево, например дерево 8 класс;
- перейти от одного дерева к другому, например от дерева Начальная школа к дереву Старшая школа;
- перейти от родительского объекта к объекту-потомку внутри дерева, например от объекта 6-А к объекту Семенец Богдан;
- перейти от одного объекта к другому в порядке, предусмотренном иерархической структурой, например от объекта 7-Б к объекту 1-А;
- вставить новый объект в указанном месте;
- удалить текущий объект

и др.

*Ограничение целостности* в иерархической модели данных, в частности, предусматривает сохранение связей между родительскими объектами и потомками. Основное правило ограничения целостности - ни один подчиненный объект (потомок) не может существовать без родительского объекта, за исключением одного основного родительского объекта. То есть объект 9-Б не может существовать без родительского объекта 9 класс, потому что если нет девятого классов в школе, то не может существовать список учеников одного из девятого классов.

Иерархическую модель данных удобно использовать, когда нужно создать базу данных предметной области, объекты которой также имеют между собой иерархическую зависимость.

При значительном количестве данных в базах данных, построенных на иерархической модели, на поиск нужных данных можно потратить много времени. Например, попробуйте найти среди всех файлов жестких дисков вашего компьютера файл, который содержит определенный фрагмент текста. Поиск может длиться несколько минут, а то и несколько десятков минут. За это же время в Интернете подобный поиск будет длиться максимум несколько секунд, при этом обрабатываются значительно большие объемы данных. При поиске данных в Интернете используются созданные поисковыми машинами базы данных, в которых содержание разнообразных сайтов проанализировано и классифицировано, и для сохранения этих данных используют другие модели данных, например реляционную. В современных операционных системах, таких как Windows 7, для сокращения времени поиска используют индексацию данных - создание дополнительной базы данных об объектах файловой системы и их содержании.

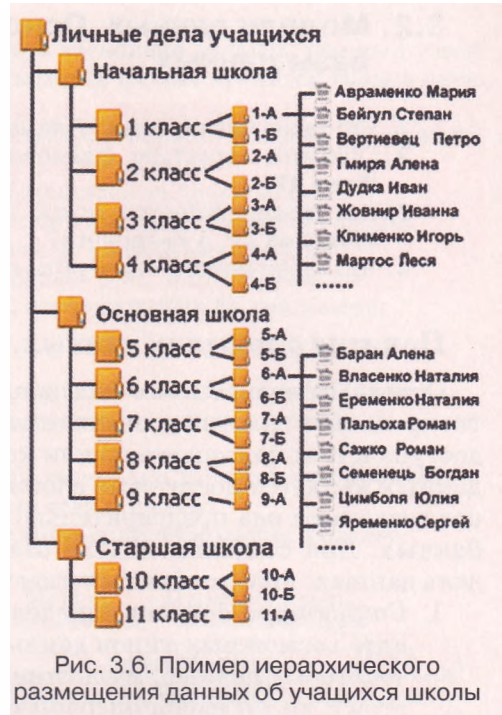


Рис. 3.6. Пример иерархического размещения данных об учащихся школы



Иерархическую модель данных трудно использовать для предметной области, в которой объекты связаны между собой более сложной зависимостью, чем иерархия. Расширенные возможности для описания такой предметной области предоставляет *сетевая модель данных*, разработанная в конце 1960-х годов. Автором концепции сетевой модели данных является американский ученый Чарльз Бахман (род. в 1924 г.) (рис. 3.7). Сетевая структура данных предусматривает, что у каждого объекта может быть как несколько объектов-потомков, так и несколько родительских объектов. Пример связей между объектами при использовании сетевой модели данных изображен на рисунке 3.8.



Рис. 3.7. Чарльз Бахман

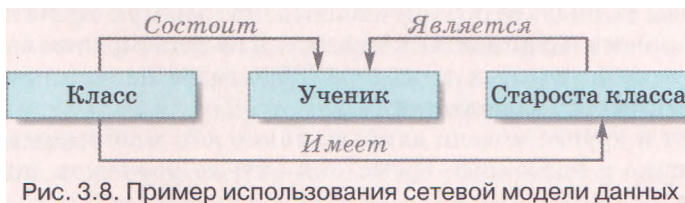


Рис. 3.8. Пример использования сетевой модели данных

Для *обработки данных* в сетевой модели данных используется такой набор команд:

- найти указанный объект среди однотипных объектов, например объект с данными об ученике *Степаненко*,
- перейти от родительского объекта к первому потомку, используя определенную связь, например к объекту с данными о первом ученике класса;
- вставить новый объект в указанном месте;
- удалить текущий объект;
- включить объект в определенную связь

И др.

*Ограничение целостности* в сетевой модели данных предусматривает сохранение связей между объектами.

Вместе с тем использование сетевой модели данных осложняется при значительном увеличении количества объектов предметной области и усложнении связей между этими объектами. Проблемы возникают и во время модификации базы данных: при добавлении новых связей, замене объектов, связанных между собой, и т. п.

Для упрощения описания объектов и связей между ними в 1970 году американским ученым Эдгаром Франком Коддом (1923-2003) (рис. 3.9) была предложена *реляционная модель данных*. Математик по образованию, он ввел в теорию баз данных математический подход, который основывается на теории множеств.

Основой структуры данных этой модели является таблица. В таблицах каждая строка содержит набор значений свойств одного из объектов предметной области. Каждый столбец таблицы содержит набор значений определенного свойства объектов предметной области.

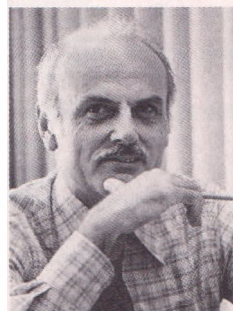


Рис. 3.9. Эдгар Франк Кодд

Для описания структуры данных Кодд употребил термин «relation» (англ. *relation* - отношение), а модель данных стали называть реляционной.

Учитывая, что таблица реляционной базы данных состоит из элементов определенных множеств, для обработки данных этой таблицы используются операции над множествами.

**Ограничение целостности** в реляционной модели базы данных предусматривает соблюдение двух принципов: обязательная возможность идентификации объекта базы данных, используя уникальность набора значений его свойств, указанных в строке реляционной таблицы, и обязательная корректность связей между таблицами базы данных.

Более детально основные понятия баз данных, построенных на реляционной модели данных, будут рассмотрены в следующих пунктах.

В последние годы некоторые ученые указывают на несовершенство реляционной модели данных, ее ограниченность при работе с мультимедийными данными и данными о составных объектах. Для расширения возможностей реляционной модели данных с конца 1990-х годов начала использоваться объектно-реляционная модель данных.

Существуют и другие модели данных, такие как многомерные, комбинированные, однако в настоящее время они еще не получили широкого распространения.

## Основные понятия реляционной базы данных

В соответствии с моделью данных, которая лежит в основе базы данных, различают иерархические, сетевые, реляционные и объектно-реляционные базы данных. В дальнейшем рассмотрим реляционные базы данных.

Ознакомимся с основными понятиями реляционной базы данных на примере таблицы, представленной на рисунке 3.10. Как вы уже знаете, основным объектом реляционной базы данных является таблица. Как любая



Рис. 3.10. Таблица реляционной базы данных

двухмерная таблица, таблица реляционной базы данных состоит из строк и столбцов. Для хранения данных о сущностях некоторой предметной области могут быть использованы несколько таблиц, которые, возможно, связаны между собой.

Строку таблицы реляционной базы данных называют записью, или кортежем. Запись содержит значение свойств одного экземпляра сущности. Например, в таблице Страны мира (рис. 3.10) пятая сверху запись содержит такой набор данных: *Австралия, Австралия и Океания, 7636, 19138, 01.01.1901, Конституционная монархия*. Эти данные являются значениями соответствующих свойств одного экземпляра сущности Страна мира: название, часть света, площадь, количество населения, дата образования государства и государственный строй. Другая запись содержит набор значений тех же свойств, но уже другого экземпляра сущности - другой страны.

Столбец таблицы реляционной базы данных называют полем, или атрибутом. Поле имеет имя, которое отображает название свойства. Поле содержит множество значений одного свойства всех экземпляров сущности. Для каждого поля во время проектирования таблицы базы данных устанавливается тип данных.

Для таблицы реляционной базы данных обязательное требование идентификации экземпляров сущности означает, что значения минимум в одном из полей не должны повторяться. Такое поле называют ключевым, или ключом таблицы. Ключевые поля еще называют идентификаторами. В отдельных реализациях реляционных баз данных допускается идентификация объекта базы данных по набору значений нескольких свойств, например трех полей *Фамилия, Имя, Отчество*.

В таблице базы данных, представленной на рисунке 3.10, как ключевое можно использовать поле *Название*, поскольку не существует одинаковых названий стран и поэтому значения в этом поле являются уникальными для каждой записи.

### Проверьте себя

- 1°. Объясните, что такое модель данных. Для чего создается модель данных?
- 2°. Назовите составляющие модели данных.
- 3°. Объясните, что определяет структура данных в модели данных.
- 4°. Какие типы данных могут использоваться в базах данных?
- 5°. Что такое ограничение целостности в модели данных?
- 6°. Назовите самые распространенные виды моделей данных.
- 7°. Опишите иерархическую модель данных, требования к ее структуре данных, средства обработки данных и ограничения целостности.
- 8°. Опишите сетевую модель данных, требования к ее структуре данных, средства обработки данных и ограничения целостности.
- 9°. Когда и кем была разработана теория реляционной модели данных?
- 10°. Опишите реляционную модель данных, требования к ее структуре данных, средства обработки данных и ограничения целостности.
- 11°. По значению какого свойства определяются типы баз данных? Какие вы знаете типы баз данных?
- 12°. Что является основным объектом реляционной базы данных?
- 13°. Опишите таблицу реляционной базы данных и ее основные части.
- 14°. Что такое ключевое поле? Каковы требования к ключевому полю?




**Выполните задания**

1°. Определите, какую модель данных иллюстрирует каждое из изображений (рис. 3.11).

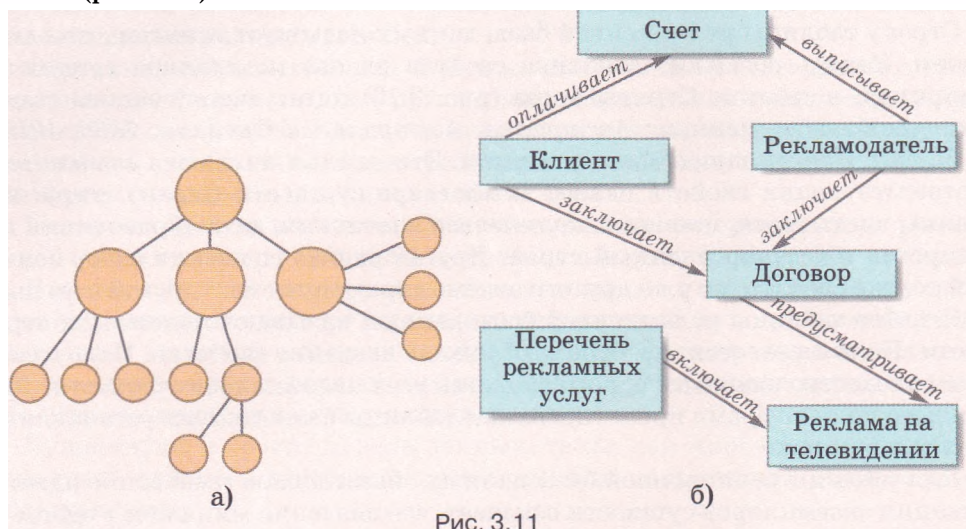


Рис. 3.11

2°. Определите, какую модель данных иллюстрирует каждое из изображений (рис. 3.12).

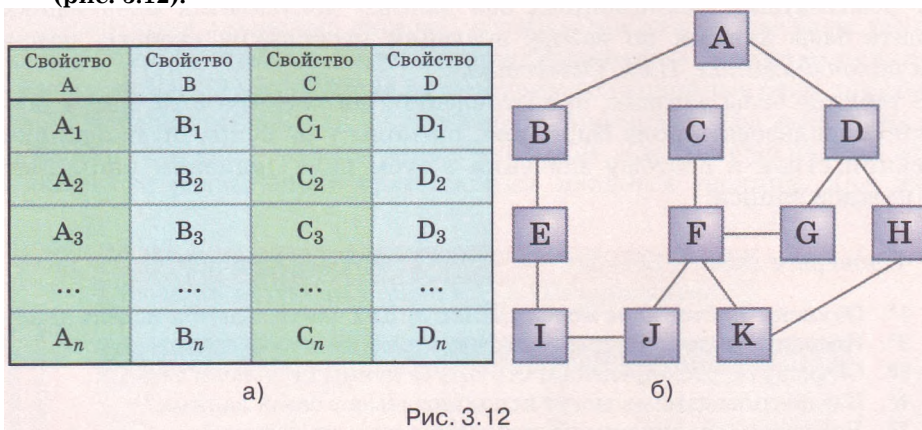



Рис. 3.12

3\*. В фирме «Бит и Байт» руководство осуществляет генеральный директор, которому подчинен совет директоров из четырех лиц - директора по маркетингу, директора по производству, директора по персоналу и директора по экономическим вопросам. Директору по маркетингу подчинен отдел маркетинга, а директору по производству - три цеха по сборке компьютеров. В подчинении директора по экономическим вопросам находятся отдел экономического прогнозирования и бухгалтерия. Используя одну из моделей данных, разработайте графическую модель управления фирмой.

 4\*. Необходимо создать базу данных библиотеки, в которой должны сохраняться такие сведения о книгах: *автор* (фамилия, имя, отчество каждого из авторов); *название*; *тип издания*; *место издания*; *название издательства*; *год издания*; *количество страниц*; *количество томов*; *номер тома*; *библиотечный шифр*. Предложите модель данных для этой базы данных.





5°. Рассмотрите рисунок 3.13. Предложите варианты проведения линий связей между объектами для случая, когда рисунок будет иллюстрировать модель базы данных, которая построена:

- а) на иерархической модели данных;
- б) на сетевой модели данных.

6°. Рассмотрите фрагмент таблицы реляционной базы данных, представленный на рисунке 3.14.

Определите:

- а) названия объектов, выделенных на рисунке другим цветом и обозначенных цифрами 1 и 2;
- б) ключевое поле для этой таблицы, объясните свой выбор;
- в) название сущности, данные о которой представлены в таблице;
- г) сколько экземпляров сущности представлено на рисунке;
- д) какие свойства экземпляров сущности обозначены в данной таблице.

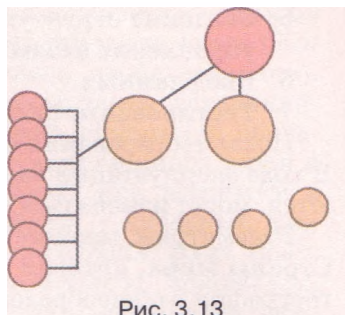


Рис. 3.13

№ личного дела	Фамилия	Имя	Отчество	День	Месяц	Год рождения
Ф-13	Руденко	Сергей	Александрович	28	12	1991
Ф-08	Орловский	Богдан	Виталиевич	15	10	1990
Ф-07	Карлаш	Марина	Анатолиевна	16	10	1990
Ф-06	Михно	Алексей	Иванович	2	4	1992
Ф-26	Паленка	Олег	Александрович	30	5	1992
Ф-33	Костюк	Сергей	Владимирович	31	12	1991
Ф-17	Булаев	Сергей	Юрьевич	2	6	1990

2

1

Рис. 3.14

### 3.3. Этапы создания базы данных.

#### Понятие о системах управления базами данных



1. Что такое база данных? Для чего создаются базы данных?
2. Опишите составляющие модели данных.
3. Какие особенности реляционной модели данных?
4. Назовите и опишите этапы решения задач с использованием компьютера.

#### Этапы создания базы данных

Процесс создания базы данных является одним из примеров решения задач с использованием компьютера. К процессу проектирования БД следует относиться ответственно. Ошибки, допущенные на этом этапе, могут привести к неоправданным потерям времени и средств.

Создание баз данных предусматривает такие этапы:

1. Постановка задания. На этом этапе определяется цель создания базы данных, очерчивается предметная область, определяются потенциальные пользователи.
2. Создание словесной информационной модели базы данных, в которой выделяются сущности будущей базы данных, описываются связи между ними.
3. Создание модели «сущность—связь».
4. Определение модели данных.

5. Создание структуры базы данных с использованием одной из систем управления базами данных или одного из языков программирования.
6. Ввод данных.
7. Тестирование базы данных, ее коррекция.

После тестирования и коррекции работа с базами данных не завершается. В ходе эксплуатации базы данных, как правило, данные регулярно обновляются, могут изменяться связи между сущностями и т. п.

Рассмотрим процесс создания базы данных на примере базы данных Страны мира, которая может быть использована во время изучения курса географии в общеобразовательных учебных заведениях.

Как было отмечено, на первом этапе в ходе постановки задания следует дать ответы на такие вопросы:

- *цель создания базы данных:* использование при изучении географии в общеобразовательных учебных заведениях для обобщения сведений о странах мира;
- *предметная область:* страны мира;
- *потенциальные пользователи:* ученики 9-10 классов, учителя.

На втором этапе создания базы данных создается ее *информационная модель*: база данных должна содержать данные о названии страны, ее площади, количестве и плотности населения, о государственных символах: флаге и гимне, о национальной валюте, о дате образования или провозглашения независимости, о наличии дипломатических отношений с Украиной, об адресе сайта правительства, о государственном строе и экономическом положении страны на момент внесения данных. В базе данных должна быть предусмотрена возможность подготовки обобщенных данных по странам определенной части света, а также по странам, которые имеют определенный государственный строй.

На третьем этапе создания базы данных определяется:

- *модель «сущность-связь»:* модель «сущность-связь» предметной области Страны мира изображена на рисунке 3.15. В ней три сущности: Страна, Часть света и Государственный строй. Между сущностью Часть света и сущностью Страна устанавливается связь *один ко многим* (1:∞), то есть одному экземпляру сущности Часть света могут отвечать много



Рис. 3.15. Модель «сущность-связь» предметной области Страны мира

экземпляров сущности Страна. Каждая страна принадлежит к одной из частей света (связь обязательная), однако существует часть света (Антарктида), на которой нет ни одной страны (связь необязательная). Между сущностями Государственный строй и Страна также устанавливается связь *один ко многим* (1:oo). Связь для обеих сущностей обязательная - каждая страна имеет один из видов государственного строя и каждая из форм государственного строя существует как минимум в одной стране. Сущность Страна имеет такие свойства - *название, площадь, дипломатические отношения, количество населения, флаг, гимн, сайт правительства, дата образования, экономическое положение*. Сущность Государственный строй имеет два свойства - *название и глава правительства*, а сущность Часть света одно - *название*.

Далее определяется *модель данных*: реляционная.

Следующие этапы создания базы данных предусматривают использование специального программного обеспечения для создания и редактирования баз данных, и поэтому рассмотрим их позже.

## Понятие о системах управления базами данных

Для работы с базами данных используют специальные программы - системы управления базами данных (СУБД). Используя их, пользователи создают базы данных, выполняют поиск, обновление данных и т. п.



*Системы управления базами данных* — это прикладные компьютерные программы, предназначенные для создания, сохранения и использования баз данных.

В зависимости от модели данных, которая используется в СУБД, их разделяют на иерархические, сетевые, реляционные, объектно-реляционные и т. п.

В зависимости от размещения баз данных и компонентов, которые обеспечивают их обработку, СУБД разделяют на локальные и распределенные. Локальные СУБД обеспечивают хранение и обработку данных на локальном компьютере, а в распределенных - данные могут храниться и обрабатываться на разных компьютерах в локальной или глобальной сети.

В зависимости от способа доступа к базе данных СУБД разделяют на файл-серверные, клиент-серверные и интегрированные (встроенные). В файл-серверных СУБД файлы с данными размещаются на сервере, а на каждом клиентском компьютере установлена полная версия СУБД. Доступ с клиентского компьютера к данным файл-сервера осуществляется с использованием сети. Недостатком таких СУБД является необходимость устанавливать иногда достаточно дорогие полные версии программ на каждый компьютер, высокая загруженность сети во время передачи данных из клиентского компьютера на сервер и наоборот, необходимость в достаточно мощных компьютерах на рабочих местах клиентов. Позитивом являются невысокие требования к производительности компьютера, который является файл-сервером. Одним из вариантов использования файл-серверных СУБД является размещение и СУБД, и данных на одном клиентском компьютере.

Наиболее широко в последние годы развивается клиент-серверная технология использования баз данных (рис. 3.16). По этой технологии на сервере



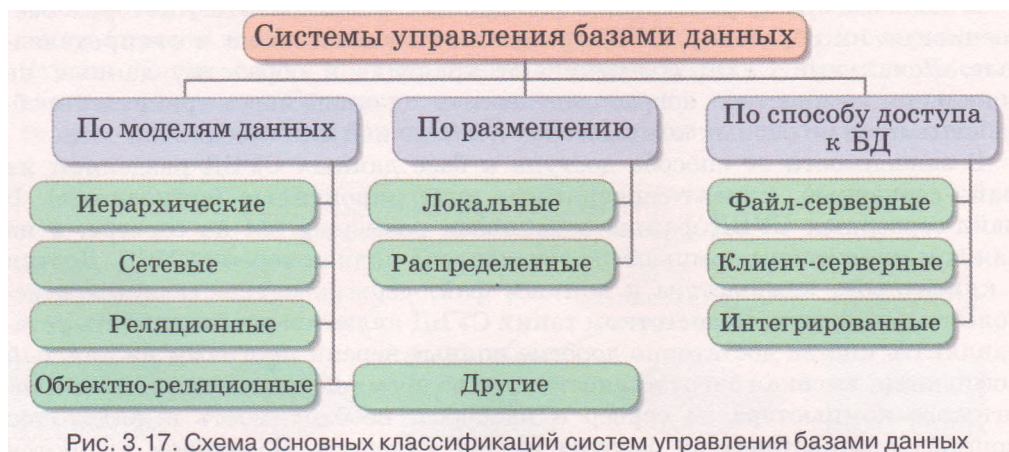


устанавливается серверная версия СУБД, на нем же, как правило, размещена и база данных. На клиентских компьютерах устанавливаются только небольшие по объему и функциональности клиентские версии СУБД, в задание которых входит формирование запросов на обработку данных и выведение результата обработки, полученного с сервера. Все операции с данными осуществляются на сервере. Такие СУБД не нуждаются в мощных клиентских компьютерах и не перегружают сеть, проще реализуется система обеспечения целостности данных, однако значительно повышаются требования к техническим характеристикам сервера.

Интегрированные (встроенные) СУБД используются в качестве составляющих других программных продуктов, например электронных энциклопедий, словарей, поисковых систем и т. п. Эти системы не нуждаются в отдельной установке и могут иметь ограниченный набор функций по управлению базой данных,

например не обеспечивать обновление данных. Доступ к данным осуществляется с использованием средств прикладной программы, в которую интегрирована СУБД.

Схема основных классификаций СУБД представлена на рисунке 3.17.



### Проверьте себя

1. Назовите этапы создания базы данных.
2. Опишите процесс постановки задания как одного из этапов создания базы данных.
- 3°. Какие задания решаются на втором этапе создания базы данных?



- 4\*. Объясните, какие задания реализуются в процессе создания модели «сущность-связь» и определения модели данных как этапа создания базы данных.
- 5°. Что такое системы управления базами данных?
- 6°. Какие типы СУБД выделяют в зависимости от модели данных, которые они используют?
- 7°. Чем отличаются локальные от распределенных СУБД?
- 8°. В чем разница между файл-серверными и клиент-серверными системами управления базами данных?
- 9°. Опишите схему основных классификаций СУБД по рисунку 3.17.

### Выполните задания

1. Разработайте первые четыре этапа создания:
  - а) базы данных учеников выпускных классов школы для хранения данных об их итоговых оценках по предметам и последующей печати таблиц успеваемости;
  - б) базы данных учета оборудования одного из кабинетов школы;
  - в) базы данных учета результатов школьных спортивных соревнований по легкой атлетике.
- 2\*. Есть шесть компьютеров, объединенных в локальную сеть. Один из компьютеров - сервер базы данных. Постройте схему связей между компьютерами при использовании файл-серверной технологии работы базы данных. Укажите на схеме, где обрабатываются данные из базы данных.
- 3\*. Постройте схему распределенной СУБД. Используйте при построении материалы Интернета.
4. Подготовьте сообщение по таким темам:
  - а) Использование интегрированных СУБД в мобильных устройствах.
  - б) Использование объектно-ориентированных СУБД.

## 3.4. Система управления базами данных Microsoft Office Access 2007



1. Опишите этапы создания базы данных.
2. Опишите основные понятия реляционной базы данных.
3. Что такое система управления базами данных?
4. Объясните, как создать новый документ в Microsoft Office Word; новую электронную таблицу в Microsoft Office Excel.

В предыдущем пункте были детально рассмотрены первые этапы создания базы данных - постановка задания, разработка информационной модели, модели «сущность-связь» и определение модели данных. Следующие этапы невозможно реализовать без ознакомления с одной из систем управления базами данных. Мы с вами будем изучать СУБД Microsoft Office Access 2007 (далее Access 2007 (англ. *Access* - доступ, выборка информации)) из пакета программ Microsoft Office 2007.

### Общая характеристика Access 2007

Система управления базами данных Access 2007 - это реляционная СУБД. Access 2007 принадлежит также к файл-серверным СУБД и предусматривает работу с небольшими по объему базами данных для личных нужд и для небольших организаций.

СУБД Access 2007 предназначена для создания и редактирования:


- таблиц, которые используются для хранения данных и установления связей между сущностями;
  - форм, которые используются для ввода и просмотра данных;
  - запросов, которые обеспечивают поиск, фильтрацию и обработку данных с использованием специального языка запросов - SQL (англ. *Structured Query Language* - язык структурированных запросов);
  - отчетов на основе имеющихся данных
- и др.

С перечнем свойств этих объектов и возможными значениями этих свойств ознакомимся в ходе создания и работы с базами данных.

Для сохранения баз данных Access 2007 использует в качестве основного формата файлов ACCDB (англ. *Access DataBase* - база данных Access), обеспечивая работу и с форматами файлов предыдущих версий программы - MDB (англ. *Microsoft DataBase* - база данных Microsoft).

### Запуск Access 2007. Окно программы

Access 2007 можно запустить на выполнение несколькими способами. Представим самые распространенные из них:

- выполнить *Пуск => Все программы => Microsoft Office => Microsoft Office Access 2007*;
- использовать ярлык программы на Рабочем столе ^ или на Панели быстрого запуска ;
- дважды щелкнуть на значке уже существующего файла базы данных формата, который определен в операционной системе для открытия с использованием СУБД Access 2007

В первом и втором случаях после запуска программы Access 2007 на экране открывается начальное окно программы (рис. 3.18).

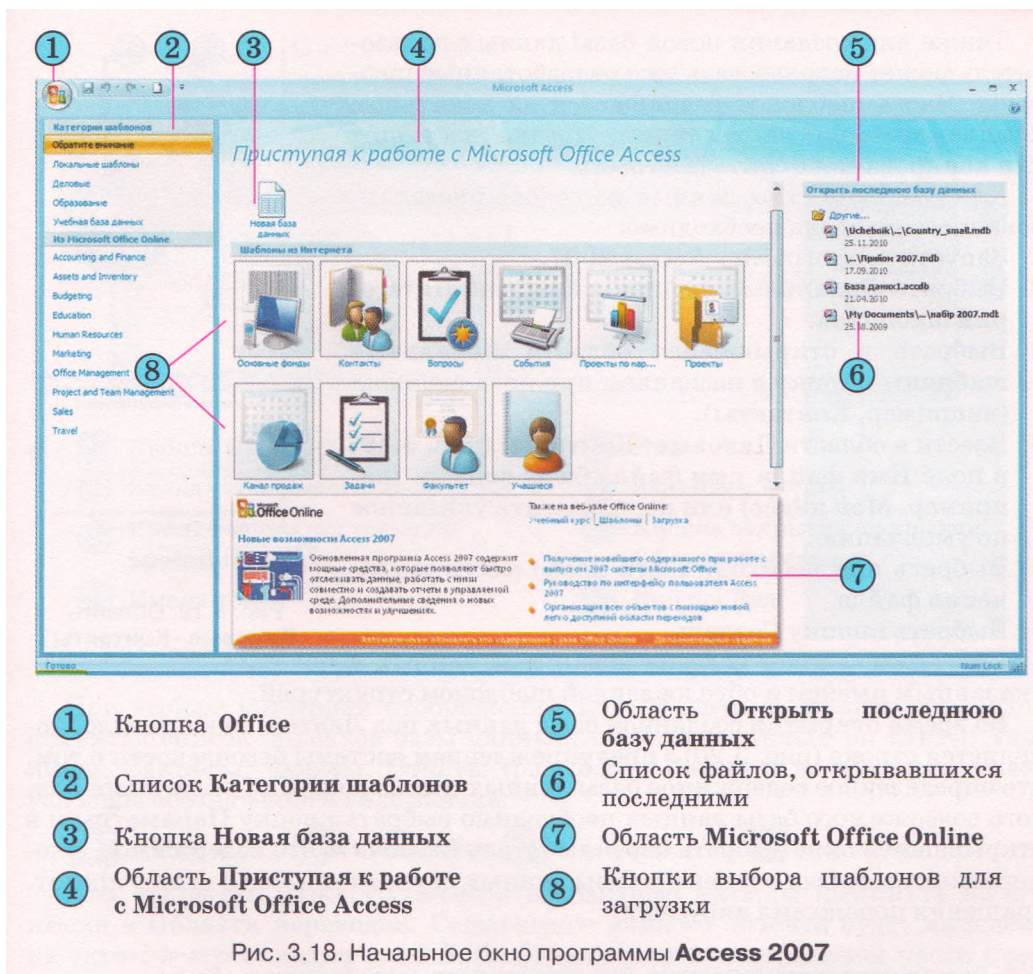
В строке заглавия окна размещены кнопка Office, кнопки управления окном и недоступная для использования в этом режиме работы Панель быстрого доступа. Основная часть окна разделена на три области. В левой находится список с именами вкладок групп шаблонов Категории шаблонов (Обратите внимание, Локальные шаблоны, Деловые и т. д.) и групп шаблонов, размещенных на сайте Microsoft Office Online (Accounting and Finance, Education, Marketing и др.).

Содержимое центральной области меняется в зависимости от избранной в левой части категории шаблонов. По умолчанию после запуска программы в центральной области окна Access 2007 отображаются элементы управления, изображенные на рисунке 3.18. Эта область горизонтально разделена на три части. Вверху есть только одна кнопка - Новая база данных



Среднюю часть занимают кнопки выбора шаблонов базы данных, а нижнюю - ссылки на справочную информацию, размещенную на сайте Microsoft Office.

В правой части начального окна программы размещена область Открыть последнюю базу данных со списком файлов баз данных, с которыми пользователь работал ранее. Над списком файлов размещена ссылка Другие..., используя которую, пользователь может открыть файл, размещенный во внешней памяти компьютера или в сети.





- |   |   |
|---|---|
| <p>1 Кнопка Office</p> <p>2 Список Категории шаблонов</p> <p>3 Кнопка Новая база данных</p> <p>4 Область Приступая к работе с Microsoft Office Access</p> | <p>5 Область Открыть последнюю базу данных</p> <p>6 Список файлов, открывавшихся последними</p> <p>7 Область Microsoft Office Online</p> <p>8 Кнопки выбора шаблонов для загрузки</p> |
|---|---|

Рис. 3.18. Начальное окно программы Access 2007

### Создание новой базы данных

В отличие от большинства прикладных программ работа с новой базой данных в Access 2007 начинается обязательно с создания файла этой базы данных.

Для создания файла новой пустой базы данных нужно:

1. Выбрать кнопку Новая база данных  в области Приступая к работе с Microsoft Office Access.
2. Ввести в правой части окна в области Новая база данных в поле Имя файла имя файла будущей базы данных или использовать указанное по умолчанию (последнее не рекомендуется).
3. Проверить, устраивает ли вас предложенное программой место сохранения файла, указанное под полем Имя файла.
4. Указать при необходимости новое размещение файла базы данных, для чего выбрать кнопку Поиск расположения для размещения базы данных .
5. Выбрать кнопку Создать.

Также для создания новой базы данных пользователь может использовать уже разработанные шаблоны. Часть шаблонов размещается на локальном компьютере при инсталляции, другие доступны для копирования с сайта Microsoft.

Для создания базы данных на основе инсталлированного шаблона необходимо:

1. Запустить программу Access 2007.
2. Выбрать Локальные шаблоны в списке Категории шаблонов.
3. Выбрать в открывшейся области Локальные шаблоны кнопку с названием нужного шаблона (например, Контакты).
4. Ввести в области Деловые Контакты (рис. 3.19) в поле Имя файла имя файла базы данных (например, Мой класс) или использовать указанное по умолчанию.
5. Выбрать при необходимости новое место сохранения файла.
6. Выбрать кнопку Создать.

Программой будет создана новая база данных с указанным именем и обусловленной шаблоном структурой.

Во время открытия созданной базы данных под Лентой, как правило, появляется строка (рис. 3.20) с предупреждением системы безопасности о том, что определенное содержимое базы данных отключено. Для включения полного содержимого базы данных необходимо выбрать кнопку Параметры и в открывшемся окне выбрать переключатель Включить это содержимое. Блокировка отдельных элементов базы данных осуществляется с целью предотвращения поражения вирусами.



Рис. 3.19. Область Деловые-Контакты

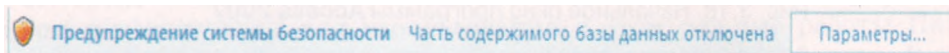




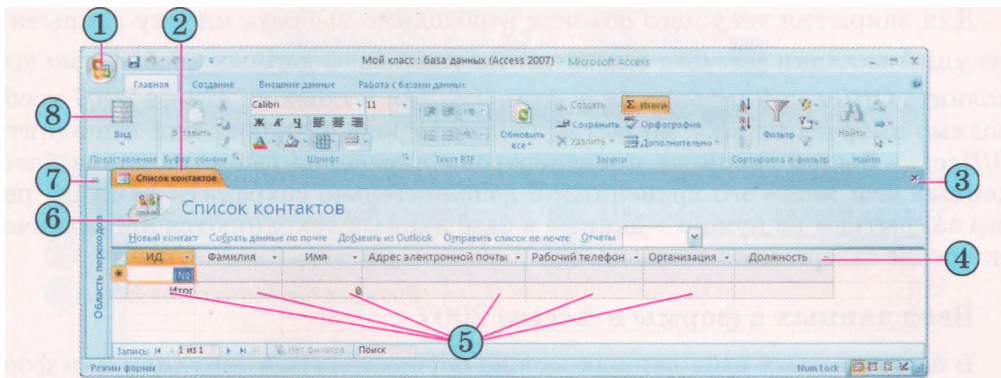
Рис. 3.20. Строка предупреждения системы безопасности при открытии файлов

На примере базы данных, созданной на основе шаблона Контакты, ознакомимся с интерфейсом и основными операциями с объектами базы данных.

После создания новой пустой базы данных на основе шаблона Контакты по умолчанию на экран выводится форма Список контактов (рис. 3.21) для ввода данных о лицах, с которыми вы планируете поддерживать деловые или дружеские контакты.

Чтобы просмотреть список других объектов базы данных, нужно открыть Область переходов, выбрав кнопку  (рис. 3.21, 7). Вид списка пользователь может изменить. Для изменения вида списка объектов необходимо выбрать кнопку  справа от заглавия области. Откроется список, который состоит из двух частей (рис. 3.22). В верхней части можно выбрать одну из категорий, в которые будут группироваться объекты, а в нижней - фильтр, в соответствии с которым в этой категории будут отбираться объекты: *Все объекты Access, Таблицы, Формы, Запросы или Отчеты.*









- |                                   |  |
|-----------------------------------|--|
| 1 Кнопка Office                   | 5 Поля                                       |
| 2 Вкладка открытого объекта БД    | 6 Заголовок формы                            |
| 3 Кнопка закрытия текущей вкладки | 7 Кнопка открытия/закрытия Области переходов |
| 4 Имена полей                     | 8 Кнопка Вид                                 |

Рис. 3.21. Окно СУБД с формой базы данных, созданной на основе шаблона **Контакты**

При выборе категории отображения Тип объекта в Области переходов появится список объектов базы данных, сгруппированный по типам: Таблицы, Запросы, Формы, Отчеты (рис. 3.23). Каждый из объектов имеет условное обозначение в виде значка:

 - таблицы;  - запросы;  - формы;  - отчеты.

Для открытия нужного объекта необходимо дважды щелкнуть на его имени в Области переходов. Содержимое данного объекта будет выведено на экран и его вкладка (рис. 3.21, 2) появится в центральной части окна программы. Исключение составляют отдельные виды форм, которые открываются в отдельном окне.

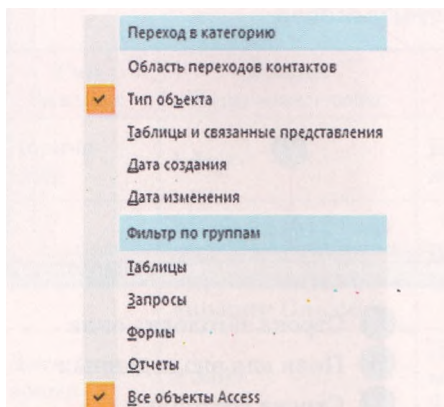


Рис. 3.22. Список выбора категории и фильтра отображения объектов в **Области переходов**

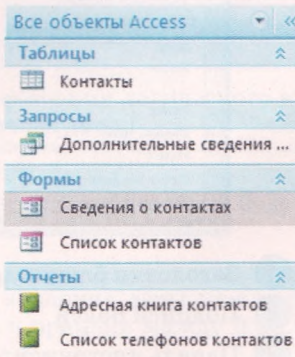



Рис. 3.23. Список объектов в **Области переходов**

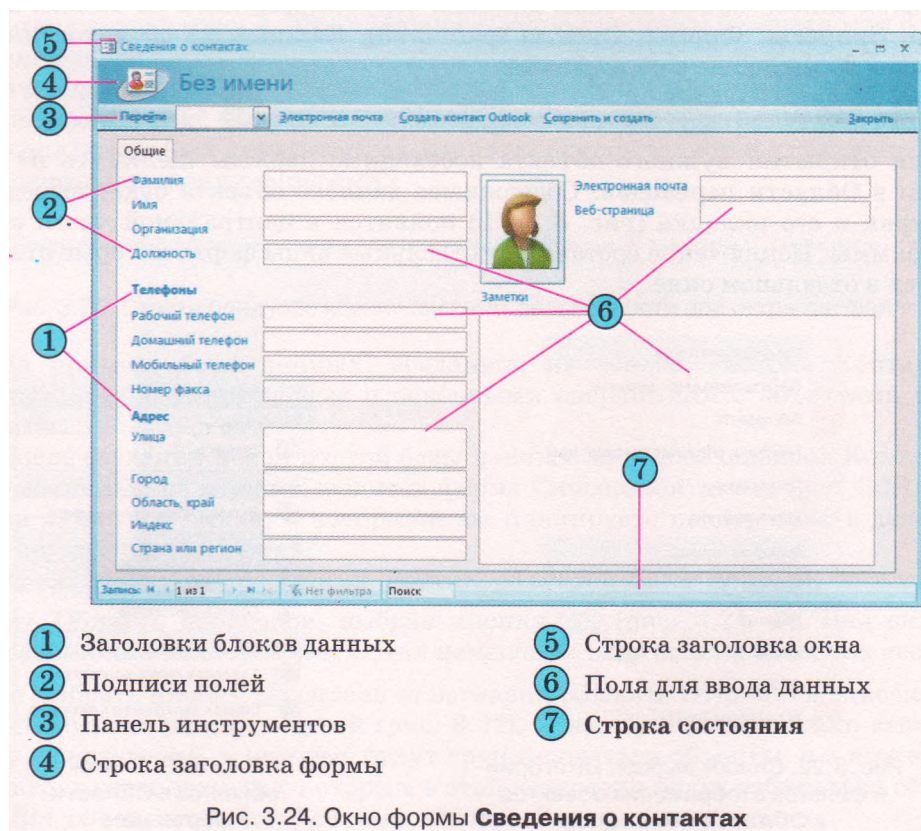
Для закрытия текущего объекта необходимо выбрать кнопку закрытия текущей вкладки . Для закрытия текущей базы данных необходимо выполнить Office => Закрыть базу данных. Для выхода из Access 2007 необходимо выбрать кнопку закрытия окна в Строке заголовка или выполнить Office => Выход. Если не предусмотрено изменение формата файла базы данных или места его хранения, то дополнительно сохранять файл БД перед закрытием не нужно - данные в базе данных сохраняются автоматически после завершения их ввода.

## Ввод данных в формы в Access 2007

В базах данных ввод данных можно осуществлять в таблицах или формах. Чаще пользователь вводит данные в формы.

Рассмотрим ввод данных на примере уже созданной на основе шаблона Контакты базы данных Мой класс. Откроем в этой базе форму Сведения о контактах. Для этого нужно дважды щелкнуть в Области переходов на имени формы. Указанная форма откроется в отдельном окне (рис. 3.24). Такой тип формы называют формой одного элемента, потому что на экран выводятся данные только об одном экземпляре сущности.

Для ввода данных в поля формы необходимо установить курсор в нужное поле и ввести с клавиатуры или вставить из Буфера обмена данные. Для перехода к следующему полю можно использовать клавишу Tab.



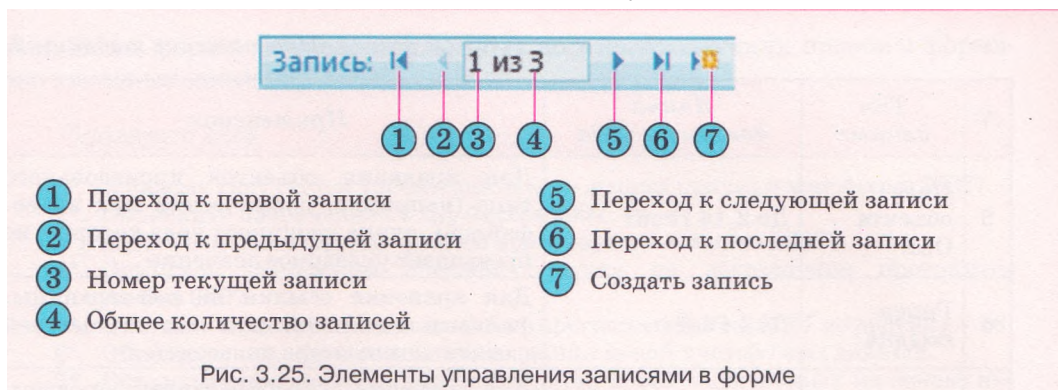


Рис. 3.25. Элементы управления записями в форме

Как уже было отмечено ранее, совокупность значений свойств экземпляра сущности является записью базы данных. Завершив ввод данных первой записи, нужно перейти к вводу данных следующей записи. Для этого используются элементы управления в левой части Строки состояния (рис. 3.25).

Введенные данные автоматически сохраняются в базе данных.

### Типы данных

Во время ввода данных в поля базы данных следует придерживаться не только определенного шаблона, но и вводить данные в соответствии с избранным типом. Как вы знаете, в одном поле базы данных могут храниться данные только одного типа.

Для хранения данных определенного типа в памяти компьютера используется разная длина двоичного кода. Так, для хранения данных логического типа нужен всего 1 бит памяти, а для текстовых данных - до 256 байт (или до 512 байт при использовании Unicode). Чем меньше объем данных, тем быстрее происходит обработка данных. Поэтому при проектировании баз данных пытаются минимизировать объемы данных за счет рационального использования данных разных типов (табл. 3.2).

Таблица 3.2. Основные типы данных, которые используются в Access 2007

№	Тип данных	Длина двоичного кода	Применение
1	Логический	1 бит	Для хранения значения логического выражения (истина или ложь)
2	Текстовый	256 байт (512 байт при использовании таблицы кодировки Unicode)	Для хранения текста длиной от 0 до 255 символов
3	Дата / время	8 байт	Для хранения значения даты между годами 100 и 9999 и времени между 0:00:00 и 23:59:59
4	Поле MEMO	До 2,14 Гбайт	Для хранения текста, длина двоичного кода которого не превышает указанного значения

№	Тип данных	Длина двоичного кода	Применение
5	Поле объекта OLE	До 2,14 Гбайт	Для хранения объектов произвольного типа (например, графических или видео-файлов), длина двоичного кода которых не превышает указанное значение
6	Гипер-ссылка	До 1 Гбайт	Для хранения ссылки на веб-страницы, файлы в локальной сети или во внешней памяти компьютера пользователя
7	Счетчик	4 байт	Для хранения сгенерированных программой уникальных данных (целых чисел от 0 до 4 294 967 295) и последующего определения поля как первичного ключа
8	Числовой	В зависимости от размера	См. табл. 3.3


**Таблица 3.3. Размеры числового типа данных, которые используются в Access 2007**

№	Размер	Длина двоичного кода	Применение
1	Байт	1 байт	Для хранения значений целых чисел в диапазоне от 0 до 255
2	Целое	2 байт	Для хранения значений целых чисел в диапазоне от -32 768 до 32 767
3	Длинное целое	4 байт	Для хранения значений целых чисел в диапазоне от -2 147 483 648 до 2 147 483 647
4	Одинарное с плавающей точкой	4 байт	Для хранения значений действительных чисел в диапазонах: <ul style="list-style-type: none"> <li>• от <math>-3,402823 \cdot 10^{38}</math> до <math>-1,401298 \cdot 10^{-46}</math> для отрицательных значений</li> <li>• от <math>1,401298 \cdot 10^{-46}</math> до <math>3,402823 \cdot 10^{38}</math> для положительных значений</li> <li>• числа 0</li> </ul>
5	Двойное с плавающей точкой	8 байт	Для хранения значений действительных чисел в диапазонах: <ul style="list-style-type: none"> <li>• от <math>-1,79769313486232 \cdot 10^{308}</math> до <math>-4,94065645841247 \cdot 10^{-324}</math> для отрицательных значений</li> <li>• от <math>4,94065645841247 \cdot 10^{-324}</math> до <math>1,79769313486232 \cdot 10^{308}</math> для положительных значений</li> <li>• числа 0</li> </ul>

Кроме типа данных, Access 2007 предоставляет возможность установить ограничение на форму представления данных или на их значения. Так, для типа данных Текст можно установить ограничение на количество символов, для полей с типом данных Дата/время - на форму представления, например короткий формат даты. Эти ограничения называют форматами дан-



**ных. Виды форматов числовых данных во многих случаях подобны форматам числовых данных в Excel.**

 **Проверьте себя**

- 1°. К какому типу систем управления базами данных принадлежит Access 2007?
- 2°. Объясните назначение СУБД Access 2007.
- 3°. Какой основной формат файлов используется в Access 2007?
- 4°. Опишите возможные способы запуска на выполнение программы Access 2007.
- 5°. Опишите структуру начального окна программы Access 2007 по рисунку 3.18.
- 6°. Объясните последовательность создания новой пустой базы данных.
- 7°. Объясните последовательность создания новой базы данных на основе инсталлированного шаблона, на основе шаблона из Интернета.
- 8°. Для чего выполняется блокировка определенного содержимого баз данных? Что нужно сделать для выключения этой блокировки?
- 9°. Опишите окно Access 2007 по рисунку 3.21.
- 10°. Объясните, для чего используется Область переходов.
- 11°. Как открыть объект базы данных, используя Область переходов?
- 12°. Как закрыть базу данных? Как закрыть программу Access 2007?
- 13°. Опишите последовательность ввода данных в форму базы данных.
- 14°. Какие типы данных используются в Access 2007?
- 15\*. Объясните, чем отличается тип данных Текст от типа данных Поле МЕМО. Объясните потребность в каждом из этих типов.
- 16\*. Объясните, чем отличается размер числового типа данных Целое от размера числового типа данных Одинарное с плавающей точкой. Объясните потребность в каждом из этих типов.
- 17°. Для чего используется в базах данных тип данных Поле объекта OLE? Какие данные можно поместить в поле с этим типом данных?
- 18\*. Почему в базах данных используются разные типы данных? Нельзя ли все типы данных свести к одному, какому-то универсальному?
- 19°. Какие размеры числового типа данных используются в Access 2007?
- 20°. Что такое формат данных? Для чего он используется?

 **Выполните задания**

- 1°. Укажите названия объектов, изображенных на рисунке 3.26.

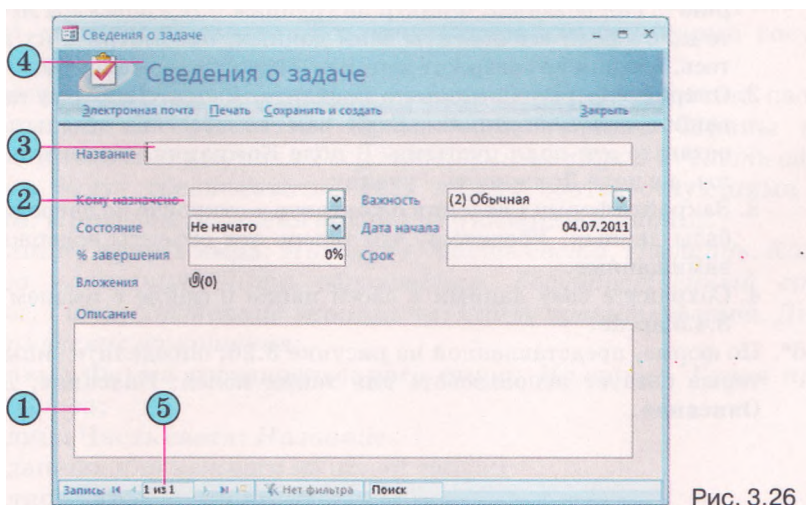


Рис. 3.26



2\*. Укажите названия объектов, изображенных на рисунке 3.27. Объясните их назначение.

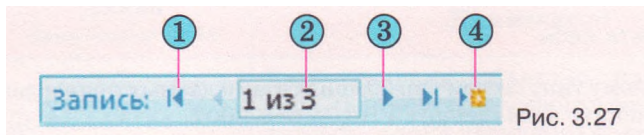



Рис. 3.27

3\*. Откройте файл базы данных (например, **Тема 3\Задания 3.4\образец 3.4.3.accdb**) и:

1. Включите полное содержание базы данных.
2. Откройте **Область переходов** и установите такие ее настройки: **Переход в категорию** - *Дата изменения*; **Фильтр по группам** - *Все даты*.
3. Определите, какой из объектов редактировался последним и какой - давно.
4. Измените настройки **Области переходов** на такие: **Переход в категорию** - *Тип объекта*, **Фильтр по группам** - *Все объекты*. *Access*.
5. Определите, сколько и какие объекты включены в базу данных.
6. Закройте **Область переходов** и базу данных.



4\*. Откройте файл базы данных (например, **Тема 3\Задания 3.4\образец 3.4.4.accdb**) и:

1. Просмотрите начальную страницу базы данных.
2. Включите полное содержание базы данных.
3. Закройте начальную страницу базы данных выбором кнопки .
4. Выберите в **Диалоговом окне входа** произвольное имя заказчика, например *Sergienko Mariya*, и кнопку **Вход**.
5. Откройте **Область переходов** и установите такие ее настройки: **Переход в категорию** - *Тип объекта*; **Фильтр по группам** - *Все объекты*. *Access*.
6. Откройте группу **Таблицы** и определите, сколько таблиц есть в этой базе данных.
7. Просмотрите одну из таблиц, например **Клиенты**. Определите основные объекты таблицы реляционной базы данных.
8. Закройте открытую таблицу, а затем и базу данных.

5\*. Откройте файл базы данных (например, **Тема 3\Задания 3.4\образец**

**3.4.5.accdb**) и:

1. Установите такие настройки **Области переходов**: **Переход в категорию** - *Тип объекта*; **Фильтр по группам** - *Все объекты*. *Access*. Откройте по очереди все объекты базы данных, просмотрите их и удостоверьтесь, что они не содержат данных. Закройте все объекты.
2. Откройте форму **Сведения о контакте**. Введите в форму данные о пяти ваших одноклассниках. Если вам неизвестны отдельные данные - оставьте эти поля пустыми. В поле **Компания** укажите название школы, а в поле **Должность** - ученик.
3. Закройте форму **Сведения о контакте** и откройте по очереди все объекты базы данных. Убедитесь, что теперь эти объекты содержат введенные вами данные.
4. Сохраните базу данных в своей папке в файле с именем **упражнение 3.4.5.accdb**.



6\*. По форме, представленной на рисунке 3.26, определите типы данных, которые следует использовать для таких полей: **Название**, **Дата начала**, **Описание**.

### 3.5. Отображение модели «сущность-связь» на базу данных. Работа с таблицами базы данных



1. Объясните этапы создания новой базы данных.
2. Какие типы данных используются в Access 2007? Для чего используются разные типы данных?
3. Как создать новую пустую базу данных в Access 2007?
4. Опишите последовательность ввода данных в форму базы данных.

#### Отображение модели «сущность—связь» на базу данных

В пункте 3.3 были рассмотрены этапы создания базы данных и определено содержание нескольких первых этапов на примере создания базы данных Страны мира.

Следующим этапом является создание структуры базы данных с использованием одной из систем управления базами данных или одного из языков программирования.

В ходе создания структуры базы данных необходимо определить:

1. Количество таблиц базы данных.
2. Перечень полей для каждой таблицы, перечень ключевых полей.
3. Типы данных для каждого поля каждой из таблиц с учетом возможностей конкретной СУБД.
4. Типы связей между объектами таблиц, если используется несколько таблиц.
5. Виды и количество других объектов базы данных: форм, запросов, отчетов.



Создание структуры базы данных на основе модели «сущность-связь» называется *отображением модели на базу данных*.

Рассмотрим один из вариантов отображения модели «сущность-связь» Страны мира на базу данных с использованием СУБД Access 2007. Структура этой базы данных такова:

1. Количество таблиц соответствует количеству сущностей (рис. 3.15) модели «сущность-связь». Их *три*: Страны мира, Форма государственного строя, Часть света.
2. Перечень полей для каждой из таблиц отвечает перечню свойств для каждой из сущностей. Кроме того, таблица Страны мира должна иметь еще два поля *Государственный строй* и *Часть света*, которые будут обеспечивать связь между соответствующими таблицами. Приведем перечень полей для указанных таблиц:
  - таблица Страны мира: *Название, Часть света, Площадь, Количество населения, Дата образования, Государственный строй, Флаг, Гимн, Состояние экономики, Сайт правительства, Дипломатические отношения;*
  - таблица Форма государственного строя: *Название, Глава правительства;*
  - таблица Часть света: *Название.*
3. Типы данных для каждого из полей таблиц:
  - таблица Страны мира:

- о **Название** - тип данных - *текстовый*, **размер** - 70 символов; **ключевое поле**;
  - о **Часть света** — тип данных — *мастер подстановок*. Этот тип данных предусматривает использование данных из другой таблицы;
  - о **Площадь** - тип данных — *числовой*, **размер** — *одинарное с плавающей точкой*;
  - о **Количество населения** - тип данных - *число*, **размер** - *длинное целое*;
  - о **Дата образования** - тип данных - *дата/время*, **формат** - *краткий формат даты*;
  - о **Государственный строй** - тип данных - *мастер подстановок*;
  - **таблица Форма государственного строя:**
    - о **Название** - тип данных — *текстовый*, **размер** — 50 символов;
    - о **Глава правительства** - тип данных - *текстовый*, **размер** - 30 символов;
  - **таблица Часть света:**
    - о **Название** - тип данных - *текстовый*, **размер** - 50 символов.
4. Типы связей (рис. 3.28):
- **таблица Форма государственного строя поле Название и таблица Страны мира поле Государственный строй** - *связь один ко многим*;
  - **таблица Часть света поле Название и таблица Страны мира поле Часть света** - *связь один ко многим*.
5. Другие объекты базы данных:
- **форма Страны мира с полями:** *Название, Часть света, Площадь, Количество населения, Дата образования, Государственный строй, Флаг, Гимн, Состояние экономики, Сайт правительства, Дипломатические отношения*;
  - **запросы:**
    - о **Площадь стран мира с полями:** *Название страны, Часть света, Площадь (с подсчетом общей площади определенной части света)*;
    - о **Государственный строй стран мира с данными из полей таблицы Страны мира:** *Название, Государственный строй*;
    - **Отчет Площадь и количество населения стран мира с данными из полей таблицы Страны мира:** *Название страны, Часть света, Площадь, Количество населения с группированием по Частям света*.

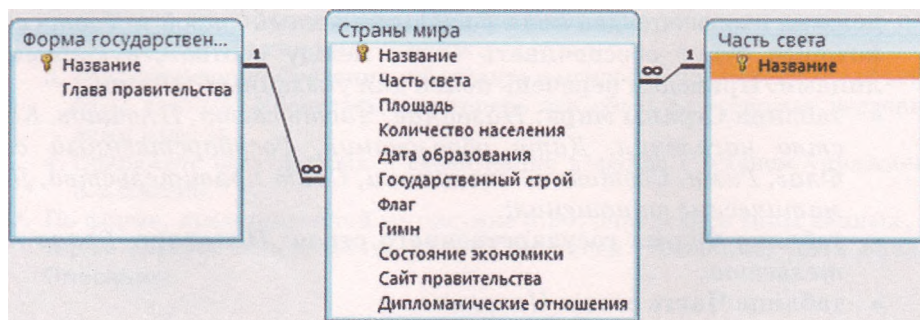


Рис. 3.28. Связи в базе данных **Страны мира**



## Создание таблиц базы данных с использованием Режима таблицы

Перейдем к созданию таблиц БД. Для этого создадим новую пустую базу данных (рис. 3.29), как это было описано в предыдущем пункте.

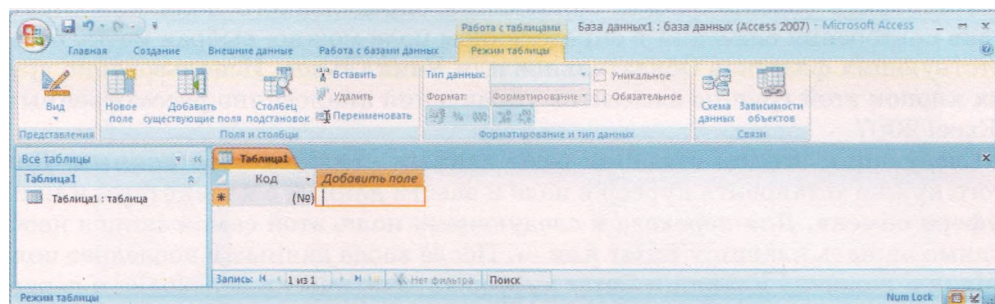


Рис. 3.29. Окно СУБД Access 2007 с открытой новой таблицей

В последующем возможны два основных способа создания полей таблицы и определения типов данных в них:

- 1) с использованием Режимы таблицы;
- 2) с использованием режима Конструктор.

Рассмотрим первый способ. Особенностью его является то, что СУБД Access 2007 автоматически определяет тип данных при вводе первого значения в поле и устанавливает значения форматов данных. Необходимо только создать поля с соответствующими именами. Одно из полей - Код - создается программой при создании новой пустой базы данных, для него устанавливается тип данных - Счетчик. Это поле определяется как ключевое.

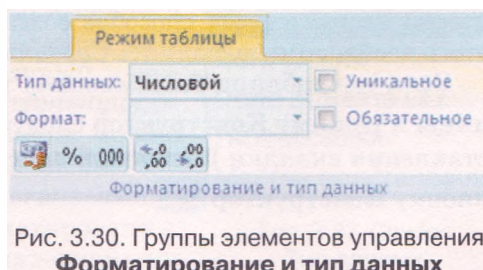
Для создания всех других полей с определенными именами следует выполнить такую последовательность действий:

1. Дважды щелкнуть на заглавии Добавить поле.
2. Ввести с клавиатуры или из Буфера обмена имя поля.
3. Нажать клавишу Enter или клавишу —к
4. Повторить выполнение пунктов 2 и 3 нужное количество раз для ввода имен всех необходимых полей.
5. Завершить процесс ввода имен полей выбором строки под именем поля или нажатием клавиши ↓.

Следует учитывать, что имена полей не должны содержать точек, восклицательного знака, квадратных скобок, скрытых символов (например, символа конца абзаца).

Для установки типов данных и их форматов используются элементы управления группы Форматирование и тип данных временной вкладки Работа с таблицами (рис. 3.30). Для установки типа и формата данных необходимо выполнить такую последовательность действий:

1. Сделать текущим нужное поле.
2. Выбрать в списке Тип данных



указанной группы (рис. 3.30) нужный тип поля, например для поля **Название таблицы** Форма государственного строя - *Текстовый*.

3. Установить выбором из списка **Формат** нужный формат представления данных (возможно лишь для типов данных *Числовой, Дата/время, Денежная единица, Логический*).

Установить обязательность заполнения определенного поля или отсутствие повторения значений в определенном поле можно, выбрав метку соответствующих флажков **Обязательное** или **Уникальное**. Использование других кнопок этой группы элементов управления аналогично рассмотренным в Excel 2007.

Следующим этапом создания базы данных является *ввод данных*. Для этого нужно установить курсор в поле и ввести данные с клавиатуры или из Буфера обмена. Для перехода к следующему полю этой самой записи необходимо нажать клавишу **Enter** или —к После ввода данных в последнее поле записи и нажатия клавиши **Enter** курсор автоматически переходит к первому полю следующей записи.

Так, для таблицы **Форма государственного строя** вводим такие значения в соответствующие поля (рис. 3.31):

Название	Глава правительства
Парламентско-президентская республика	Премьер-министр
Парламентская республика	Премьер-министр
Конституционная монархия	Премьер-министр
Президентская республика	Президент
Абсолютная монархия	Монарх
Военная диктатура	Диктатор (военный руководитель)
Однопартийная система	Глава партии

Рис. 3.31. Таблица **Форма государственного строя**



Подобным способом можно создать также и таблицу **Часть света** с одним полем **Название** и значениями: *Азия, Европа, Америка, Африка, Австралия и Океания, Антарктида*.

Для сохранения созданных таблиц следует выполнить:

1. *Панель быстрого доступа => Сохранить.*
2. Ввести в поле **Имя таблицы** окна **Сохранение нового имени таблицы**, например **Форма государственного строя**.
3. Выбрать кнопку **ОК**.

## Создание таблиц базы данных в режиме Конструктор

Для создания новой таблицы базы данных с более сложной структурой и для установления значений других свойств полей, как правило, используют второй из указанных способов создания таблиц - режим **Конструктор**. Для этого необходимо выполнить *Создание => Таблицы => Конструктор таблиц*.

Если же таблица была создана ранее, то для перехода от **Режима таблицы** к режиму **Конструктор** следует выбрать кнопку **Вид**  группы **Представления** вкладки **Главное**. В списке, который откроется, следует выбрать кнопку **Конструктор** 

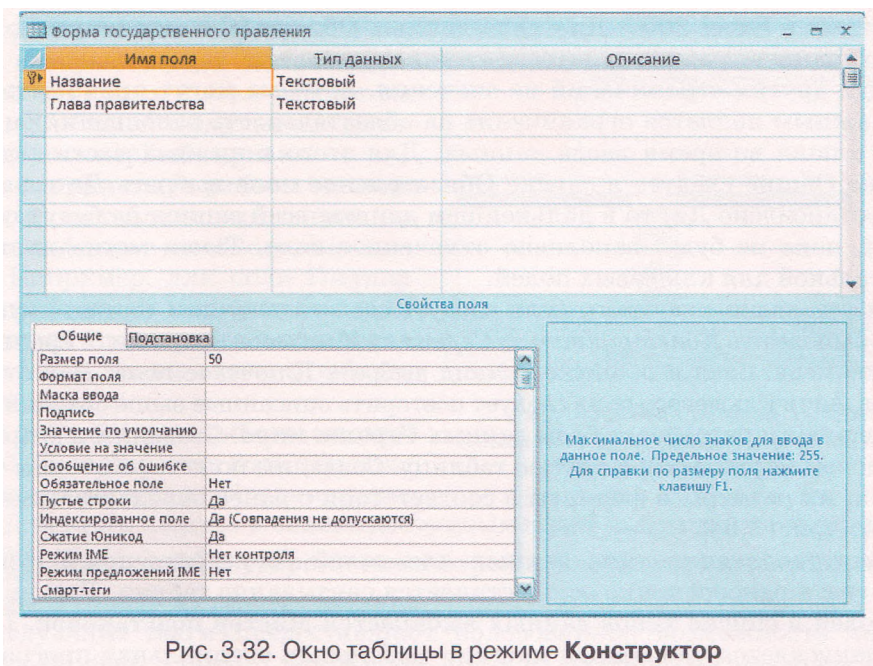


Рис. 3.32. Окно таблицы в режиме **Конструктор**

В обоих случаях откроется окно таблицы в представлении **Конструктор**, аналогичное представленному на рисунке 3.32. В верхней части окна есть три столбца: **Имя поля**, **Тип данных** и **Описание**. Левый столбец используется для ввода имен полей. Для установления типа данных следует выбрать соответствующее поле справа от имени поля и выбрать кнопку открытия списка. Откроется список типов данных (рис. 3.33), Среди них необходимо выбрать нужный.

После выбора типа данных поля можно изменить установленные по умолчанию и установить неустановленные значения свойств этого типа данных в нижней части окна на вкладке **Общие**. Содержимое этой вкладки меняется в зависимости от выбранного типа данных.

Так, для текстовых данных основным свойством, значение которого следует установить в поле **Размер поля**, является максимальное количество символов, которое можно будет вводить в поле. Для типа данных **Числовой** в списке **Размер поля** выбирается один из типов числовых данных, рассмотренных выше. Кроме того, для большинства числовых типов данных на вкладке **Общие** можно выбрать в списке **Формат** один из форматов отображения числа (*основной денежный, евро, фиксированный, с разделителями разрядов, процентный, экспоненциальный*). В списке **Число десятичных знаков** можно выбрать нужное значение этого свойства числа.

Для типов данных **Дата/время**, **Логический** и **Денежный** можно выбрать вид форматирования. Эти виды имеют много общего с аналогичными

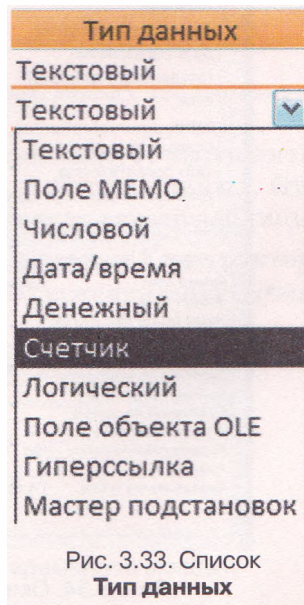


Рис. 3.33. Список **Тип данных**



форматами в Excel 2007. Для типа данных Объект OLE можно установить только подпись.

Среди других ограничений на значения, которые могут принимать данные, важным является ограничение на обязательность заполнения определенного поля во время ввода данных. Для этого в нижней части окна на вкладке Общие следует в списке Обязательное поле выбрать Да или Нет. Если установлено Да, то в дальнейшем данные всей записи не будут сохраняться, пока не будет заполнено отмеченное поле. Такая метка является обязательной для ключевых полей.

Для установки ключевого поля следует сделать текущим соответствующее поле и выполнить *Конструктор => Сервис => Ключевое поле* или открыть контекстное меню поля и в списке команд выбрать Ключевое поле. Для отмены установления ключевого поля следует повторить описанные выше операции.

Продолжим разработку базы данных Страны мира. Создадим и сохраним под именем Страны мира третью таблицу. Создадим поля и установим типы данных, их размеры и форматы в соответствии с ранее определенными значениями (рис. 3.34).

При установлении типа данных для полей Государственный строй и Часть света одновременно осуществляется связывание таблиц. Для указанных полей в списке типов данных выбирается Мастер подстановок. После этого запускается указанный мастер — небольшая специальная программа, которая помогает пользователю задать список для подстановки в поле и установить связи между полями таблиц.

На первом шаге работы мастера предлагается выбрать один из двух видов источников данных — поле из уже существующей таблицы или запроса

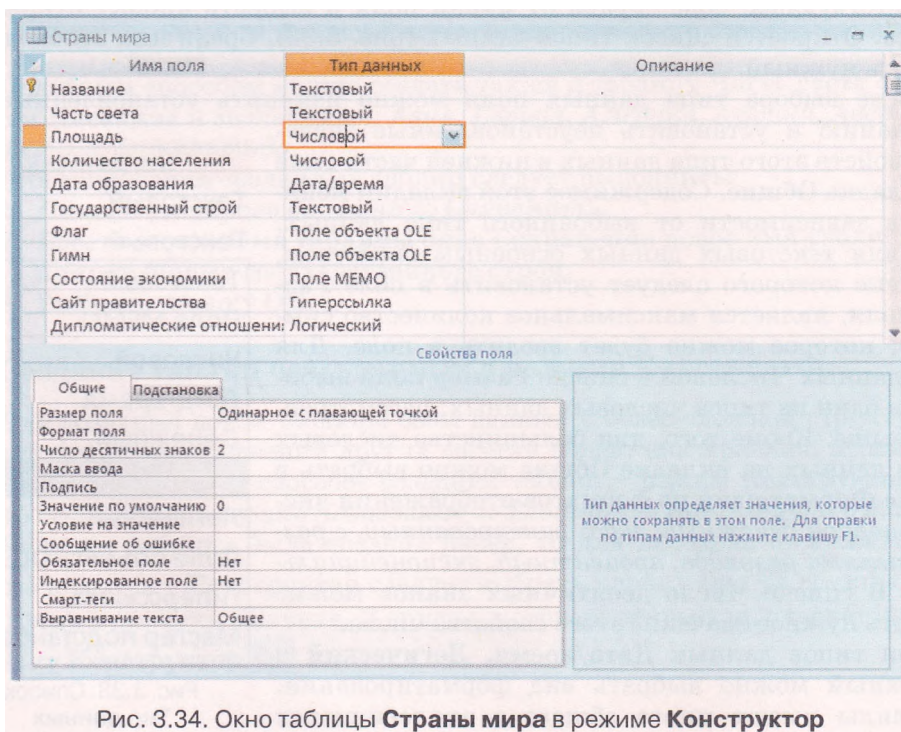



Рис. 3.34. Окно таблицы Страны мира в режиме Конструктор



базы данных, или перечень значений, которые введет пользователь на последующих шагах работы мастера. Для нашего случая мы должны выбрать первый вариант.

На втором шаге работы мастера выбирается таблица или запрос, которые содержат необходимые данные. Например, для поля Государственный строй выбирается таблица Форма государственного строя, а для поля Части света - таблица Части света.


На третьем шаге работы мастера (рис. 3.35) выбирается поле или поля, которые содержат необходимые данные. Для этого следует в списке доступных полей выбрать имя поля и кнопку . Для полей Государственный строй и Часть света выбираются поля Название только из разных таблиц.

Следующие шаги предоставляют пользователю возможность выбрать вид сортировки данных, установить ширину столбца. На последнем шаге пользователь может установить возможность выбирать несколько значений из таблицы для подстановки в поле. Для этого нужно установить метку флажка Разрешить несколько значений.

После такого связывания в полях Государственный строй и Часть света таблицы Страны мира ввод данных может происходить путем выбора значения из списка. При необходимости для дополнения этих списков достаточно ввести новые значения в поля таблицы Форма государственного строя или Часть света.

## Связи в СУБД Access 2007

Вы ознакомились с одним из способов создания связей между таблицами - с использованием Мастера подстановок.

Для просмотра уже созданных связей в базе данных следует выполнить *Работа с базами данных => Показать или скрыть => Схема данных*. Откроется окно Схема данных и для отображения в нем всех связей таблицы следует выбрать кнопку Все связи  группы Связи вкладки Конструктор временной вкладки Работа со связями (рис. 3.36). Изображение окна Схема

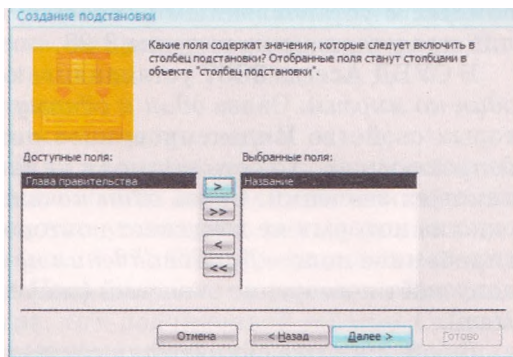


Рис. 3.35. Выбор полей, которые содержат необходимые значения, в окне **Создание подстановки**

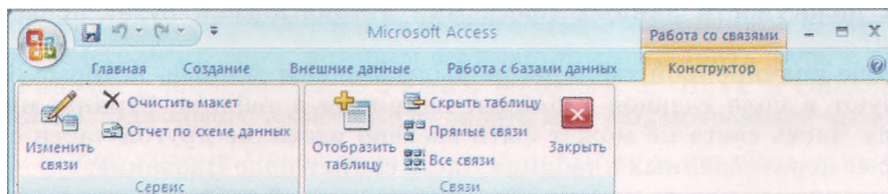


Рис. 3.36. Элементы управления вкладки **Конструктор** временной вкладки **Работа со связями**

данных с установленными связями между полями соответствующих таблиц представлено на рисунке 3.28.

В СУБД Access 2007 устанавливаются два типа связей - *один к одному* и *один ко многим*. Связь *один к одному* устанавливается между полями, в которых свойство Индексированное поле имеет значение *Да* (*Совпадения не допускаются*). То есть эти поля являются ключевыми и не допускают одинаковых значений. Связь *один ко многим* устанавливается между полями, одно из которых не допускает повторов значений (значение свойства Индексированное поле - *Да* (*Совпадения не допускаются*)), а поле другой таблицы допускает повторение значений (значение свойства Индексированное поле - *Нет*).

В большинстве случаев для создания, редактирования и удаления связей между таблицами используют окно Схема данных. Для установления новых связей необходимо в окне Схема данных выполнить такую последовательность действий:

1. Выполнить *Работа со связями => Конструктор => Связи => Отобразить таблицу*.
2. В окне Добавление таблицы выбрать название таблицы для создания связи и выбрать кнопку *Добавить*.
3. Повторить действия для отображения других таблиц, с которыми планируется связать первую таблицу.
4. Закрыть окно Добавление таблицы.
5. Выбрать в списке имен полей первой таблицы имя поля, которое будет связано, и перетянуть его на имя поля связанной таблицы.
6. Уточнить в окне Изменение связей (рис. 3.37) в списках Таблица/запрос и Связанная таблица/запрос названия полей, между которыми устанавливается связь.
7. Установить тип связи (кнопка *Объединение...*).
8. Установить при необходимости метку флажка *Обеспечение целостности данных*.
9. Выбрать кнопку *ОК*.

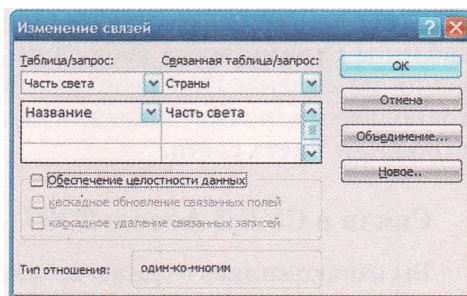


Рис. 3.37. Окно **Изменение связей**

В ходе установки свойств связи при выборе кнопки *Объединение...* открывается окно *Параметры объединения*, в котором выбором соответствующего переключателя можно установить обязательность входа в связь отдельных или всех записей из соответствующих таблиц.

При установке в окне *Редактирования связей* метки флажка *Обеспечение целостности данных* программа автоматически будет проверять, чтобы:

- в поле подчиненной таблицы не были введены данные, которые отсутствуют в поле главной таблицы, например в таблице *Страны мира* в поле *Часть света* не может быть введено название другой части света, кроме перечисленных в таблице *Часть света* в поле *Название*;
- не удалялись или не изменялись записи главной таблицы, если в подчиненной таблице есть связанные с ними записи, то есть из поля *Название* таблицы *Часть света* не может быть удалена запись *Америка*, если

в связанной с ней таблице Страны мира есть запись, которая содержит в поле Часть света значение Америка, например запись для страны Канада;

- поле главной таблицы было ключевым;
- связанные поля имели одинаковый тип данных.

Для того чтобы изменения значений в связанном поле главной таблицы приводили к автоматическому изменению во всех записях соответствующего поля подчиненной таблицы, необходимо установить метку флажка каскадное обновление связанных полей (рис. 3.37). Например, если в таблице Форма государственного строя в поле Название вместо значения Военная диктатура запишем значение Диктатура, то в подчиненной таблице Страны мира для всех записей в поле Государственный строй значение Военная диктатура заменится на Диктатура.

Если установить метку флажка каскадное удаление связанных записей, то во время удаления значения из связанного поля главной таблицы в подчиненной таблице будут удалены все записи, которые содержат это значение. Например, если в таблице Форма государственного строя в поле Название удалим значение Военная диктатура, то из таблицы Страны мира будут удалены все записи о странах, в которых в поле Государственный строй было значение Военная диктатура.

Для удаления связи между таблицами достаточно открыть контекстное меню линии связи между таблицами и выбрать команду Удалить.

## Редактирование и форматирование таблицы базы данных

В процессе создания и использования таблиц базы данных может возникнуть потребность в редактировании самой таблицы. К основным операциям редактирования структуры таблицы следует отнести: вставка и удаление полей (столбцов), переименование полей (столбцов), вставка и удаление записей (строк), перемещение полей (столбцов), изменение ключевого поля и т. п.

Для редактирования структуры таблицы используют элементы управления группы Записи вкладки Главная и группы Поля и столбцы вкладки Режим таблицы временной вкладки Работа с таблицами (рис. 3.38).

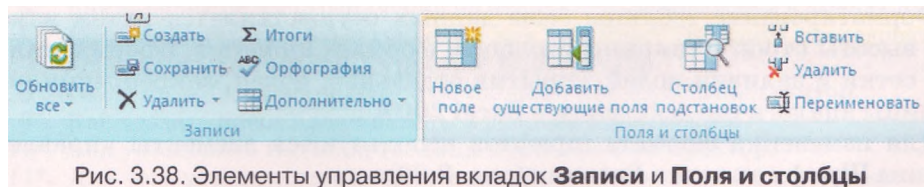


Рис. 3.38. Элементы управления вкладок **Записи** и **Поля и столбцы**

Для вставки поля (столбца) в Режиме таблицы необходимо сделать текущим поле (столбец), перед которым вы планируете вставить поле, и выполнить *Работа с таблицами => Режим таблицы => Поля и столбцы => Вставить*. Для нового поля следует задать имя рассмотренным выше способом. Выбор кнопки Новое поле группы Поля и столбцы приводит к созданию нового поля в конце списка полей.

Для удаления поля (столбца) необходимо:

1. Сделать текущим поле (столбец), которое планируется удалить.

2. Выполнить *Работа с таблицами* => *Режим таблицы* => *Поля и столбцы* => *Удалить*.
3. Подтвердить удаление поля и всех данных в нем в соответствующем окне.

Другой способ удаления полей:

1. Выделить поле выбором его имени.
2. Выполнить *Главная* => *Записи* => *Удалить*.
3. Подтвердить удаление столбца в соответствующем окне.

Можно также воспользоваться командой *Удалить столбец* контекстного меню имени поля.

Для удаления нескольких полей необходимо предварительно выделить эти поля, используя заглавия столбцов.

При удалении полей следует учитывать, что в таблице должно быть минимум одно поле. Также следует учитывать, что изменить или удалить ключевое поле в Режиме таблицы нельзя. Это можно сделать только в режиме Конструктор.

Вставка новых записей происходит путем заполнения нижней пустой строки.

*Обращаем ваше внимание.* В таблице базы данных всегда есть нижняя пустая строка. Как только ее начинают заполнять, автоматически создается новая запись, а ниже появляется новая пустая строка.

Для вставки новой записи можно выполнить *Главная* => *Записи* => *Создать*. Этот способ удобно использовать при создании новой записи в большом массиве записей.

Для удаления записи необходимо сделать текущей запись и выполнить *Главная* => *Записи* => *Удалить* => *Удалить запись*.

Для редактирования имени поля необходимо дважды щелкнуть на имени поля, которое вы собираетесь редактировать. Содержимое полей редактируется аналогично редактированию таблицы в текстовом процессоре, за исключением связанных полей, в которые данные могут вводиться путем выбора из списка и полей с логическим типом данных, в которых редактирование содержимого сводится к установке/снятию метки флажка.

Для перемещения поля необходимо выделить его и перетянуть за заглавие в нужное место.

Форматирование таблицы базы данных осуществляется путем изменения высоты строк и ширины столбцов, формата шрифтов, оформления линий сетки и заливки полей, скрытия отдельных полей, закрепления ячеек таблицы аналогично действиям в Excel 2007.

Для изменения формата шрифтов используются элементы управления группы Шрифт вкладки Главная (рис. 3.39).

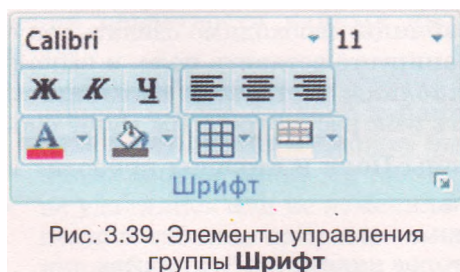


Рис. 3.39. Элементы управления группы Шрифт

Расширенные возможности форматирования таблицы базы данных предоставляют пользователю элементы управления окна *Формат таблицы* (рис. 3.40). Это окно можно открыть выбором кнопки открытия диалогового окна группы Шрифт. Дополнительно можно установить один из трех эффектов оформления таблицы (*обычный, приподнятый,*



утопленный), направление размещения полей (слева направо или справа налево), стиль линий границ таблицы и стиль линии подчеркивания заглавия (сплошная, штриховая, пунктирная, редкая точечная линия и т. п.).

Обращаем ваше внимание. Форматирование в таблице базы данных нельзя осуществлять для отдельных полей, записей, символов. Установленные значения форматирования применяются для всей таблицы.

При вводе данных возможно неправильное определение типа данных, которые вводятся, и в дальнейшем - неправильное представление этих данных.

Установленные типы и форматы полей можно изменить. Так, при вводе в поле Номер оператора числа 095 программа заменит его на 95, поскольку числовым форматом не предусмотрено наличие нуля перед числом. Если пользователь предусматривает, что номер оператора должен содержать нуль перед числом, ему следует изменить тип данных для данного поля с числового на текстовый. Для этого следует сделать текущим соответствующее поле и выполнить Работа с таблицами => Режим таблицы => Форматирование и тип данных и в списке Тип данных выбрать Текстовый.

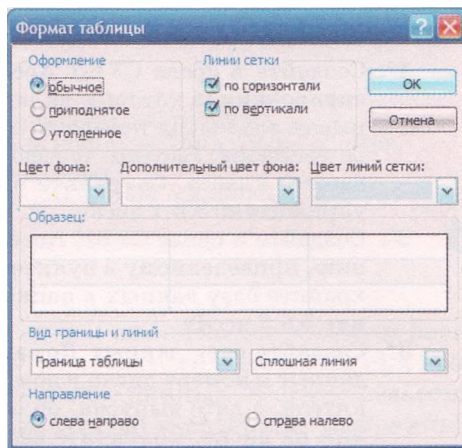


Рис. 3.40. Окно Формат таблицы

### Проверьте себя

- 1°. Объясните, в чем заключается отображение модели «сущность-связь» на базу данных. Приведите примеры.
- 2°. Опишите перечень операций, которые выполняются при отображении модели «сущность-связь» на базу данных.
- 3°. Назовите два основных способа создания таблиц базы данных.
- 4°. Какое поле создается автоматически при создании таблицы, какой тип данных этого поля?
- 5°. Опишите последовательность создания новых полей в Режиме таблицы.
- 6°. Какие существуют ограничения на использование символов в именах полей?
- 7°. Как изменить вид представления для таблицы? Опишите последовательность открытия таблицы в режиме Конструктор.
- 8°. Объясните назначение объектов таблицы в режиме Конструктор.
- 9°. Как создать новое поле таблицы в режиме Конструктор?
- 10°. Как установить тип поля таблицы в режиме Конструктор?
- 11°. Для чего устанавливается значение **Да** свойства поля Обязательное поле? В каких случаях оно используется?
- 12°. Как установить ключевое поле таблицы?
- 13°. Что значит значение 25 в строке Размер поля с типом данных Текстовый?
- 14°. Что такое Мастер подстановок? Для чего он используется?
- 15°. Опишите последовательность связывания таблиц с использованием Мастера подстановок.
- 16\*. Опишите последовательность связывания таблиц в окне Схема данных.
- 17°. Опишите основные операции редактирования таблиц базы данных.
- 18°. Как осуществляется форматирование таблицы базы данных? Объясните основные операции по форматированию таблиц.



## Выполните задания



- 1°. Создайте в среде СУБД Access 2007 таблицу базы данных Телефонный справочник, в которую включите поля для хранения данных о фамилии и имени владельца телефона, о типе (стационарный или мобильный) и номере телефона. Введите телефонные номера и другие данные пяти учеников вашего класса. Сохраните базу данных в своей папке в файле с именем упражнение 3.5.1.accdb.
- 2°. Создайте в среде СУБД Access 2007 базу данных Страны мира по описанию, приведенному в пункте учебника. Введите данные о пяти странах. Сохраните базу данных в папке Мои документы в файле с именем упражнение 3.5.2.accdb.
- 3\*. Создайте базу данных Домашняя аудиотека, которая должна содержать данные о номере диска в домашней аудиотеке, о названии диска, об исполнителе, о дате выпуска, об общей длительности музыкальных произведений на диске. Сохраните базу данных в папке Мои документы в файле с именем упражнение 3.5.3.accdb.
- 4\*. По модели «сущность-связь», представленной на рисунке 3.5, создайте базу данных Хранение товаров. Предварительно определите для этой базы данных: количество таблиц, перечень полей для каждой таблицы, какие из этих полей будут ключевыми, типы данных для каждого поля каждой из таблиц, типы связей между объектами таблиц для случая, когда есть пять видов товара, хранящихся на двух складах. К сущности Товар добавьте свойство Место хранения. Все таблицы создайте, используя режим Конструктор. Заполните таблицы данными из файла, указанного учителем, например Тема 3\Задания 3.5\образец 3.5.4.docx, в соответствии с моделью БД и заданием. Сохраните базу данных в своей папке в файле с именем упражнение 3.5.4.accdb.



## Практическая работа № 12. Разработка модели «сущность—связь» заданной предметной области. Создание базы данных в среде СУБД

**Внимание!** Во время работы с компьютером соблюдайте правила безопасности и санитарно-гигиенические нормы.

1. По заданию на создание базы данных разработайте модель «сущность-связь» и определите модель данных:

**Задание.** Создать базу данных Библиотека, состоящую из двух сущностей - сущности Единица хранения библиотеки, со свойствами: *код*, *наименование*, *автор*, *количество*, *цена*, *год издания*, *изображение обложки* и сущности Категория, с одним свойством - *название*, которое может иметь только такие значения: *книга*, *аудиокнига*, *газета*, *журнал*. Между сущностью Единица хранения библиотеки и сущностью Категория есть связь: одно значение свойства Название сущности Категория, например *аудиокнига*, может иметь много экземпляров сущности Единица хранения библиотеки.

2. Создайте необходимые таблицы базы данных, определите имена полей и их типы.
3. Установите связь между таблицами или с использованием Мастера подстановок, или с использованием инструмента Связи.
4. Введите данные в таблицу Фонд библиотеки, используя таблицу (файл таблица 12.doc) и графические файлы из папки Тема 3\Практическая 12.
5. Установите для таблицы Фонд библиотеки: цвет сетки - темно-зеленый, цвет фона - светло-желтый, альтернативный цвет фона - светло-голубой.
6. Сохраните базу данных в своей папке в файле с именем практическая работа 12.accdb.

### 3.6. Формы. Создание, редактирование и форматирование форм



1. Назовите основные объекты базы данных и их назначение.
2. Опишите последовательность создания новой таблицы базы данных.
3. Как вводятся данные в таблицах базы данных?
4. Какие виды режимов работы с таблицами базы данных вы знаете? Для чего они используются?

#### Виды форм в базе данных

Как уже упоминалось, форма является одним из основных объектов базы данных. Форма обеспечивает удобный для пользователя интерфейс при вводе, просмотре или редактировании данных в базе данных. Форма создается на основе одной или нескольких таблиц (запросов) и содержит выбранные пользователем поля из этих таблиц (запросов). Кроме полей, форма может иметь и другие объекты: рисунки, текстовые надписи, диаграммы, элементы управления - кнопки, флажки, переключатели и т. п. Данные, введенные пользователем в определенные поля формы, сохраняются в тех таблицах, на основе которых была создана форма. Кроме того, в формах можно создать поля, в которых будет проведено вычисление нового значения на основе значений из других полей. В СУБД Access 2007 выделяют несколько видов форм:

- *обычная (столбец или одна форма)* - форма, в которой поля размещаются в один или несколько столбцов с подписями полей слева от поля. На экран выводятся данные из одной записи (см. рис. 3.24);
- *таблица (таблица данных)* - форма, аналогичная объекту базы данных таблица с тем только отличием, что в форму можно включать не все поля таблицы, а также можно создавать поля, в которых выполняются вычисления на основе данных из других полей. Графические данные в форме этого вида не отображаются. На экран выводятся данные из нескольких записей;
- *несколько элементов (табличная)* - поля в этой форме размещаются в виде таблицы, однако, в отличие от предыдущего вида, эта форма имеет заглавие, в которое, как и в основное поле, можно поместить текстовые и графические объекты, элементы управления. Графические данные, содержащиеся в полях, отображаются в форме. На экран выводятся данные из нескольких записей (см. рис. 3.21);
- *выровненная* - форма, в которой поля, как и в обычной форме, размещены в столбик, только подписи полей размещаются над полем. На экран выводятся данные из одной записи;
- *разделенная* - форма, которая содержит и обычную форму, и таблицу. Просматривать и редактировать данные можно и в форме, и в таблице (рис. 3.42);
- *форма, содержащая таблицу*, - также содержит два объекта - форму и таблицу, однако, в отличие от предыдущего вида, таблица может быть не связана с формой;
- *кнопочная (модальная) форма* - форма, не содержащая данных, она содержит элементы управления для выполнения определенных действий или инициализации выполнения макросов.

Как и таблицы, формы могут быть открыты в разных представлениях:

- Режим формы;
- Режим макета;
- Конструктор;
- Режим таблицы и т. п.

Для перехода от одного режима к другому используется список кнопки Вид группы Представление вкладки Главная.

## Создание новой формы на основе таблицы базы данных

Есть несколько способов создания форм с использованием элементов управления группы Формы вкладки Создание (рис. 3.41):

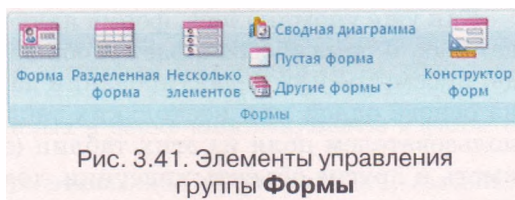


Рис. 3.41. Элементы управления группы **Формы**

- выбором кнопки **Форма** создается *обычная*, или *столбцовая, форма*;
- выбором кнопки **Разделенная форма** создается *разделенная форма* (рис. 3.42);
- выбором кнопки **Несколько элементов** создается форма *несколько элементов (табличная)*;
- выбором кнопки **Пустая форма** создается *обычная форма* путем перетаскивания полей из окна **Список полей** в область формы;
- с использованием списка кнопки **Другие формы** можно создать:
  - о разные виды форм, используя **Мастер форм**;
  - о форму **Таблица**, используя команду **Режим таблицы**;

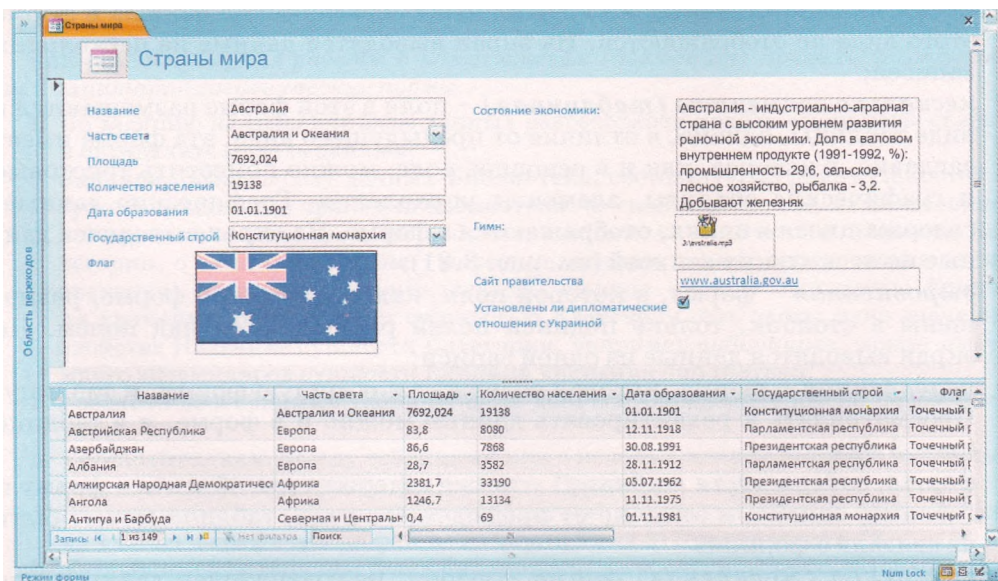




Рис. 3.42. Разделенная форма **Страны мира**, созданная на основе одноименной таблицы



о *кнопочную (модальную) форму*, используя команду **Модальное диалоговое окно**;

- выбором кнопки **Конструктор форм**  можно перейти к созданию формы произвольного вида, используя элементы управления вкладки **Конструктор временной вкладки Инструменты конструктора форм**.

При использовании кнопок управления - **Форма**, **Разделенная форма**, **Несколько элементов**, а также команды **Режим таблицы списка** **Другие формы** форма создается автоматически на основе открытого или выделенного в **Области переходов объекта**. Как правило, после создания форма открывается в представлении **Режим макета**. В этом представлении пользователь имеет возможность просмотреть вид формы, данные, которые она содержит, и внести изменения в оформление, но вносить новые данные в форму и редактировать уже существующие данные пользователь не может. Для этого следует открыть форму в представлении **Режим формы**.

Перед внесением изменений в данные необходимо сохранить форму. Для этого следует выбрать кнопку **Сохранить**  на **Панели быстрого доступа** и ввести имя формы в соответствующее поле окна **Сохранение**.

## Создание новой формы с использованием Мастера форм

Рассмотренные выше способы создания форм предусматривают, что к новой форме будут включены все поля избранной таблицы, их размещение и размеры определяются программой. Больше возможностей для выбора полей и их размещения предоставляет пользователю **Мастер форм**. Для его запуска следует выполнить *Создание => Формы => Другие формы => Мастер форм*. После этого необходимо выполнить такую последовательность действий:

1. Выбрать в списке **Таблицы и запросы** (рис. 3.43) имя таблицы или запроса, поля из которых будут включены в форму.
2. Выбрать в списке **Доступные поля** имя поля, которое планируется включить в форму.

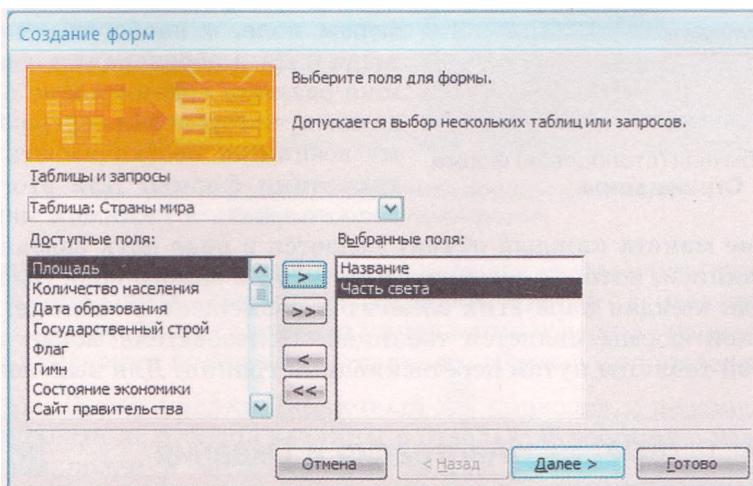



Рис. 3.43. Окно **Создание форм** на первом шаге **Мастера форм**

3. Выбрать кнопку  для перемещения имени поля в список Выбранные поля.
4. Повторить действия, описанные в п. 2 и 3, нужное количество раз для включения в список Выбранные поля всех нужных имен полей из выбранной таблицы или запроса.
5. Выбрать при необходимости в списке Таблицы и запросы имя другой таблицы или запроса и переместить описанным выше способом имена необходимых полей в список Выбранные поля.
6. Выполнить действия, описанные в п. 5, нужное количество раз.
7. Выбрать кнопку Далее.
8. Выбрать один из видов макета формы (в один столбец, ленточный, табличный, выровненный).
9. Выбрать кнопку Далее.
10. Выбрать в списке стиль оформления формы (например, Поток).
11. Выбрать кнопку Далее.
12. Ввести имя формы в поле Задайте имя формы (например, Страны мира).
13. Выбрать действие, которое будет выполняться после завершения работы мастера: Открыть форму для просмотра или ввода данных или Изменить макет формы.
14. Выбрать кнопку Готово.

## Редактирование и форматирование формы в Режиме макета

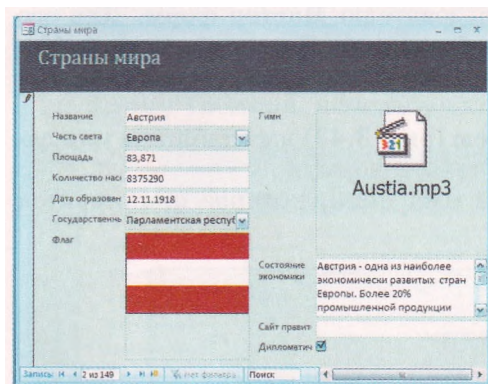


Рис. 3.44. Обычная (столбцовая) форма Страны мира

После создания новой формы, как правило, она нуждается в редактировании. Так, например, в созданной с использованием Мастера форм форме Страны мира (рис. 3.44) подписи к большинству полей закрыты другими объектами, содержимое поля Состояние экономики тяжело читать, потому что оно отображается в малом по размерам поле, и наоборот - поля Площадь и Дата образования имеют большие размеры сравнительно с объемом данных, которые они содержат. Поэтому возникает необходимость в редактировании формы. Для этого можно воспользоваться Режимом макета.

В Режиме макета каждый объект подается в виде двух связанных элементов — надписи, которая содержит имя поля, и самого поля (рис. 3.45). В свою очередь каждая пара этих элементов размещена в таблице. Каждый столбец такой формы является таблицей. Пользователь может изменить размеры этой таблицы путем перетаскивания границ. Для выделения одно-

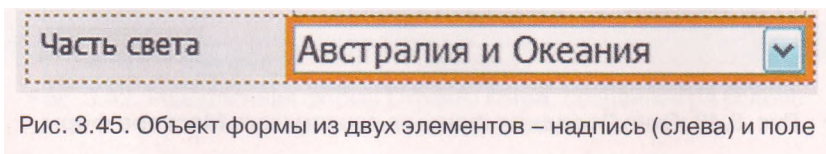



Рис. 3.45. Объект формы из двух элементов – надпись (слева) и поле



Рис. 3.46. Группа **Элементы управления**



Рис. 3.47. Группа **Автоформат**

го из объектов формы необходимо выбрать этот объект. Для выделения всех объектов в таблице следует выбрать один из объектов таблицы, а затем - маркер  в левом верхнем углу таблицы.

В Режиме макета для выделенных объектов формы, используя элементы управления группы Шрифт вкладки Формат временной вкладки Работа с макетами форм, можно изменить:

- шрифт символов, их размер, цвет, стиль и выравнивание;
- цвет заливки фона;
- условное форматирование содержимого полей.

Элементы управления группы Форматирование той же вкладки предоставляют пользователю возможность установить различные форматы числовых данных, а группы Сетка - стиль, ширину, цвет и вид сетки таблицы в форме.

Объекты группы Элементы управления вкладки Формат временной вкладки Работа с макетами форм (рис. 3.46) предназначены:

- кнопка Эмблема - вставка изображения в заглавие формы;
- кнопка Заголовок - вставка текста заглавия формы;
- кнопка Дата и время - вставка текущей даты и времени в заглавие формы;
- кнопки Толщина линии, Тип линии, Цвет линии - установление формата отображения линий контура текущего объекта формы;
- кнопка Добавить существующие поля - открытие окна со списком полей таблицы базы данных для их вставки в форму.

Используя элементы управления группы Автоформат (рис. 3.47), можно изменить тему оформления формы. Изменение происходит аналогично подобным операциям, рассмотренным в Word 2007, PowerPoint 2007.

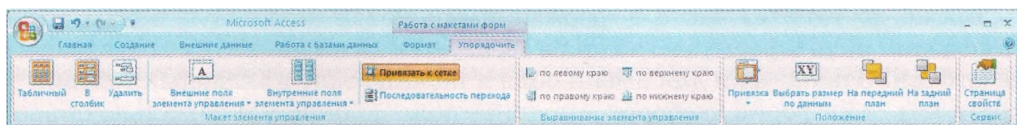


Рис. 3.48. Вкладка **Упорядочить** временной вкладки **Работа с макетами форм**



Инструменты для изменения размещения объектов формы находятся на вкладке Упорядочить временной вкладки Работа с макетами форм (рис. 3.48). Группа Макет элемента управления содержит элементы управления для изменения положения отдельного объекта или выделенной группы объектов. Выбор кнопки Табличный  приводит к перемещению выбранного объекта из столбца таблицы в отдельную таблицу с размещением надписи над полем в заглавии формы (рис. 3.49). При выборе кнопки В столбик  объект также выделяется в отдельную таблицу с размещением



Рис. 3.49

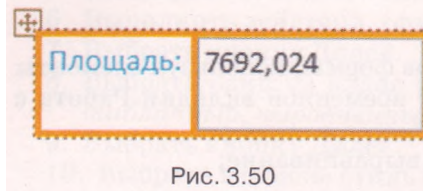






Рис. 3.50

надписи слева от поля (рис. 3.50). Этот объект можно разместить в любом месте формы, используя для перетаскивания маркер таблицы .

Кнопка Удалить  используется для удаления объекта из примененного макета формы. Объект удаляется из таблицы и размещается в произвольном месте формы. Положение объекта можно изменить перетаскиванием его границ или перемещением. При этом можно отдельно изменять размеры надписи и размеры поля.

При выборе кнопки Внешние ПОЛЯ эле-

мента управления  открывается список, в котором пользователь может установить отступы текста, чисел или рисунков от края надписи или поля. Возможные варианты: *Нет, Узкое, Среднее, Широкое*. Используя список кнопки Внутренние поля элемента управления , можно изменить расстояния между ячейками таблицы в форме.

## Редактирование формы с использованием Конструктора

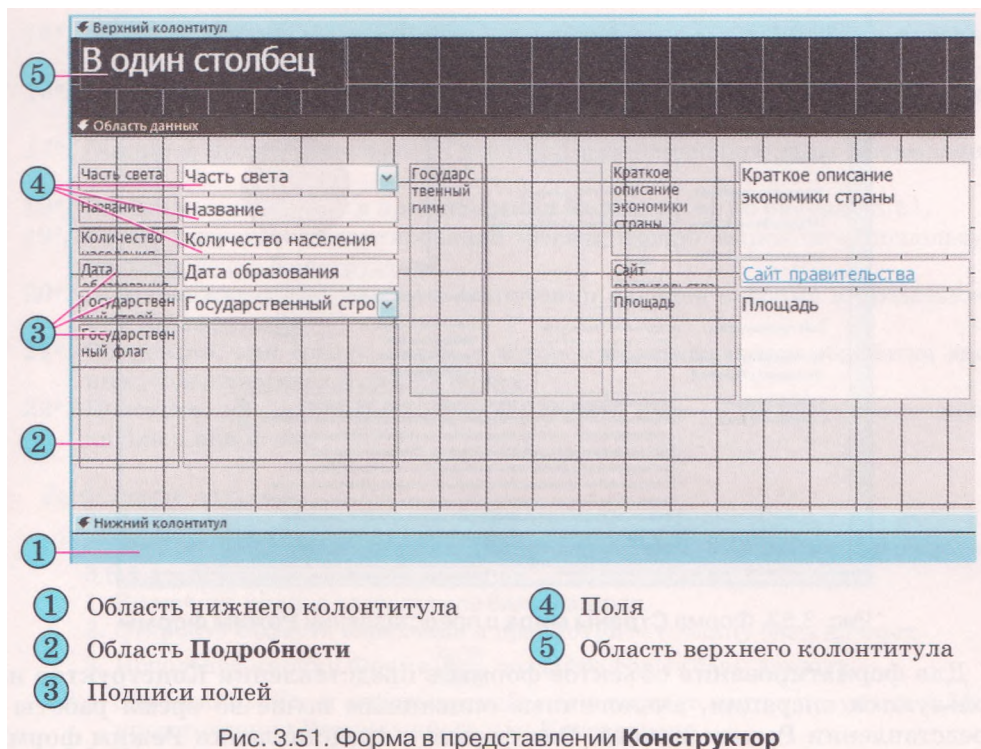
Для редактирования формы достаточно часто используют Конструктор, который предоставляет более широкие возможности изменения взаимного размещения объектов, вставки элементов управления и т. п. Переход к представлению Конструктор (рис. 3.51) осуществляется описанным выше способом.

Одной из основных операций редактирования в представлении Конструктор является изменение положения и размеров объектов - полей и подписей к ним. Осуществляется она аналогично операциям в представлении Режим макета.

Также достаточно часто в формах изменяются подписи к полям. При включении поля в форму его подпись образуется из имени поля с добавлением двоеточия. Однако если для имени поля в таблице одно из главных условий - краткость, то в форме наоборот — подпись поля должна максимально объяснять данные, которые являются содержимым поля. Так, вместо подписи Название желательно ввести Название страны, вместо Дата образования желательно ввести более точную подпись Дата образования государства или провозглашения независимости и т. д. Для редактирования содержимого подписи необходимо установить курсор во внутрь надписи и изменить текст с использованием стандартных операций редактирования текста. При этом следует заметить, что если текст подписи длинный, то его можно разместить в несколько строк. Для этого следует перетаскиванием границы увеличить высоту рамки подписи.

Для пояснения содержимого полей используют не только подписи полей, но и отдельные надписи. Например, для уточнения, в каких единицах представлена площадь стран в этой базе данных, желательно после поля вста-





вить надпись с единицами измерения. Для этого следует выполнить такую последовательность действий:

1. Выбрать кнопку **Надпись Аа** группы **Элементы управления** вкладки **Конструктор** временной вкладки **Инструменты конструктора форм**.
2. Выделить в области **Подробности** место размещения надписи.
3. Ввести в надпись текст **кв. км**.
4. Уточнить размещение надписи путем перетаскивания объекта и его границ.
5. Просмотреть результаты в представлении **Режим формы**.

Как уже упоминалось, формы могут содержать поля, в которых осуществляется вычисление на основе данных, содержащихся в других полях. Рассмотрим последовательность создания такого поля на примере создания поля **Плотность населения**, значение которого вычисляется путем деления значения поля **Количество населения** на значение поля **Площадь**. Для этого необходимо выполнить такую последовательность действий:

1. Выбрать кнопку **Поле ab** группы **Элементы управления** вкладки **Конструктор** временной вкладки **Инструменты конструктора форм**.
2. Выделить в области **Подробности** место размещения текстового поля.
3. Уточнить размещение обеих частей текстового поля путем перетаскивания объекта и его границы.
4. Ввести подпись поля **Плотность населения** вместо стандартного **Текст №**.
5. Ввести в поле формулу для подсчета плотности населения страны в виде **= [Количество населения] / [Площадь]**.
6. Просмотреть результаты в представлении **Режим формы**.

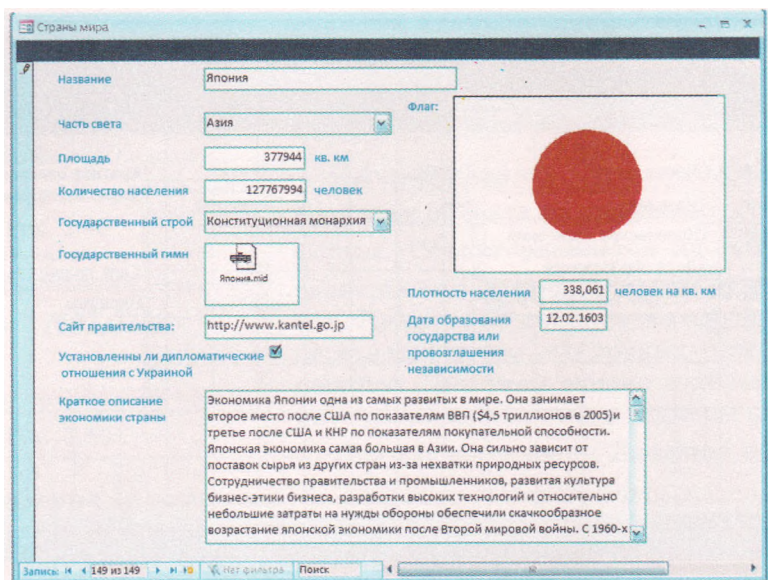





Рис. 3.52. Форма **Страны мира** в представлении **Режим формы**

Для форматирования объектов формы в представлении Конструктор используются операции, аналогичные описанным выше во время работы в представлении Режим макета. Вид формы в представлении Режим формы изображен на рисунке 3.52.


### Проверьте себя

- 1°. Для чего используются формы в базах данных?
- 2°. Укажите основные виды форм в СУБД Access 2007.
- 3°. Объясните, чем отличается таблица базы данных от формы вида *таблица*.
- 4°. Объясните, чем отличается вид формы *таблица* от вида формы *несколько элементов (табличная)*.
- 5°. Объясните, чем отличается вид формы *столбец* от *выровненная*.
- 6°. С какой целью создается форма вида *разделенная*? Опишите общий вид такой формы.
- 7°. Назовите виды представления, в которых могут открываться формы базы данных. Опишите последовательность перехода.
- 8°. Опишите последовательность создания формы с использованием кнопки  группы Формы вкладки Создание.
- 9°. Опишите последовательность сохранения созданной формы.
- 10°. Объясните, как создать форму в один столбец, на основе одной из таблиц базы данных, используя Мастер форм.
- 11°. Какие виды форм можно создать, используя Мастер форм?
- 12°. В каких представлениях можно редактировать форму?
- 13°. Опишите, как изменить положение отдельных объектов формы в представлении Режим макета.
- 14°. Для чего используется кнопка Табличный , а для чего - кнопка



В столбик  группы Макет элемента управления вкладки Упорядочить временной вкладки Работа с макетами форм?

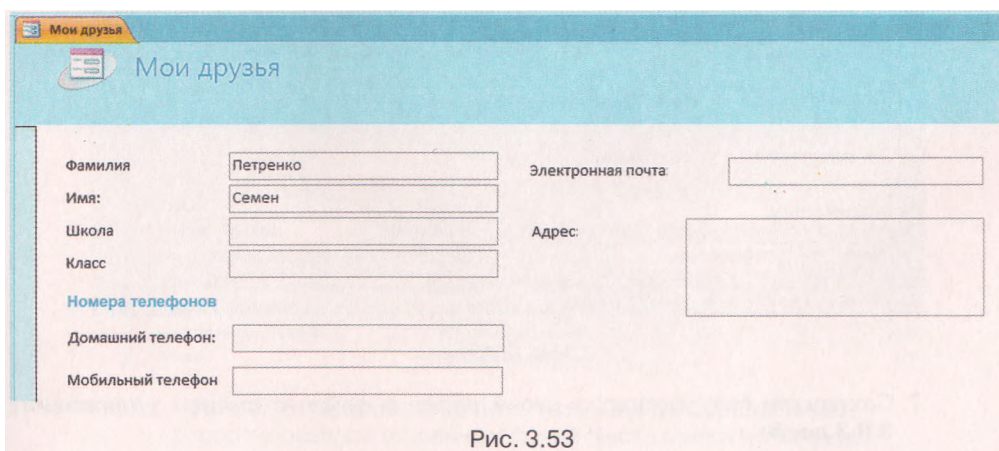
15. Объясните назначение объектов группы Элементы управления вкладки Формат временной вкладки Инструменты конструктора форм.
16. Для чего используются элементы управления группы Шрифт вкладки Формат временной вкладки Инструменты конструктора форм?
17. Опишите последовательность действий для изменения темы оформления формы.
18. Опишите окно формы в представлении Конструктор по рисунку 3.51.
19. Какие операции редактирования формы можно выполнить, используя представление Конструктор?
20. Опишите последовательность включения надписи в форму в представлении Конструктор.
- 21\*. Объясните, как создать поле, в котором осуществляется обработка данных, содержащихся в других полях.
22. Поясните, как установить отступы текста, чисел или рисунков от края надписи или поля.

### Выполните задания

- 1°. Откройте файл базы данных (например, Тема 3\Задания 3.6\образец 3.6.1.accdb) и:
  1. Включите полное содержимое базы данных.
  2. Откройте Область переходов и просмотрите таблицу базы данных.
  3. Используя кнопку Форма , создайте форму базы данных.
  4. Отредактируйте и отформатируйте ее в соответствии с рисунком 3.53 в представлении Режим макета или Конструктор.
  5. Введите, используя форму, данные в пять записей.
  6. Сохраните базу данных в своей папке в файле с именем упражнение 3.6.1.accdb.



2. Откройте файл базы данных (например, Тема 3\Задания 3.6\образец 3.6.2.accdb) и:
  1. Включите полное содержимое базы данных.
  2. Откройте Область переходов и просмотрите  таблицы базы данных.
  3. Используя кнопку Разделенная форма , создайте форму базы данных.
  4. Отредактируйте и отформатируйте ее в соответствии с рисунком 3.54 в представлении Режим макета или Конструктор.
  5. Введите, используя форму, данные в пять записей.



The screenshot shows a software interface for a form titled "Мои друзья" (My Friends). The form is displayed in a "Form View" (Режим макета) and contains several input fields for user data. The fields are arranged in two columns. The first column includes fields for "Фамилия" (Last Name) with the value "Петренко", "Имя:" (Name), "Школа" (School), "Класс" (Class), "Номера телефонов" (Phone Numbers), "Домашний телефон:" (Home Phone), and "Мобильный телефон" (Mobile Phone). The second column includes fields for "Электронная почта" (Email) and "Адрес:" (Address). The interface has a light blue header and a light pink background for the form area.

Рис. 3.53



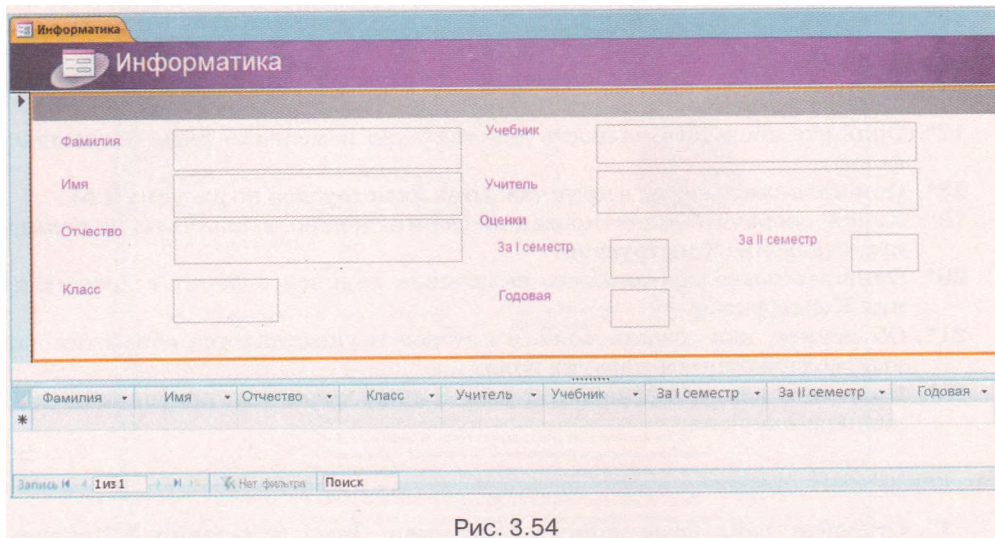


Рис. 3.54

6. Сохраните базу данных в своей папке в файле с именем упражнение 3.6.2.accdb.



3. Откройте файл базы данных (например, Тема 3\Задания 3.6\образец 3.6.3.accdb) и:

1. Включите полное содержимое базы данных.
2. Откройте Область переходов и просмотрите таблицу базы данных.
3. Просмотрите связи в базе данных, определите, какой вид связи установлен и между какими полями таблиц.
4. Используя кнопку Мастер форм, создайте форму базы данных.
5. Отредактируйте и отформатируйте ее в соответствии с рисунком 3.55 в представлении Конструктор.
6. Создайте на форме поле для подсчета суммы имеющегося товара, учитывая количество и стоимость соответствующего товара.

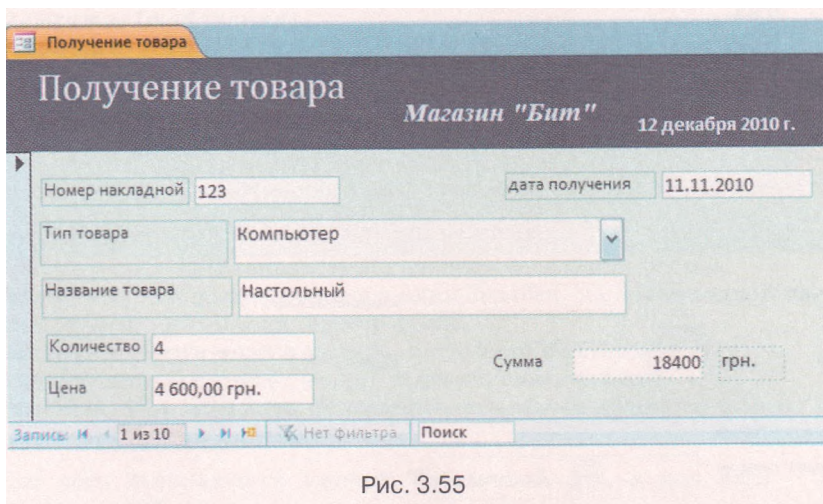


Рис. 3.55

7. Сохраните базу данных в своей папке в файле с именем упражнение 3.6.3.accdb.



### 3.7. Сортировка, поиск и фильтрация данных в базе данных




1. Что такое сортировка? Как осуществляется сортировка в таблицах Word 2007, Excel 2007?
2. Поиск каких данных можно осуществлять в Word 2007, Excel 2007?
3. Что такое фильтр в Excel 2007? Как осуществляется фильтрация?
4. Как в Excel 2007 провести фильтрацию данных, которые имеют значение больше 0, но меньше 100?

#### Сортировка данных

Как и в таблицах Word 2007 и Excel 2007, данные в таблицах баз данных можно сортировать. По умолчанию при открытии таблицы данные в Access 2007 сортируются по возрастанию значений в ключевом поле. Для изменения порядка сортировки следует выполнить такую последовательность действий:

1. Открыть таблицу базы данных, данные которой нужно отсортировать.
2. Установить курсор в пределах поля, по данным которого будет выполнена сортировка записей.
3. Выполнить *Главная => Сортировка и фильтр => По возрастанию (По убыванию)*.

Возле имени поля, по данным которого осуществлена сортировка, появляется стрелка, указывающая на вид сортировки: ↑ - по возрастанию, ↓ - по убыванию. Для того чтобы отменить сортировку, необходимо выбрать

кнопку **Очистить все сортировки**  группы **Сортировка и фильтр** вкладки **Главная**. Для сортировки по данным нескольких полей с одинаковыми значениями параметров сортировки следует выделить эти поля (выделить можно лишь соседние поля) и выполнить *Главная => Сортировка и фильтр => По возрастанию (По убыванию)*. При сортировке по данным нескольких полей сначала сортировка происходит по данным полей, размещенных слева. Фрагмент таблицы **Страны мира**, отсортированной по убыванию по данным полей **Часть света** и **Площадь**, представлен на рисунке 3.56.

Можно выполнить сортировку по данным нескольких полей, произвольно размещенных в таблице, последовательно выполнив сортировку для каждого из них. При этом можно использовать разные значения параметров сортировки.

Страны мира	Название	Часть света	Площадь	Количество населения
Катар	Азия	11	565	
Ливан	Азия	10,4	3496	
Бахрейн	Азия	0,6	640	
Мальдивская Республика	Азия	0,3	291	
Австралия	Австралия и Океания	7636	19138	
Новая Зеландия	Австралия и Океания	268,7	3778	
Соломоновы Острова	Австралия и Океания	29	447	
Тонга	Австралия и Океания	0,7	100	

Рис. 3.56. Фрагмент таблицы базы данных **Страны мира**, отсортированной по данным полей **Часть света** и **Площадь**

Сортировка данных в форме выполняется аналогично сортировке в таблице, только в форме нельзя выделить одновременно несколько полей для установления одинаковых параметров сортировки.

### Поиск данных

Как было отмечено ранее, базы данных предназначены для обеспечения быстрого доступа к необходимым данным. Достаточно часто пользователи осуществляют поиск нужных данных в базах данных: железной дороги, библиотеки, поискового сервера Интернета и др. В СУБД Access 2007 поиск в базе данных осуществляется подобно поиску в Excel 2007. Для поиска нужных данных необходимо:

1. Открыть файл базы данных, в которой нужно осуществить поиск.
2. Открыть таблицу или форму, в которой будет осуществляться поиск.
3. Выполнить *Главная => Поиск => Найти*.
4. В окне Поиск и замена (рис. 3.57) на вкладке Поиск в поле Образец ввести образец данных, по которому будет осуществлен поиск.
5. Выбрать в списке Поиск в имя поля или всей таблицы (формы), в которых будет осуществлен поиск.
6. Установить в списке Совпадение одно из значений: *Поля целиком*, *С любой частью поля*, *С начала поля*.
7. Установить в списке Просмотр одно из направлений поиска: *Все*, *Вверх*, *Вниз*.

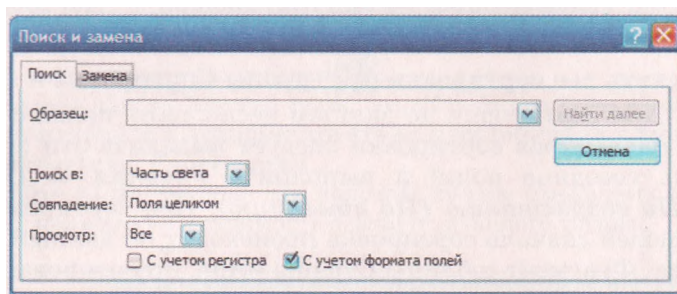


Рис. 3.57. Окно Поиск и замена

8. Установить при необходимости соответствующие метки флажков для поиска с учетом регистра и поиска с использованием шаблонов (С учетом формата полей).
9. Выбрать кнопку Найти далее.

Следует заметить, если в списке Совпадение выбрать значение *Поля целиком*, то во время поиска образец будет сравниваться с содержанием всего поля, а при выбранном значении *С любой частью поля* сравниваться с образцом будет любая часть содержимого поля и соответственно при значении *С начала поля* сравнение будет осуществлено по начальным символам поля.

Значения в списке Просмотр устанавливают направление поиска от текущей записи: *Вверх* - к первой записи, *Вниз* - к последней записи и *Все* - по всей таблице (форме).

## Замена данных

Замена данных в полях таблицы или формы выполняется с использованием элементов управления вкладки Замена окна Поиск и замена. Кроме описанных параметров для поиска, во время замены в поле Заменить на вводятся данные, которые должны заменить найденные. Замену можно осуществлять по шагам (кнопки Найти далее и Заменить) или сразу всех найденных данных (кнопка Заменить все).



### Использование подстановочных знаков для поиска данных

Во время поиска данных можно использовать специальные символы, которые в Access 2007 еще называют **подстановочными знаками**. В базах данных, созданных в Access 2007 и предыдущих версиях Access, использовались подстановочные знаки по стандарту ANSI-89 (один из стандартов языка запросов SQL):

- \* - любое количество любых символов, в том числе и их отсутствие;
- ? - любой один символ;
- [] - любые символы, указанные в квадратных скобках;
- [!] - любые символы, кроме тех, которые указаны в квадратных скобках перед восклицательным знаком;
- - любой символ из последовательности символов, например **д-к** отвечает символам от **д** до **к** в алфавите;
- # - любая одна цифра.

## Фильтрация данных

Для отбора записей, данные в которых соответствуют определенным условиям, используют фильтры. Для фильтрации данных используют элементы управления группы Сортировка и фильтр вкладки Главная. Например, для нахождения записей о странах, в которых государственный строй *монархия* или *конституционная монархия*, с использованием кнопки **Фильтр**

необходимо выполнить такую последовательность действий:

1. Сделать текущим поле, по данным которого будет осуществлена фильтрация, например *Государственный строй*.
2. Выбрать кнопку **Фильтр**.
3. Установить в списке (рис. 3.58) метки флажков возле одного или нескольких значений, которые соответствуют условию фильтрации, например *Монархия* и *Конституционная монархия*.
4. Выбрать кнопку **ОК**.

Можно осуществлять фильтрацию по данным нескольких полей. Например, для определения стран *Африки*, в которых *государственный строй монархия* или *конституционная монархия*, сле-

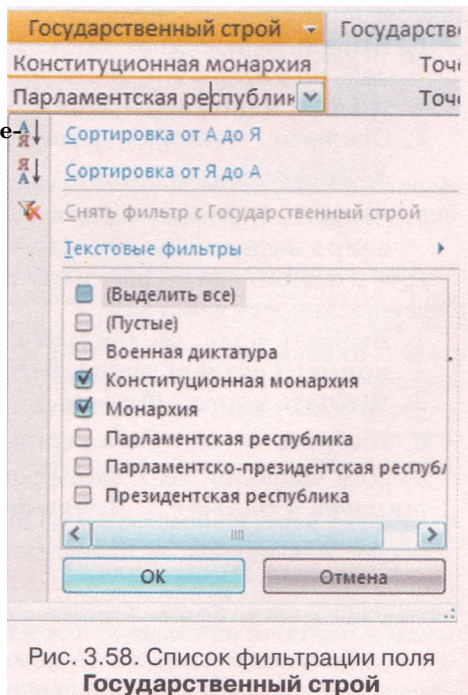


Рис. 3.58. Список фильтрации поля **Государственный строй**

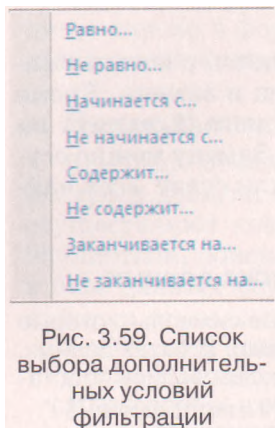


Рис. 3.59. Список выбора дополнительных условий фильтрации

дует после проведения фильтрации по данным поля Государственный строй еще провести фильтрацию по данным поля Часть света со значением параметра фильтрации *Африка*.

Количество записей, которые соответствуют условиям фильтрации, можно определить по счетчику в нижней левой части окна программы - там указан номер текущей записи и общее количество отфильтрованных записей.

Для отмены фильтрации необходимо выполнить *Главная* => *Сортировка и фильтр* => *Применить фильтр*. Для повторного применения уже определенных значений параметров фильтрации следует опять выполнить *Главная* => *Сортировка и фильтр* => *Применить фильтр*.

При проведении фильтрации с использованием кнопки *Фильтр* есть возможность задать более сложные условия фильтрации. Так, в списке фильтрации любого текстового поля при наведении указателя на надпись *Текстовые фильтры* (рис. 3.58) открывается список условий фильтрации (рис. 3.59). Выбор любого элемента этого списка открывает дополнительное окно с полем для ввода фрагмента текста, который станет составляющей соответствующего условия: *Равно*, *Не содержит*, *Заканчивается на* и т. п.

Если поле содержит числовые данные, то список условий будет другим: *Равно*, *Не равно*, *Больше*, *Меньше*, *Между*, а для данных даты и времени - *Равно*, *Не равно*, *До*, *После*, *Между*.

Для применения фильтра по выделенному необходимо выполнить такую последовательность действий:

1. Выделить фрагмент данных в одном из полей, по которым будет осуществлена фильтрация, например *Африка*.
2. Открыть список кнопки *Выделение*



(рис. 3.60).

3. Выбрать в списке одно из условий применения выделенного фрагмента для фильтрации данных.

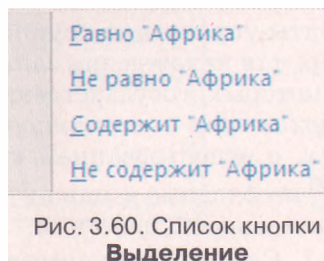


Рис. 3.60. Список кнопки *Выделение*

Для удаления фильтра определенного поля следует выполнить такую последовательность действий:

1. Выбрать поле, по данным которого была проведена фильтрация, например *Государственный строй*.
2. Выбрать кнопку *Фильтр*.
3. Выбрать кнопку *Снять фильтр с Государственный строй*

Для удаления всех фильтров необходимо выполнить *Главная* => *Сортировка и фильтр* => *Дополнительно* => *Очистить все фильтры*.

### Проверьте себя

- 1°. Что такое сортировка? Как выполнить сортировку по данным одного из полей?
- 2°. Опишите последовательность выполнения сортировки по значениям размещенных рядом полей с одинаковыми значениями параметров сортировки.



3. Опишите последовательность выполнения сортировки по значениям полей, размещенных произвольно в таблице, с разными значениями параметров сортировки.
- 4°. Как отменить сортировку?
- 5\*. Сравните сортировку в таблицах Word 2007, Excel 2007 и Access 2007. Что общего и чем отличаются эти операции в указанных прикладных программах Microsoft Office 2007?
6. Опишите последовательность выполнения поиска, например, слова **информатика**, которое содержится в начале поля; в любой части содержимого поля.
7. Можно ли осуществить поиск только в пределах одного поля, одной записи?
- 8\*. Как просмотреть все найденные в результате поиска записи?
- 9\*. Опишите последовательность осуществления замены содержимого всего поля.
- 10\*. Как осуществить замену фрагмента текста **метр** на **километр**, учитывая, что этот фрагмент может встретиться в разных частях содержимого полей? Как провести эту замену максимально быстро?
11. Для чего используются подстановочные знаки во время поиска данных?
12. Опишите использование подстановочных знаков [ ] и [-] в ходе поиска данных.
- 13\*. Сравните осуществление поиска и замены в таблицах Word 2007, Excel 2007 и Access 2007. Что общего и чем отличаются эти операции в указанных прикладных программах Microsoft Office 2007?
- 14°. Что такое фильтрация?
15. Объясните последовательность выполнения фильтрации по значениям одного из полей таблицы.
16. Объясните, как вывести записи, содержащие хотя бы одно поле, в которое не введены значения.
- 17°. Как определить количество записей, которые удовлетворяют условиям фильтрации?
- 18°. Как осуществить фильтрацию по значениям трех полей?
19. Чем отличаются условия фильтрации, которые можно установить для полей с данными типа Текст от полей с типами Число, Дата и время?
- 20°. Как отменить фильтрацию, как ее применить опять с теми же значениями параметров?
21. Сравните результаты выполнения операций поиска и фильтрации. Что у них общего и в чем отличие?



### Выполните задания

- 1°. Откройте файл базы данных (например, Тема 3\Задания 3.7\образец 3.7.1.accdb). Включите полное содержимое базы данных. Откройте таблицу Страны мира и выполните сортировку данных:
  - а) Отсортируйте записи по данным поля Количество населения **по убыванию**. Определите пять стран с наибольшим количеством населения. Определите, какое место занимает Украина в мире по количеству населения.
  - б) Выполните сортировку по данным полей Количество населения и Часть света для определения стран, которые имеют наибольшее количество населения в определенной части света. Для какого поля (полей) следует выполнить сортировку по возрастанию, а для какого (каких) - по убыванию? Сохраните базу данных в своей папке в файле с именем упражнение 3.7.1.2.accdb.
  - в) Отсортируйте записи по данным поля Государственный строй и определите, сколько стран имеют государственный строй **монархия** и сколько **конституционная монархия**.
  - г) Проведите сортировку записей по данным полей Государственный строй и Часть света для определения количества стран, которые имеют государственный строй **президентская республика** в каждой части света. Для какого поля (полей) следует выполнить сортировку по возрастанию.



нию, а для какого (каких) - по убыванию? Сохраните базу данных в папке Мои документы в файле с именем упражнение 3.7.1.4.accdb.

2. Откройте файл базы данных (например, Тема 3\Задания 3.7\образец 3.7.2.accdb). Включите полное содержимое базы данных. Откройте таблицу Страны мира и выполните поиск и замену данных:
  - а) Последовательно найдите страны, в которых денежной единицей является: *Крузадо, Кетсаль, Золотая кордоба, Доллар*. Для последней денежной единицы осуществите поиск в любой части поля.
  - б) Выполните замену значения *Северная Америка на Америка*. Сохраните базу данных в собственной папке в файле с именем упражнение 3.7.1.1.accdb.
  - в) Найдите страны, в которых в значении площади встречаются цифры **1** или **2**. Найдите страны, в которых дата образования **1960** год.
  - г) Выполните замену значения *Центральная и Южная Америка на Америка*. Сохраните базу данных в папке Мои документы в файле с именем упражнение 3.7.2.4.accdb.
3. Откройте файл базы данных (например, Тема 3\Задания 3.7\образец 3.7.3.accdb). Включите полное содержимое базы данных. Откройте таблицу Географические и физические карты и выполните фильтрацию данных:
  - а) Выберите *административные* карты, которые изготовлены *на картоне* или *на картоне с планками*. Сохраните базу данных в собственной папке в файле с именем упражнение 3.7.3.1.accdb.
  - б) Отмените все фильтры. Выберите карты *Украины* с масштабом просмотра *М1:2 100 000* и ценой больше *20 грн*. Сколько таких карт найдено? Сохраните базу данных в собственной папке в файле с именем упражнение 3.7.3.2.accdb.
  - в) Отмените все фильтры. Выберите все карты, *кроме* карт с данными об *Украине* и ценой *от 5 до 40 грн*. Сколько таких карт найдено? Сохраните базу данных в собственной папке в файле с именем упражнение 3.7.3.3.accdb.

## Практическая работа № 13. Формы. Ввод данных. Сортировка, фильтрация и поиск данных

Внимание! Во время работы с компьютером соблюдайте правила безопасности и санитарно-гигиенические нормы.

1. Откройте файл базы данных (например, Тема 3\Практическая 13\Образец 13.accdb).
2. Дополните таблицу Страны новым полем Союзы стран.
3. На основе таблицы Страны создайте форму Страны мира по образцу, приведенному на рисунке к этой практической работе. Установите тему оформления Строгая.
4. Используйте поле для создания в форме поля с вычислением плотности населения.
5. Используя форму и поиск, введите данные о вхождении некоторых стран в объединение стран: СНГ и ОПЕК (англ. *The Organization of the Petroleum Exporting Countries* - Организация стран - экспортеров нефти). Данные о вхождении стран в эти объединения возьмите из файла Союзы странное папки Тема 3\Практическая 13.
6. Откройте таблицу Страны и отсортируйте данные по возрастанию по полю Союзы стран. Проверьте, правильно ли вы ввели данные о членстве стран в названные объединения.

Страна	<input type="text" value="Австралия"/>	Государственный флаг	
Часть света	<input type="text" value="Австралия и Океания"/>		
Площадь	<input type="text" value="7836"/> тыс. кв. км	Плотность населения	<input type="text" value="2,506"/>
Количество населения	<input type="text" value="19138"/> тыс. чел.		
Государственная валюта	<input type="text" value="Австралийский доллар"/>	Сайт правительства	<input type="text" value="www.australia.gov.au"/>
Входит в объединение стран	<input type="checkbox"/>		

- Установите фильтрации данных: для поля **Площадь** - *менее 1 тыс. кв. км*, а для поля **Население** - *более 1 млн чел.*
- Сохраните базу данных в своей папке в файле с именем **практическая работа 13.accdb**.

### 3.8. Запросы в Access 2007



- Что такое формула в Excel 2007? Какие объекты может содержать формула в Excel 2007?
- Для чего используются фильтры в базах данных? Как применить фильтр к таблице базы данных в Access 2007?
- Какие операторы используются в формулах Excel 2007?
- Опишите последовательность действий для создания фильтров для нескольких полей.

#### Понятие о запросах в базе данных

Запросы являются одним из основных инструментов обработки данных в базах данных. Запросы могут обеспечивать не только поиск данных, которые соответствуют определенным критериям, как это осуществляется во время фильтрации, но и одновременное выполнение операций над данными и сохранение результатов поиска.

Основные операции, которые может осуществить пользователь с использованием запросов:

- создание новых таблиц на основе анализа данных в уже существующих таблицах базы данных;
- вычисление обобщенных данных (суммы, максимального или минимального значения и т. п.) для заданных полей;
- нахождение значений новых свойств (проведение вычислений), используя данные из разных таблиц или запросов, например нахождение плотности населения определенной страны на основе площади и количества населения;
- внесение изменений в уже существующие таблицы (обновление данных, вставка и удаление записей и т. п.).

В зависимости от назначения запросов их разделяют на:

- запросы на выборку данных - запросы, с использованием которых на основе существующих таблиц (рис. 3.61) создается таблица с данными, соответствующими определенным условиям (рис. 3.62);
- перекрестные запросы - запросы, в которых на первом этапе осуществляется обобщение данных (находится сумма, среднее, максимальное значение и т. п.), а на втором - группирование этих данных по двум наборам данных, один из которых определяет заглавия столбцов таблицы, а второй - заглавия строк.

Номер	Спортсмен	Дистанция	Результат
2	Голованов Василий	100 м	11,4
3	Петренко Степан	100 м	10,5
4	Федченко Николай	100 м	10,8
5	Хмиз Дмитрий	100 м	11,1
6	Субботин Виктор	100 м	10,9
7	Голованов Василий	1500 м	4:01,23
8	Петренко Степан	1500 м	4:02,00
9	Федченко Николай	1500 м	4:03,11
10	Хмиз Дмитрий	1500 м	4:02,45
11	Петренко Степан	400 м	54,65
12	Хмиз Дмитрий	400 м	51,4
13	Субботин Виктор	1500 м	4:03,22
14	Субботин Виктор	400 м	52,34
15	Голованов Василий	1500 м	4:01,54
16	Голованов Василий	100 м	11,6
17	Федченко Николай	400 м	55,7
18	Федченко Николай	1500 м	4:03,7
19	Голованов Василий	400 м	52,6

Рис. 3.61. Таблица с результатами выступления спортсменов на разных соревнованиях на нескольких дистанциях

Спортсмен	Дистанция	Min - Результат
Голованов Василий	100 м	11,4
Петренко Степан	100 м	10,5
Субботив Виктор	100 м	10,9
Федченко Николай	100 м	10,8
Хмиз Дмитрий	100 м	11,1

Рис. 3.62. Таблица, созданная как результат запроса на выборку лучших результатов спортсменов на дистанции 100 м

Спортсмен	100 м	1500 м	400 м
Голованов Василий	11,4	4:01,23	52,6
Петренко Степан	10,5	4:02,00	54,65
Субботив Виктор	10,9	4:03,22	52,34
Федченко Николай	10,8	4:03,11	55,7
Хмиз Дмитрий	11,1	4:02,45	51,4

Рис. 3.63. Таблица, созданная как результат перекрестного запроса на выборку лучших результатов каждого из спортсменов на трех дистанциях

Например, в таблице, созданной запросом на выборку наилучших результатов каждого из спортсменов на трех дистанциях (рис. 3.63), заглавиями столбцов стали данные из поля Дистанция, а заглавиями строк - данные из поля Спортсмен. В клеточках на пересечении столбцов и строк указан лучший (минимальный) результат определенного спортсмена на определенной дистанции;

запросы на внесение изменений — запросы, используя которые, пользователь может изменять значение в полях определенных записей, создавать новые записи или удалять существующие записи и т. п.;

запросы с параметрами - запросы, после запуска на выполнение которых пользователь должен ввести значение определенных параметров, по которым будет осуществлена обработка данных. Например, можно создать запрос, который будет создавать таблицу - список учеников класса определенной школы, данные о котором нужны пользователю в данный момент. После запуска такого запроса на выполнение откроется окно с полем, в которое пользователь должен ввести название нужного класса. В зависимости от введенного значения будет построена таблица - список учеников указанного класса. Запрос сохраняется в базе данных под определенным именем. При открытии запроса происходит обработка данных в соответствии с условиями запроса, и результаты запроса выводятся на экран в виде таблицы.

## Выражения и операторы в запросах

В запросах для записи условий отбора данных или для выполнения действий над данными используют выражения. Выражение в Access 2007 похоже на формулу в Excel 2007. Выражения могут содержать:

идентификаторы - имена полей таблиц и элементов управления в формах и отчетах;

операторы - последовательность символов для обозначения операций. Различают операторы: арифметические, сравнения, логические, объединения (сцепления), специальные;

функции - аналогичны по назначению и использованию функциям в Excel 2007, однако в Access 2007 немного другой набор функций и они сгруппированы по другим категориям (массивы, преобразование, база данных, математические, финансовые, текст и др.);

константы - неизменные значения, например фрагмент текста или число;

круглые скобки - для определения приоритета операций в выражении.


При создании выражений следует соблюдать определенные правила:

имена полей и другие идентификаторы записываются в квадратных скобках;



- при ссылке на поле определенной таблицы используется восклицательный знак, например выражение [Товары]![Код товара] содержит ссылку на поле Код товара таблицы Товары;
- текст записывается в кавычках "".

## Запросы на выборку

Рассмотрим создание запроса на выборку с использованием Мастера запросов. Для базы данных Страны мира создадим запрос, который бы содержал данные о площади стран. Для создания запроса необходимо открыть базу данных Страны мира и выбрать кнопку Мастер запроса  группы Другие вкладки Создание. Далее следует выполнить такую последовательность действий:

1. Выбрать в окне Новый запрос в списке видов мастеров запросов Простой запрос, после чего выбрать кнопку ОК.
2. Выбрать в списке Таблицы и запросы окна Создание простых запросов таблицу, на основе которой будет создан запрос, например таблицу Страны мира.
3. Переместить из списка Доступные поля в список Выбранные поля нужные имена полей, например Название, Часть света, Площадь.
4. Выбрать один из вариантов отображения записей с данными: *подробный* (без выполнения операций сведения) или *итоговый* с использованием сведения (нахождение суммы, среднего значения, минимума, максимума для определенных полей) выбором соответствующего переключателя. (Этот и следующий шаг выполняется только для записей, которые содержат числовые данные.)
5. Выбрать при необходимости кнопку Итоги и установить для каждого из полей нужные операции сведения установлением соответствующих меток флажков. Выбрать кнопку ОК.
6. Ввести имя запроса в поле Задайте имя запроса, например Площадь стран мира.
7. Выбрать вариант продолжения работы с запросом - *открыть запрос для просмотра данных* или *изменить макет запроса* выбором соответствующего переключателя.
8. Выбрать кнопку Готово.

Если вы выбрали вариант *открыть запрос для просмотра данных*, то на экран будет выведена таблица результатов запроса. Это новая таблица, которая содержит поля, выбранные при конструировании запроса.

Если же избран второй вариант - *изменить макет запроса*, то новый запрос откроется в представлении Конструктор (рис. 3.64). В этом представлении в верхней части рабочего поля размещен список полей таблицы, на основе которой был создан запрос. Количество списков с перечнем полей таблиц или запросов можно изменить, выполнив *Работа с запросами => Конструктор => Настройка запроса => Отобразить таблицу*. В окне Добавление таблицы следует выбрать название нужной таблицы и выбрать Добавить. Для закрытия списка полей таблицы следует в контекстном меню соответствующего списка выбрать Удалить таблицу.

В нижней части рабочего поля окна запроса размещена таблица конструктора. Она содержит такие строки:

- Поле, в которой отображаются имена выбранных полей;

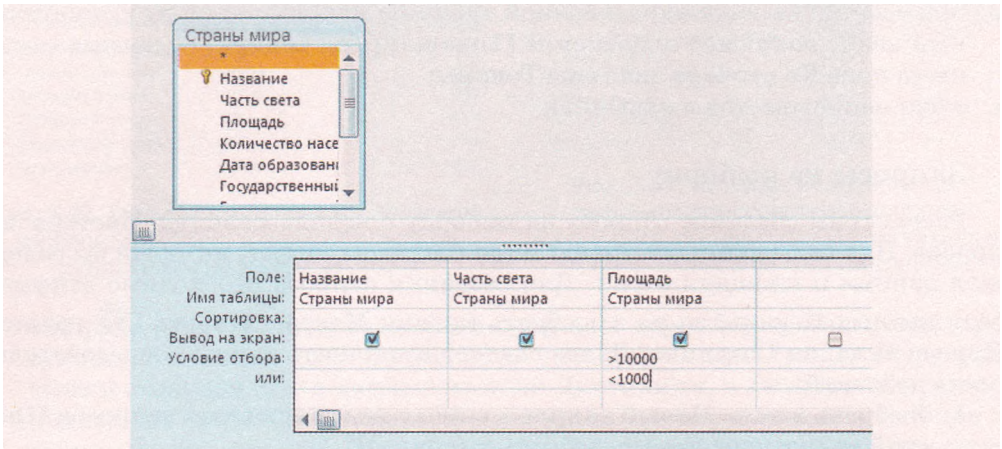


Рис. 3.64. Фрагмент окна запроса **Площадь стран мира** в представлении **Конструктор**

- **Имя таблицы**, в которой отображаются имена таблиц, в состав которых входят соответствующие поля;  
**Сортировка**, в которой может отображаться вид сортировки, который будет применен к записям данного поля в таблице запроса;  
**Вывод на экран**, содержащая флажок, установленная метка которого свидетельствует о том, что данное поле будет отображаться в запросе;
- **Условие отбора**, которая может содержать условие - выражение для выборки данных из соответствующего поля;  
или - для записи еще одного выражения, которое является частью составного условия и соединяется с выражением в строке **Условие отбора** логической операцией **ИЛИ**.

Следующие строки могут содержать другие выражения, которые связаны с выражениями в строках **Условие отбора** и **или** логической операцией **ИЛИ**.

Если в строке **Условие отбора** записаны выражения для нескольких столбцов (полей), то они объединяются в одно выражение логической операцией **И**.

Название	Часть света	Площадь
Австралия	Австралия и Океания	7636,0
Новая Зеландия	Австралия и Океания	268,7
Соломоновы Острова	Австралия и Океания	29,0
Тонга	Австралия и Океания	0,7
Итого		7934,4

Рис. 3.65. Результат выполнения запроса **Площадь стран мира** для части света **Австралия и Океания**

Для применения избранных значений параметров запроса необходимо перейти к **Режиму таблицы**. Для этого нужно выполнить **Работа с запросами** => **Конструктор** => **Результаты** => **Выполнить**.

К полученному запросу можно добавить строку с итогами. Для этого следует выполнить **Главная** => **Записи** =>

**Итоги**. И в строке **Итого**, например, поля **Площадь** открыть список возможных вариантов обобщения данных и выбрать, например, **Сумма**. Созданная таблица запроса представлена на рисунке 3.65.

### Запросы с полями, содержащими данные, вычисленные по данным из других полей

Достаточно часто возникает потребность использовать данные из двух или больше полей для получения нового значения. Например, по данным

Название	Часть света	Площадь	Количество населения	Плотность населения
Австралия	Австралия и Океания	7692,024	22510123	2926
Австрия	Европа	83,871	8356707	99638
Азербайджан	Европа	86,6	9047000	104469
Албания	Европа	28,7	3187009	111046
Алжир	Африка	2381,7	35200000	14779

Рис. 3.66. Таблица, содержащая результат запроса с полем **Плотность населения**, данные которого вычислены на основании данных из других полей

полей **Количество населения** и **Площадь** таблицы **Страны мира** определить плотность населения для каждой из стран. Для этого создадим простой запрос на выборку, в который из таблицы **Страны мира** включим поля **Название**, **Часть света**, **Площадь** и **Количество населения**. Назовем запрос **Плотность населения**. Для создания поля **Плотность населения** необходимо открыть созданный запрос в представлении **Конструктор**.

В представлении **Конструктор** в строке **Поле** следует в столбце, следующем за столбцами с данными, ввести **Плотность населения**: [**Количество населения**]/[**Площадь**], где:


- **Плотность населения** — имя нового поля (так называемого вычисляемого поля);
- **Количество населения** и **Площадь** - имена полей, значения которых используются в выражении;
- / - оператор деления и округления до целого.

Результат запроса с полем, которое содержит данные, вычисленные по данным из других полей, представлен на рисунке 3.66.

### Запросы с параметрами

Параметр запроса - изменяемая часть условия отбора, значение которой пользователь вводит в начале выполнения запроса. Используются параметры как для запросов на выборку, так и для перекрестных запросов, или запросов на внесение изменений.

Например, при создании запроса на нахождение площади стран определенной части света, указывать сразу в условиях отбора определенную часть света не рационально - при нахождении площади стран другой части света необходимо будет создавать новый запрос или редактировать существующий, изменяя условия отбора. Этого можно избежать, если использовать запрос с параметром. Для его создания следует открыть запрос в представлении **Конструктор** и выполнить такую последовательность действий:

1. Сделать текущей строку **Условие отбора** столбца с именем поля, для которого будет вводиться значение параметра.
2. Ввести в квадратных скобках текст, который будет отображаться в диалоговом окне, открываемом в начале выполнения запроса, например [**Название части света**].
3. Ввести при необходимости текст другого сообщения для ввода значения другого параметра этого же поля или других полей.
4. Выбрать кнопку **Выполнить** 

Для продолжения работы с запросом следует ввести в поле диалогового окна **Введите значение параметра** (рис. 3.67) значение нужного параметра, например **Африка**.

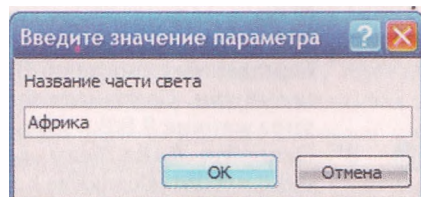


Рис. 3.67. Окно **Введите значение параметра**

Для перехода к просмотру данных о странах другой части света необходимо закрыть таблицу результатов выполнения запроса и запустить его на выполнение опять и ввести другое значение параметра - название другой части света.

При создании запросов с параметрами можно использовать составные условия. Приведем несколько примеров таких условий с их описанием. Выражение **>=[Наименьшее число] And <=[Наибольшее число]** можно использовать для выведения записей, которые в выбранном поле будут содержать значения из указанного интервала чисел от **Наименьшее число** до **Наибольшее число**. При запуске запроса пользователь сначала должен будет в поле диалогового окна ввести наименьшее число, после этого во втором диалоговом окне - наибольшее число.

Для создания запросов с параметром по значениям нескольких полей можно в представлении **Конструктор** в строке **Критерии** для каждого из таких полей ввести соответствующие выражения. Например, в поле **Пол** ввести выражение **[Укажите пол]**, а в поле **Год рождения** - **[Укажите год рождения]**. Такой запрос с параметрами можно использовать для выведения данных о лицах определенного пола и определенного года рождения, например *юношей 1998 года рождения*.

### Проверьте себя

- 1°. Для чего предназначены запросы?
- 2°. Назовите основные операции, которые выполняются с использованием запросов в базах данных.
- 3°. Опишите назначение основных видов запросов.
- 4°. Приведите примеры использования запросов на выборку.
- 5°. Чем отличается запрос на выборку от перекрестного запроса?
- 6\*. Сравните запрос с параметрами в базе данных и запрос на поиск данных в поисковой системе Интернета. Что у них общего и что отличного?
- 7°. Для чего в запросах используют выражения?
- 8°. Из чего могут состоять выражения?
- 9°. Назовите известные вам операторы, которые используются в выражениях для создания запросов.
- 10°. Опишите последовательность создания запроса на выборку с использованием Мастера запросов.
- 11°. Что является результатом выполнения запроса на выборку?
- 12°. Опишите структуру окна запроса в представлении Конструктор.
- 13°. Объясните последовательность создания запроса с использованием полей, которые содержат данные, вычисленные по данным из других полей.
- 14°. Для чего используются запросы с параметрами?

### Выполните задания

1. Составьте схему классификации запросов.
2. Откройте файл базы данных (например, Тема 3\Задания 3.8\образец 3.8.2.accdb). Создайте с использованием мастера Простых запросов запрос на выборку данных из таблицы Географические и исторические карты. Запрос должен включать данные из полей Страна, Регион, Вид карты, Размер, Цена. В представлении Конструктор установите такие условия отбора данных: по полю Вид карты - *Политическая*-, по полю Цена - *меньше 45 грн*. Сохраните базу данных в собственной папке в файле с именем упражнение 3.8.2.accdb.
3. Откройте файл базы данных (например, Тема 3\Задания 3.8\образец 3.8.3.accdb). Создайте с использованием мастера Простых запросов запрос на выборку данных из таблицы Географические и исторические карты. Запрос должен включать данные из полей Страна, Регион, Вид карты, Вид оформления, Цена. В представлении Конструктор установите такие усло-



вия отбора данных: по полю Вид карты - *Иллюстрированная* или *Обзорная*; по полю Цена - *от 10 до 35 грн*. Сохраните базу данных в папке Мои документы в файле с именем упражнение 3.8.3.accdb.

4. Откройте файл базы данных (например, Тема 3\Задания 3.8\образец 3.8.4.accdb). Создайте с использованием мастера Простых запросов запрос на выборку данных из таблицы Страны мира. Запрос должен включать данные из полей Название, Часть света, Дата образования, Государственный строй, Государственный флаг, Государственный гимн. В представлении Конструктор установите такие условия отбора данных: по полю Дата образования - *после 1990 года*; по полю Государственный строй - *Парламентская республика*. Сохраните базу данных в собственной папке в файле с именем упражнение 3.8.4.accdb.
5. Откройте файл базы данных (например, Тема 3\Задания 3.8\образец 3.8.5.accdb). Создайте с использованием мастера Простых запросов запрос на выборку данных из таблицы Получение товара. Запрос должен включать данные из полей Тип товара, Название товара, Цена, Количество. В представлении Конструктор дополните запрос новым полем Сумма, вычислив в нем данные на основе значения полей Цена и Количество. Сохраните базу данных в собственной папке в файле с именем упражнение 3.8.5.accdb.
6. Откройте файл базы данных (например, Тема 3\Задания 3.8\образец 3.8.6.accdb). Создайте с использованием мастера Простых запросов запрос на выборку данных из таблицы Электронные средства для учебы в школе. Запрос должен включать данные из полей Название, Тип программы, Цена, Количество, Гриф МОН. В представлении Конструктор дополните запрос новым полем Сумма, вычислив в нем сумму на основе значений полей Цена и Количество. Предусмотрите выведение в результате запроса данных только о тех средствах, которые имеют гриф МОН. Сохраните базу данных в папке Мои документы в файле с именем упражнение 3.8.6.accdb.

### 3.9. Отчеты. Создание отчетов в Access 2007



1. Как напечатать документ в **Word 2007, Excel 2007**?
2. Опишите последовательность создания запросов в **Access 2007**.
3. Как в запросе создать поле с данными, которые вычисляются по данным из других полей? В каких случаях такие поля используются?
4. Опишите возможности редактирования форм в представлении **Конструктор СУБД Access 2007**.

#### Понятие об отчетах в базе данных

Отчеты в БД предназначены, как правило, для выведения на печать результатов обработки данных. Отчет - это готовый к печати электронный документ. Отчеты могут быть использованы для заполнения бланков документов, например сертификатов о прохождении внешнего независимого оценивания. При создании отчета могут быть использованы данные из разных таблиц и запросов, средства обобщения, сортировки и группирования данных, можно провести операции над данными полей, используя данные из других полей и встроенные функции. Также в отчет можно включать надписи, которые поясняют представленные данные, диаграммы и графики, рисунки и т. п.

На первом этапе создания отчета следует разработать его проект как документ, который будет напечатан или передан в электронном виде, определить источники данных, которые будут включены в отчет.

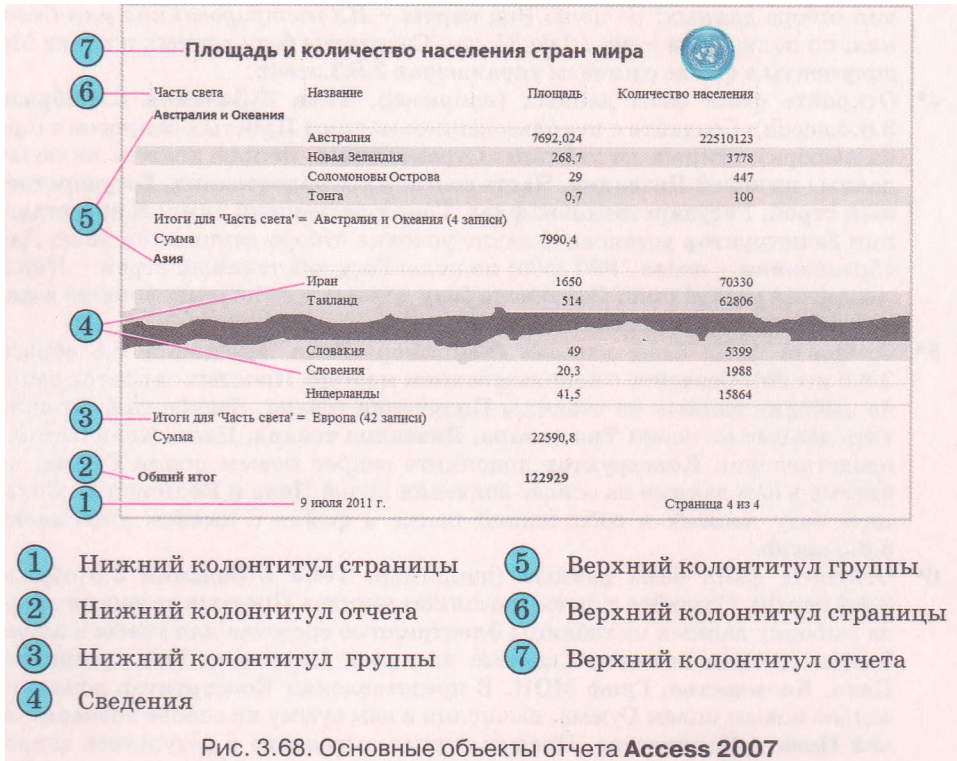


Рис. 3.68. Основные объекты отчета Access 2007



Рассмотрим структуру отчета на примере отчета **Площадь и количество населения стран мира** (рис. 3.68), который создан на основе таблицы **Страны мира**. Этот отчет содержит:

- верхний колонтитул отчета (рис. 3.68, 7), в котором размещено название отчета. Он может содержать графические объекты (логотип, эмблема, иллюстрация и т. п.), текущие дату и время, элементы управления и др. Верхний колонтитул отчета размещается только в начале отчета перед колонтитулом страницы и больше не повторяется;
- верхний колонтитул страницы (рис. 3.68, 6), в котором размещены заглавия столбцов отчета. Этот объект повторяется в верхней части каждой следующей страницы отчета аналогично колонтитулам страницы в Word 2007;
- верхний колонтитул группы (рис. 3.68, 5), в котором размещено название группы объектов. Размещается перед началом данных каждой группы. Например, для отчета, представленного на рисунке 3.68, в начале списка с данными о странах определенной части света выведено заглавие - название этой части света. Может содержать элементы управления для обобщения данных соответствующей группы объектов;
- сведения (рис. 3.68, 4) - в этой части размещены данные о каждом объекте в соответствии с полями, которые были включены в отчет;
- нижний колонтитул группы (рис. 3.68, 3), который размещается после данных обо всех объектах группы и может содержать обобщенные данные о группе объектов: количество объектов, сумму по определенному полю и т. п.;

- нижний колонтитул отчета (рис. 3.68, 2) размещается на последней странице отчета после всех данных отчета и используется для вывода обобщенных данных (количество, сумма, наибольший или наименьший по значению объект группы и т. п.);
- нижний колонтитул страницы (рис. 3.68, 1) размещается внизу каждой страницы отчета, может содержать данные о номере текущей страницы отчета и общем количестве страниц отчета, текущие дату и время и другие служебные данные.

## Создание отчетов

Access 2007 предоставляет пользователю несколько средств для создания отчетов. Доступ к этим средствам обеспечивается с использованием элементов управления группы Отчеты вкладки Создание (рис. 3.69). Самое

простое средство создания отчетов - с использованием кнопки Отчет . Для создания отчета достаточно открыть или сделать текущей таблицу или запрос, данные из которых будут включены в отчет, и выбрать кнопку Отчет . Будет создан отчет в виде таблицы со всеми данными, что содержит таблица (запрос).

Более широкие возможности по созданию отчетов предоставляет пользователю Мастер отчетов. Рассмотрим последовательность создания отчета на примере, приведенном на рисунке 3.68.

Для запуска Мастера отчетов следует выполнить *Создание => Отчеты, => Мастер отчетов*. Далее следует придерживаться указаний программы и выполнить такую последовательность действий:

1. Выбрать в окне Создание отчетов в списке Таблицы и запросы нужный объект, на основе которого будет создан отчет, например таблицу Страны мира. Выбрать, переместив в список Выбранные поля, нужные для создания отчета поля, например поля Название, Часть света, Площадь, Количество населения.
2. Выбрать при необходимости другую таблицу или запрос и выбрать нужные поля. Выбрать кнопку Далее.
3. Выбрать при необходимости поле, по данным которого будет осуществлено группирование объектов в отчете, например Часть света, и выбрать кнопку Далее.
4. Установить при необходимости порядок сортировки объектов в отчете (в пределах групп).
5. Установить при необходимости выбором кнопки Итоги и установлением меток флажков виды обобщения данных для соответствующих полей. Например, параметр Sum для поля Площадь.

*Обращаем ваше внимание:* при выборе переключателя данные и итоги в группе Показать окна Итоги (рис. 3.70) отчет будет содержать данные обо всех объектах избранной таблицы или запроса и обобщенные данные по определенному полю (полям), например по полю Площадь, а при выборе переключателя только итоги - только обобщенные данные для определенной группы.

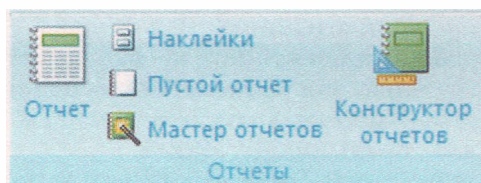
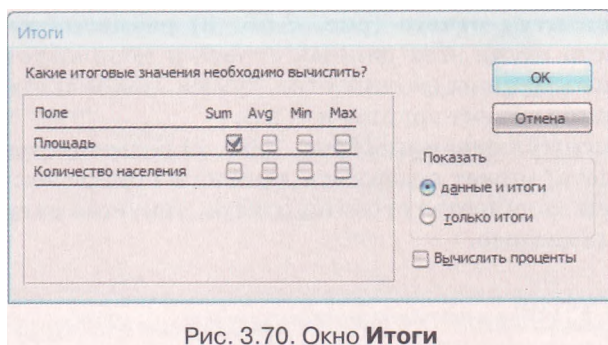


Рис. 3.69. Элементы управления группы Отчеты

Рис. 3.70. Окно **Итоги**

6. Установить при необходимости метку флажка **Вычислить процент** и выбрать кнопку **ОК**.
7. Выбрать кнопку **Далее**.
8. Выбрать макет будущего отчета выбором соответствующего переключателя в группе **Макет**, например **Ступенчатый**. Установить выбором соответствующего переключателя в группе **Ориентация** вид ориентации страницы.
9. Установить при необходимости метку флажка **Настроить ширину полей**, для размещения на одной странице. Выбрать кнопку **Далее**.
10. Выбрать в соответствующем списке стиль оформления отчета и выбрать кнопку **Далее**.
11. Ввести название отчета в поле **Задайте имя отчета**, например **Площадь** и количество населения стран мира. Выбрать соответствующим переключателем следующее действие после создания отчета, например **Просмотреть отчет**.
12. Выбрать кнопку **Готово**.

После завершения создания отчета по описанному алгоритму на экран будет выведен отчет в режиме предварительного просмотра перед печатью. А на Ленте появятся элементы управления временной вкладки **Предварительный просмотр** (рис. 3.71). Эти элементы управления предоставляют пользователю возможность: изменить макет отчета, то есть изменить размеры страницы, ее ориентацию, размеры полей страницы, количество столбцов и т. п.; изменить масштаб просмотра отчета; экспортировать отчет в другой документ — документ Word, текстовый документ и т. п.



Если в отчет не нужно вносить изменения, кроме тех, которые можно сделать с использованием элементов управления вкладки **Предварительный просмотр**, то можно сразу напечатать документ. Для этого следует выбрать кнопку **Печать** . Однако, как правило, отчет необходимо отредактировать и для этого следует сначала закрыть окно **Предварительного просмотра** выбором соответствующей кнопки .

Рис. 3.71. Элементы управления временной вкладки **Предварительный просмотр**



## Редактирование и форматирование отчетов

Для редактирования отчетов в Access 2007 нужно использовать один из двух видов представления - Режим макета или Конструктор.

Для открытия отчета в Режиме макета выполним Главная ⇒ Представление ⇒ Вид ⇒ Режим макета.

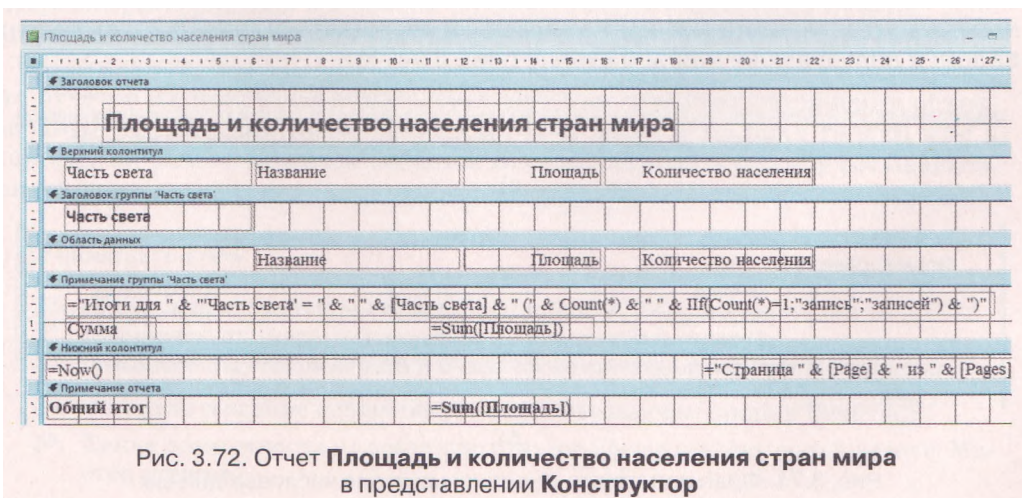
Для изменения названия столбца необходимо дважды щелкнуть на соответствующей клеточке и отредактировать текст, например ввести Название страны. Таким же путем можно добавить к названиям столбцов Площадь и Количество населения единицы измерения кв. км и человек соответственно. Но выведение наименований таким способом не совсем удобно при просмотре длинного отчета. Желательно поместить обозначение единиц измерения возле числа. Однако при работе в Режиме макета это сделать невозможно. В этих случаях редактирование лучше осуществить в представлении Конструктор.

Для открытия отчета в представлении Конструктор следует выполнить Главная ⇒ Представление ⇒ Вид ⇒ Конструктор. Вид окна отчета в представлении Конструктор представлен на рисунке 3.72.

При редактировании колонтитула отчета можно дополнить его надписями или рисунками. Например, заглавие отчета *Площадь и количество населения стран мира* можно дополнить надписью с текстом: *Разработано на основе данных компьютерной энциклопедии <http://uk.wikipedia.org>*. Для вставки этой надписи с текстом как подзаглавия отчета необходимо выполнить такие действия:

1. Увеличить размеры поля верхнего колонтитула отчета путем перетаскивания его нижней границы.
2. Выполнить *Инструменты конструктора отчетов* ⇒ *Конструктор* ⇒ *Элементы управления* ⇒ *Надпись*.
3. Указать на поле верхнего колонтитула отчета место размещения надписи.
4. Ввести текст.
5. Просмотреть полученный результат в Представлении отчета.

Добавить графический объект в верхний колонтитул отчета можно несколькими способами, используя разные элементы управления одноименной вкладки: Эмблема, Рисунок, Свободная рамка объекта, Присоединен-



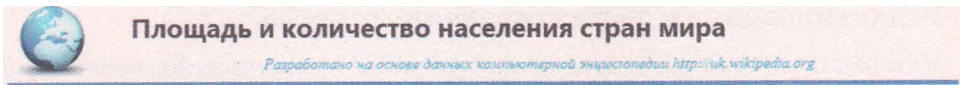




Рис. 3.73. Заглавие отчета (верхний колонтитул отчета) после редактирования

ная рамка объекта. Так, для добавления графического объекта с использованием элемента управления Эмблема следует:

1. Выбрать кнопку Эмблема .
2. Выбрать в окне Выбор рисунка файл рисунка, который будет вставлен в верхний колонтитул отчета, например GLOBE.WMF, и выбрать кнопку ОК.
3. Разместить вставленное изображение в нужном месте, используя перетаскивание самого рисунка и надписей заглавия и подзаглавия или их границ.
4. Просмотреть полученный результат в режиме Представление отчета.

Вставка рисунка с использованием других инструментов почти не отличается от описанного, за исключением того, что при вставке нужно указать место размещения рисунка, а при вставке как объекта - еще указать и вид объекта. Можно также отделить заглавие - то есть верхний колонтитул отчета от других объектов отчета горизонтальной линией.

Для этого необходимо выбрать кнопку Линия  и нарисовать линию внизу верхнего колонтитула отчета. Цвет, тип и толщину линий можно установить соответствующими элементами управления группы Элементы управления. Результат редактирования верхнего колонтитула представлен на рисунке 3.73.

Вставка текстовых и графических объектов в нижний колонтитул отчета происходит по аналогичным алгоритмам. При вставке в колонтитулы номера страницы и названия группы эти объекты будут повторяться соответственно на каждой странице или в начале, или в конце данных каждой из групп.

Часть отчета Сведения (рис. 3.68, 4) в представлении Конструктор называется Область данных (рис. 3.72). При вставке объектов в эту часть они будут повторены для каждой строки данных. Например, если вставить возле полей Площадь и Количество населения надписи с названиями единиц измерения этих величин (*кв. км* и *человек* соответственно) (рис. 3.74, а), то эти надписи повторятся в каждой строке (рис. 3.74, б).

а)

Часть света	Название	Площадь	Количество населения
Австралия и Океания			
	Австралия	7692,024 кв. км	22510123 человек
	Новая Зеландия	268680 кв. км	4377000 человек
	Соломоновы Острова	28896 кв. км	478000 человек
	Тонга	747 кв. км	120898 человек

б)

Рис. 3.74. Фрагмент отчета с надписями возле числовых данных

Как вы уже знаете, для ввода формул, подсчитывающих определенные значения, используют элементы управления, которые называют поля. Рассмотрим использование формулы для нахождения количества объектов в каждой группе стран. Заменяем малопонятную для пользователя надпись из нижнего колонтитула группы (рис. 3.71) на другую. Для этого следует открыть отчет в представлении Конструктор, удалить из нижнего колонтитула группы указанный объект и выполнить следующую последовательность действий:

1. Выполнить *Инструменты конструктора отчетов => Конструктор => Элементы управления => Поле.*
2. Выбрать на поле нижнего колонтитула место для текстового поля.
3. Установить путем перетаскивания границ длину и ширину левой и правой частей текстового поля.
4. Ввести в левую часть текстового поля текст, например *Количество стран.*
5. Ввести в правую часть поля выражение  $=\text{Количество}([\text{Название}])$ .
6. Отформатировать при необходимости тексты в соответствии со стилем оформления отчета.
7. Просмотреть полученный результат в режиме Представление отчета.

В соответствии с введенным выражением будет осуществлен подсчет количества объектов в столбце Название для каждой группы стран, то есть для каждой части света. В этом самом колонтитуле можно заменить подпись обобщения данных *Сумма*, предложенную программой автоматически, на *Общая площадь*. После редактирования нижний колонтитул группы будет иметь вид, представленный на рисунке 3.75.


Камбоджа	181040 кв. км	14805500 человек
Узбекистан	447400 кв. км	28095900 человек
Количество стран	35	
Общая площадь	27706588	

Рис. 3.75. Вид нижнего колонтитула группы после выполнения редактирования

Форматирование отчетов, как правило, осуществляется в Режиме макета. Для этого используются элементы управления вкладок *Формат*, *Упорядочить* и *Параметры страницы* временной вкладки *Работа с макетами отчетов*. Операции с использованием этих инструментов аналогичны описанным ранее.

Сохраняются отчеты автоматически при создании. После редактирования отчетов их можно сохранить стандартным способом сохранения объектов СУБД Access 2007. Изменения, которые вносятся в поля таблиц или в запросы, на основе которых созданы отчеты, автоматически учитываются при открытии отчетов.

### Проверьте себя

- 1°. Объясните назначение отчетов базы данных.
- 2°. Приведите примеры использования отчетов базы данных.
- 3°. Опишите структуру отчета и назначение его отдельных составляющих.
- 4°. Опишите создание нового отчета с использованием кнопки *Отчет* .
- 5°. Какие возможности по созданию отчетов предоставляет пользователю Мастер отчетов?

6. Объясните этапы создания отчета с использованием Мастера отчета.
7. Для чего используются при создании отчетов Итоги?
8. Опишите возможности, которые предоставляет пользователю по подготовке к печати режим Предварительный просмотр.
9. Когда возникает необходимость в редактировании и форматировании отчетов?
10. Какие объекты можно отредактировать и отформатировать в верхнем колонтитуле отчета? Как это выполнить?
11. Опишите особенности редактирования и форматирования верхнего колонтитула страницы и верхнего колонтитула группы.
12. Опишите особенности редактирования и форматирования нижнего колонтитула страницы и нижнего колонтитула группы. Какие объекты, как правило, размещаются в этих частях отчета?
13. Как включить в отчет графические объекты, где они могут размещаться?
14. Какие объекты, как правило, размещаются в Области данных представления Конструктор? Почему?
15. С какой целью используются выражения в отчетах базы данных? В каких областях отчета (представление Конструктор) они размещаются?



## Выполните задания

1. Откройте файл базы данных (например, Тема 3\Задания 3.9\образец 3.9.1.accdb). Создайте, используя кнопку Отчет, отчет по данным таблицы Получение товара. Отредактируйте и отформатируйте его в соответствии с рисунком 3.76. Эмблему вставьте из файла Тема 3\Задания 3.9\optdisk.wmf. Сохраните базу данных в собственной папке в файле с именем упражнение 3.9.1.accdb.

Номер накладной	Дата получения	Название товара	Тип товара	Количество	Цена
123	11.11.2011	19"	Монитор	3	1 567,00 грн.
123	11.11.2011	Настольный	Компьютер	14	4 600,00 грн.
123	11.11.2011	Планшетный	Сканер	4	437,00 грн.
124	21.11.2011	Ноутбук	Компьютер	5	3 656,00 грн.
124	21.11.2011	Цифровой	Фотоаппарат	6	1 273,00 грн.
125	23.11.2011	Лазерный	Принтер	12	954,00 грн.
125	23.11.2011	Настольный	Компьютер	7	4 398,00 грн.
125	23.11.2011	Струйный	Принтер	5	673,00 грн.
126	27.11.2011	22"	Монитор	11	1 895,00 грн.
126	27.11.2011	Нетбук	Компьютер	10	3 100,00 грн.


Страница 1 из 1

Рис. 3.76



2. Откройте файл базы данных (например, Тема 3\Задания 3.9\образец 3.9.2.accdb). Создайте, используя Мастер отчетов, отчет Площадь и количество населения стран мира в соответствии с описанием, представленным в тексте пункта. Выполните соответствующее редактирование и форматирование отчета в представлениях Конструктор и Режим макета. Эмблему вставьте из файла Тема 3\Задания 3.9\global.wmf. Сохраните базу данных в папке Мои документы в файле с именем упражнение 3.9.2.accdb.





### Географические и исторические карты

Вид карты	Размер	Масштаб	Вид оформления	Цена
<b>Автономная Республика Крым</b>				
Административная	42 x 30 см	M1:850 000	ламинация	15,00 грн.
<b>Европа</b>				
Политическая	160 x110 см	M1:3 750 000	ламинация	90,00 грн.
Политическая	160 x110 см	M1:3 750 000	бумага	24,00 грн.
Политическая	110x77 см	M1:5 400 000	ламинация с планками	72,00 грн.
Политическая	110x77 см	M1:5 400 000	ламинация	60,00 грн.
Политическая	110x77 см	M1:5 400 000	картон с планками	48,00 грн.
Политическая	110x77 см	M1:5 400 000	картон	30,00 грн.
Политическая	65 x 45 см	M1:10 000 000	ламинация	24,00 грн.
Политическая	65 x 45 см	M1:10 000 000	картон	9,00 грн.
Политическая	160 x110 см	M1:3 750 000	ламинация с планками	120,00 грн.
Физическая	65 x 45 см	M1:11 000 000	ламинация	24,00 грн.
Физическая	65 x 45 см	M1:11 000 000	бумага	4,50 грн.
<b>Молдова</b>				
Политическая	100 x 68 см	M1:35 000	бумага	1,00 грн.

Рис. 3.77

- Откройте файл базы данных (например, Тема 3\Задания 3.9\образец 3.9.3.acddb). Создайте с использованием Мастера отчетов отчет по данным таблицы Географические и исторические карты. Отредактируйте и отформатируйте его в соответствии с рисунком 3.77. Эмблему вставьте из файла Тема 3\Задания 3.9\worldmap.wmf. Сохраните базу данных в собственной папке в файле с именем упражнение 3.9.3.acddb.
- Откройте файл базы данных (например, Тема 3\Задания 3.9\образец 3.9.4.acddb). Создайте отчет Электронные средства для обучения в школе в соответствии с рисунком 3.78. Группировку проведите по полю класс. Дополните отчет подсчетом количества средств обучения по каждому классу в нижнем колонтитуле группы. Сохраните базу данных в папке Мои документы в файле с именем упражнение 3.9.4.acddb.

### Электронные средства для обучения в школе

Учебный предмет	Название	Цена
6 класс география	Наша планета	30,00 грн.
	Украина и её регионы	30,00 грн.
7 класс география	География материков и океанов, 7кл.	140,00 грн.
	География материков и океанов, 7кл.	30,00 грн.
10-11 класс география	Экономическая и социальная география мира, 10-11 кл.	140,00 грн.
	Экономическая и социальная география мира, 10-11 кл.	30,00 грн.

Рис. 3.78

### Практическая работа № 14. Создание запросов и отчетов

Внимание! Во время работы с компьютером соблюдайте правила безопасности и санитарно-гигиенические нормы.

1. Откройте файл базы данных (например, Тема 3\Практическая 14\образец 14.accdb).
2. На основе данных таблицы Калорийность блюд создайте запрос, включив в него поля: Блюда, Калорийность, Включено в меню.
3. В представлении Конструктор установите для запроса условие отбора по полю Включено в меню - не включать записи, которые имеют в этом поле значение *Не включено*. Выражение должно иметь такой вид <>"Не включено".
4. Создайте на основе созданного вами запроса отчет согласно образцу, представленному на рисунке 3.79.
5. В колоннитуле отчета включите выведение текущей даты в полном формате.
6. Группирование осуществите по полю Включено в меню.
7. Сумму значений найдите для каждой группы по полю Калорийность.
8. Отредактируйте и отформатируйте отчет.
9. Закройте созданные запрос и отчет.
10. Сохраните базу данных в своей папке в файле с именем практическая работа 14.accdb.

Меню столовой на 13 декабря 2011 г.		
Блюдо	Калорийность	Масса
<b>Завтрак</b>		
Винегрет	60 ккал	200 г
Блины с сыром и сметаной	650 ккал	209 г
Кофе с молоком	150 ккал	200 г
<b>Общая энергетическая ценность блюд</b>	<b>860 ккал</b>	
<b>Обед</b>		
Салат зеленый с огурцами и помидорами	100 ккал	150 г
Борщ, овощной суп	46 ккал	100 г
Рыба, тушенная с овощами	220 ккал	200 г
Картофель вареный	250 ккал	209 г
Соки ягодные, фруктовые, томатный	160 ккал	200 г
<b>Общая энергетическая ценность блюд</b>	<b>776 ккал</b>	
<b>Ужин</b>		
Икра баклажанная	175 ккал	100 г
Курица отварная	130 ккал	100 г
Макаронные изделия	330 ккал	209 г
Чай с сахаром	50 ккал	200 г
<b>Общая энергетическая ценность блюд</b>	<b>685 ккал</b>	
13 декабря 2011 г.		Страница 1 из 1

Рис. 3.79

# ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ПЕРСОНАЛЬНОЙ И КОЛЛЕКТИВНОЙ КОММУНИКАЦИИ

В этом разделе вы узнаете о:

- мультимедийных данных и программном обеспечении для их обработки;
- создании аудиофрагментов и видеоклипов;
- публикациях, компьютерных публикациях, их классификации и назначении;
- средствах создания компьютерных публикаций;
- особенностях работы с объектами при создании компьютерных публикаций;
- разновидностях веб-страниц и веб-сайтов, их структуре, этапах создания;
- средствах автоматизированной разработки сайтов и создании сайтов с их использованием;
- основах веб-дизайна;
- технологиях и сервисах **Веб 2.0**, их назначении и видах;
- создании блогов, их оформлении и публикации сообщений на блогах;
- назначении и особенностях функционирования веб-сообществ;
- интегрированном использовании средств обработки электронных документов;
- совместной работе с документами.

### 4.1. Понятие о мультимедийных данных и их обработке



1. Какие устройства для воспроизведения мультимедийных данных вы знаете?
2. Что такое тип (формат) файла? Какие форматы файлов вы знаете?
3. Что такое сжатие данных? В каких форматах файлов используется сжатие?
4. Какие типы оптических дисков вы знаете? Чем они отличаются друг от друга?
5. Что такое конвертация форматов файлов?

#### Понятие о мультимедийных данных

Как вы уже знаете из курса 9-го класса, для воспроизведения мультимедийных данных используются специальные устройства компьютера. К этим устройствам относятся звуковые колонки, микрофоны, головные телефоны, мультимедийные проекторы, мультимедийные доски и т. п. Однако нам следует уточнить, какие данные называют мультимедийными.



*Мультимедийными данными* называют данные, для представления которых используют разные способы и которые человек воспринимает одновременно несколькими органами чувств.

Примером объекта, содержащего мультимедийные данные, есть компьютерная презентация. Презентация может содержать текст, графические изображения, звуковые и видеообъекты. Поэтому очень часто такие презентации называют мультимедийными. Мультимедийные данные мы получаем, когда просматриваем видеофильмы, спектакли, посещаем концерты.



## Программное обеспечение для обработки мультимедийных данных

Компьютерных программ, обрабатывающих мультимедийные данные, достаточно много. С некоторыми из них вы уже ознакомились в 9-х и 10-х классах. Например, текстовый процессор Word 2007 обрабатывает текстовые и графические данные, документы, созданные с его использованием, могут содержать аудио- и видеоданные. Вместе с тем, когда говорят о программах обработки мультимедийных данных, то чаще всего под ними понимают программы, которые обрабатывают звуковые и видеоданные. Классификация программ для обработки звуковых и видеоданных по назначению представлена на рисунке 4.1.

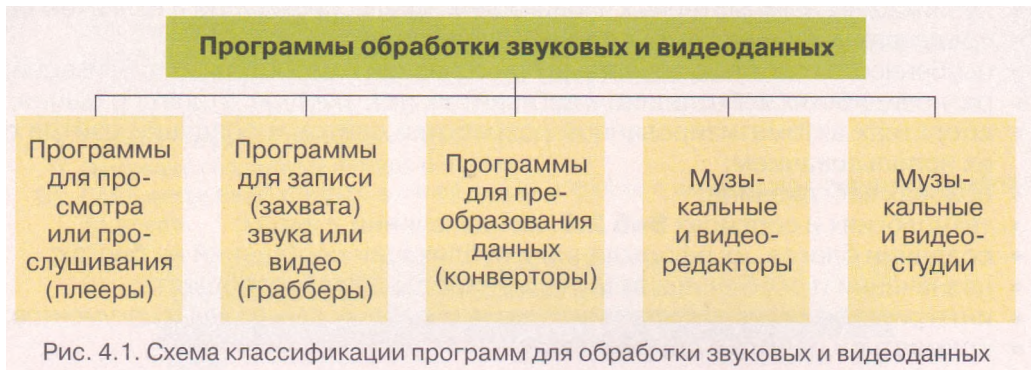


Рис. 4.1. Схема классификации программ для обработки звуковых и видеоданных

*Программы для просмотра или прослушивания мультимедийных данных* предназначены для воспроизведения видео и звука. Эти программы еще называют мультимедийными проигрывателями (плеерами). Существует много разнообразных мультимедийных проигрывателей. Большинство из них может воспроизводить и видео, и звуковые данные, некоторые специализируются только на одном из видов данных, и тогда их называют аудио- или видеоплеерами. Распространенными являются мультимедийные плееры Winamp (англ. *Windows Advanced Media Player* - расширенный проигрыватель мультимедиа для Windows), Media Player Classic, QuickTime Player, Flv Player и др.

Часто к компьютеру подсоединяют дополнительные устройства для ввода звука и видео. К таким устройствам относятся: микрофон, видекамера, веб-камера, телевизионный тюнер, DVD-проигрыватель, видеомэгнитофон и т. п. Как правило, эти устройства обеспечивают ввод звука и видео, но не имеют функций для их сохранения в виде звуковых и видеофайлов определенных форматов.

Для сохранения звуковых и видеоданных, которые вводятся с указанных устройств, предназначены специальные *программы записи (захвата) звука и видео*, которые еще называют грабберами (англ. *grabber* - тот, кто захватывает, хапуга). Эти программы во время воспроизведения, например, видеофильма с DVD-проигрывателя одновременно осуществляют перекодирование данных и их запись в видеофайл выбранного пользователем формата. Примерами программ для захвата звука являются стандартная программа ОС Windows Звукзапись, Exact Audio Copy, Audio Grabber 3.0, Streamripper, AML Easy Audio Recorder и др. Программами для захвата



видео являются Extra Screen Capture Pro, Wondershare Streaming Video Recorder, Wondershare P2P TV Recorder и др.

В связи с большим количеством разнообразных форматов аудио- и видео-файлов пользователям часто приходится выполнять перекодирование этих файлов в форматы, которые могут быть воспроизведены на тех или иных технических устройствах или теми или иными программами. Для перекодирования файлов из одного формата в другой предназначены *программы конвертации файлов*, такие как: Easy Video Converter, MediaCoder, Xilisoft Video Converter, Flash to Video Encoder, Movavi VideoSuite, Direct MIDI to MP3 Converter, AVS Audio Converter и др.

Программы *музыкальных и видеоредакторов* предназначены для создания и редактирования музыкальных произведений, аудиокниг, звуковых и видеоэффектов, звукового сопровождения, видеофильмов и т. п. Они обеспечивают выполнение таких функций: вставка, удаление, копирование, изменение длительности аудио- или видеофрагментов, объединение нескольких фрагментов, вставка текстовых и графических объектов, синхронизация звука и видео и т. п. Музыкальными редакторами являются такие программы: Anvil Studio, ALO Audio Editor, AKRAM Audio Editor и др. Примерами видеоредакторов являются стандартная программа Windows Movie Maker, Xilisoft Video Editor, VideoMach и др.



*Музыкальные и видеостудии* имеют приблизительно такое же назначение, что и редакторы, однако имеют расширенный набор функций и используются в профессиональной деятельности на студиях звукозаписи, кино- и анимационных студиях для создания звукозаписей, аудио- и видеокомпозиций, анимации, кино- и видеофильмов, рекламных роликов, видеоклипов и т. п. Примерами программ-музыкальных студий являются: Steinberg Cubase, Sound Forge, Cakewalk Sonar, Adobe Audition и др., а программам-видеостудий - Pinnacle Systems STUDIO, Apple Final Cut, Adobe Premiere Pro, Sony Vegas Movie Studio и др.















## Форматы аудио- и видеофайлов

В 9-м классе вы рассматривали понятие типа (формата) файла. Тип (формат) файла определяет структуру данных в файле. Вы уже знаете разные форматы текстовых и графических файлов. Мультимедийные данные могут быть сохранены в файлах разных форматов.

Примеры форматов мультимедийных файлов представлены в таблице 4.1.

Таблица 4.1. Типы (форматы) мультимедийных файлов

Тип (формат) файла, его описание	Расширение имени	Значок Windows	
		XP	Vista
<i>Файлы, в которых не используется сжатие или используется сжатие без потери данных</i>			
WAV или WAVE (англ. <i>waveform audio format</i> – аудиоформат типа <i>волна</i> ) – формат звуковых файлов, используемый в качестве основного для несжатых аудиоданных в операционных системах семейства Windows	wav		

Тип (формат) файла, его описание	Расширение имени	Значок Windows	
		XP	Vista
MIDI (англ. <i>Musical Instrument Digital Interface</i> – цифровой интерфейс музыкальных инструментов) – формат файлов, которые содержат команды для воспроизведения звука специальным устройством или программой – синтезатором (секвенсером) (лат. <i>sequentia</i> – последовательность)	mid, midi		
<b>Файлы, в которых используется сжатие с частичной потерей данных</b>			
MP3, точнее MPEG-1/2/2.5 Layer 3 (англ. <i>Motion Picture Experts Group</i> – экспертная группа по вопросам подвижных изображений, <i>Layer 3</i> – третий уровень), – один из самых распространенных форматов аудиофайлов. Используется для передачи звуковых данных в глобальных сетях, в файлообменных системах	mp3		
AVI (англ. <i>Audio Video Interleave</i> – звук и изображение, размещенные в слоях (каналах)) – формат файлов, который чаще используется для сохранения видеоданных. В одном файле этого формата могут быть использованы несколько разных алгоритмов сжатия	avi		
WMA/WMV (англ. <i>Windows Media Audio/Video</i> ) – формат файлов для хранения и воспроизведения аудио-/видеоданных, который разработан корпорацией Microsoft и стандартно используется в ОС Windows	wma, wmv		
3GP (англ. <i>3rd Generation Phone</i> – третье поколение телефонов) – формат файлов для передачи по сети мобильных телефонов третьего поколения. Высокая степень сжатия предусматривает малые размеры файлов и невысокое качество воспроизведения	3gp		
QuickTime (англ. <i>Quick Time</i> – быстрое время) – формат файлов, базирующийся на технологии сохранения и воспроизведения видео- и звуковых данных, которая была разработана корпорацией Apple	mov		
Flash видео – формат файлов для хранения и воспроизведения файлов, созданных программой Adobe Flash или другой, которая обрабатывает файлы этого типа. Отмечаются высокой степенью сжатия данных	flv		

В 9-м классе вы ознакомились с оптическими дисками, которые используются в компьютерах для сохранения данных. Однако сначала оптические диски (компакт-диски) были разработаны корпорациями Philips и Sony для хранения аудиозаписей. В 1979 г. был создан специальный формат записи этих данных, известный как «красная книга» (англ. *Red Book* – красная книга), а сами диски, записанные в этом формате, стали называться Audio CD (англ. *Audio Compact Disk* – звуковой компакт-диск). Звуковые данные записываются на дисках в виде отдельных участков на спиральной дорожке –







треков (англ. *track* - след, дорога, звуковая дорожка). Один трек содержит один звуковой фрагмент, например одну песню.

Для воспроизведения звуковых данных использовались специальные CD-проигрыватели. В последующем были разработаны форматы и устройства для записи и воспроизведения видеодисков - Video CD (англ. *Video Compact Disk* - видео компакт-диск), а с появлением оптических дисков DVD были разработаны специальные форматы этих дисков для сохранения аудио- и видеоданных.

Невзирая на то что диски отмеченных форматов предназначены для воспроизведения специальными проигрывателями, большинство современных компьютерных устройств для работы с оптическими дисками могут обрабатывать аудио- и видеодиски. Примеры форматов этих дисков представлены в таблице 4.2.

Таблица 4.2. Типы (форматы) мультимедийных дисков

Тип (формат) диска, его описание	Обозначение
<p><b>Audio CD</b> – формат, используемый для записи музыкальных произведений на компакт-дисках.</p> <p>Минимальная длительность звучания одного трека – 2 с, максимальное количество треков – 99, максимальное время звучания всех произведений (треков) – 74 мин</p>	
<p><b>Video CD</b> – формат, используемый для записи видеофильмов на компакт-диски. Максимальное время проигрывания видеофильма – 74 мин низкого качества</p>	
<p><b>DVD-Audio</b> – формат, используемый для записи музыкальных произведений на DVD-диски.</p> <p>Звук, записанный в этом формате, имеет лучшее качество за счет использования нескольких каналов записи данных</p>	
<p><b>DVD-Video</b> – формат записи видеофильмов на DVD-диски. Максимальное время проигрывания видеофильма – 120 мин высокого качества</p>	

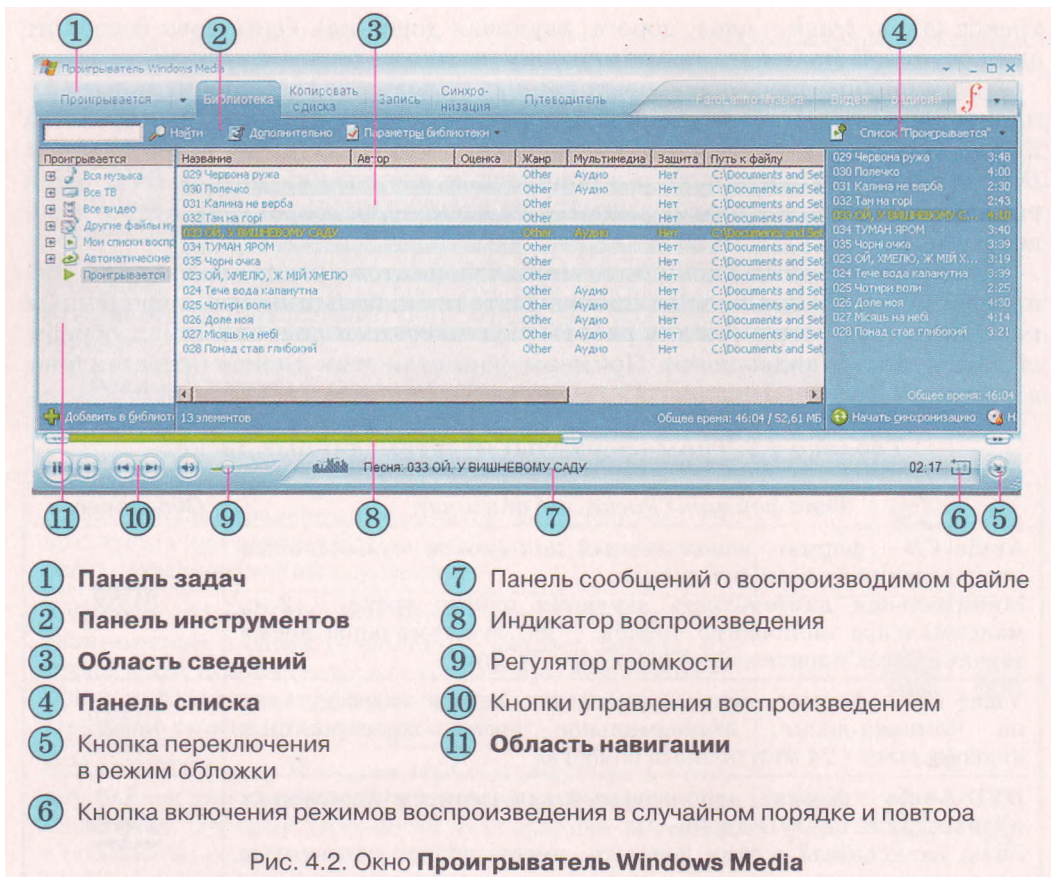
В связи с тем что большинство аудио- и видеоданных сжимаются с использованием разнообразных алгоритмов, для их воспроизведения нужно применять специальные программы - кодировщики/декодировщики, или кодеки. Кодеки входят в состав программ-проигрывателей аудио- или видеофайлов, а также могут выпускаться как дополнительный набор программ, например K-Lite Codec.

### Проигрыватель Windows Media

Одним из мультимедийных проигрывателей является программа Проигрыватель Windows Media, которая входит в состав стандартных программ операционной системы Windows. На рисунке 4.2 изображено окно этой программы версии 10.0.

Проигрыватель Windows Media предоставляет пользователю такие возможности в обработке мультимедийных (медиа) файлов:

- осуществлять поиск и включение в библиотеку (медиа-теку) медиафайлов;
- прослушивать и просматривать медиафайлы разных форматов;



- 1 Панель задач
- 2 Панель инструментов
- 3 Область сведений
- 4 Панель списка
- 5 Кнопка переключения в режим обложки
- 6 Кнопка включения режимов воспроизведения в случайном порядке и повтора
- 7 Панель сообщений о воспроизводимом файле
- 8 Индикатор воспроизведения
- 9 Регулятор громкости
- 10 Кнопки управления воспроизведением
- 11 Область навигации

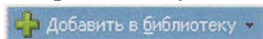
Рис. 4.2. Окно **Прогрыватель Windows Media**

- осуществлять конвертацию мультимедийных данных с аудио- и видеооптических дисков в разные форматы мультимедиафайлов;
  - создавать аудио- и видеооптические диски путем перекодирования мультимедийных файлов и записи их на диски в соответствии с отмеченными в таблице 4.2 форматами;
- загружать и воспроизводить мультимедийные данные с сайтов в Интернете (*он-лайн хранилищ*), предоставляющих услуги по хранению медиафайлов и доступа к ним зарегистрированных пользователей;
- синхронизировать медиафайлы на компьютере и на выбранном портативном устройстве, например флэш-плеере и др.

Программа осуществляет просмотр папок и внесение найденных мультимедийных файлов в медиатеку. По умолчанию автоматический просмотр проводится в папках Мои видеозаписи и Моя музыка. Для каждого из аудиофайлов в медиатеку программы заносятся данные об исполнителе, названии альбома, названии песни, жанре и годе создания. Для видеофайлов - данные об актерах и жанре видео.

Для изменения папок, в которых осуществляется поиск мультимедийных файлов, нужно:

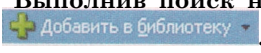
1. Выбрать внизу Области навигации кнопку Добавить в библиотеку






2. Выбрать в списке кнопки команду Просмотр папки.
3. Выбрать в открывшемся окне Просмотр папок кнопку Добавить.
4. Указать папку, в которой будет осуществляться автоматический поиск файлов мультимедиа.
5. Выбрать кнопку ОК.
6. Повторить пункты 3-5 нужное количество раз.
7. Выбрать кнопку ОК.

В дальнейшем программа будет автоматически отслеживать наличие файлов в указанных папках и добавлять новые в библиотеку.

Для проведения поиска файлов мультимедиа на компьютере пользователь может использовать команду **Выполнив поиск** на компьютере из списка кнопки **Добавить в библиотеку** . В зависимости от установленных параметров будет произведен поиск на отдельных дисках (в отдельных папках) или на всех внешних запоминающихся устройствах.




Пользователь может создать свои **Списки воспроизведения** - подборки любимых произведений или произведений, которые необходимо воспроизвести во время проведения определенного мероприятия, например рассказа об исполнителе с обзором его произведений из разных альбомов. Для создания списка воспроизведения нужно:

1. Выбрать в **Панели списка** кнопку **Список**.
2. Выбрать в списке команду **Редактировать с помощью редактора списков воспроизведения**.
3. Выбрать в открывшемся окне **Редактировать список воспроизведения** в списке **Критерии сортировки библиотеки** необходимую сортировку записей в библиотеке **Проигрывателя Windows Media**, например *Исполнитель и альбом*.
4. Открыть в левом окне нужный список произведений.
5. Переместить в поле **Список записи** необходимые мультимедийные объекты.
6. Выполнить команды 4-5 нужное количество раз.
7. Отсортировать список, используя кнопки .
8. Выбрать кнопку **ОК**.






Для сохранения созданного списка необходимо выбрать команду **Сохранить список как** из списка кнопки **Список**. Имя созданного списка воспроизведения появляется в папке **Мои списки воспроизведения Области навигации**. Пользователь может создать несколько списков воспроизведения.

Для воспроизведения аудио- или видеофайлов нужно:

1. Выбрать в **Области навигации** один из объектов:
  - в папке **Списки воспроизведения** один из списков воспроизведения;
  - в папке **Библиотека** одну из групп сортировки занесенных в библиотеку мультимедийных объектов.
2. Выбрать в **Области сведений** нужный альбом или файл.
3. Выбрать кнопку **Воспроизвести** .

При выборе видеофайла программа автоматически перейдет к отображению вкладки Проигрывается и в **Области просмотра** начнется воспроизведение видео. Для управления процессом воспроизведения звуковых или видеоданных используют кнопки управления (рис. 4.2, 10). Назначение этих кнопок описано в таблице 4.3.

**Таблица 4.3.** Кнопки управления воспроизведением в программе Проигрыватель Windows Media

Кнопка управления		Назначение
Изображение	Название	
	Остановить	Для остановки процесса воспроизведения
	Предыдущая дорожка	Для перехода к воспроизведению предыдущего произведения
	Следующая дорожка	Для перехода к воспроизведению следующего произведения
	Воспроизвести	Для начала или продолжения воспроизведения (во время воспроизведения изображение кнопки изменяется на изображение кнопки Пауза)
	Пауза	Для временной остановки процесса воспроизведения (во время остановки изображение кнопки изменяется на изображение кнопки Воспроизвести)

При необходимости пользователь может уменьшить окно проигрывателя и перейти к более компактному виду. Для этого следует выбрать кнопку Переключиться в режим обложки (рис. 4.2, 5). Вид окна Проигрывателя Windows Media в режиме обложки представлен на рисунке 4.3.

### Средства преобразования аудио- и видеофайлов из одного формата в другой




Рис. 4.3. Окно Проигрыватель Windows Media в режиме обложки

В связи с большим количеством разнообразных форматов аудио- и видеофайлов у пользователей часто возникает необходимость перекодировать эти файлы из одного формата в другой. Как вы знаете из курса информатики 9-го класса, процесс перекодирования файла одного формата в другой называется конвертацией файлов.

Конвертация может осуществляться в процессе открытия мультимедийного файла или при его сохранении на диске в другом формате. Одним из способов конвертации в программе Проигрыватель Windows Media является процесс копирования данных с оптических дисков форматов Audio CD, Video CD, DVD-Audio и DVD-Video и сохранение

их, например, на жестком магнитном диске в файлы формата WMA, либо WMV, либо другого по выбору пользователя. Для выполнения этой операции нужно:

1. Вставить аудио- или видеодиск в устройство для работы с оптическими дисками.
2. Выбрать кнопку Копировать с диска на Панели задач и открыть соответствующую вкладку.
3. Установить метки флажков возле выбранных для копирования произведений.
4. Выбрать кнопку Копировать музыку с компакт-диска .
5. Дождаться завершения процесса конвертации.

Ход процесса конвертации отображается в столбце Состояние процесса копирования компакт-диска. Пользователь может в любое время прекратить процесс конвертации, выбрав кнопку Остановить копирование.

Выбранные произведения копируются по умолчанию в папку Моя музыка, а их список отображается на вкладке Библиотека.

Обратный процесс конвертации медиафайлов для записи на диск и создание соответствующего формата аудио- или видеодиска в программе Проигрыватель Windows Media называется Запись. Создание подобных дисков необходимо, когда предусматривается прослушивание или просмотр медиаданных с использованием устройств, которые не воспроизводят данные, записанные в компьютерных форматах, например проигрыватели DVD-дисков устаревших моделей.

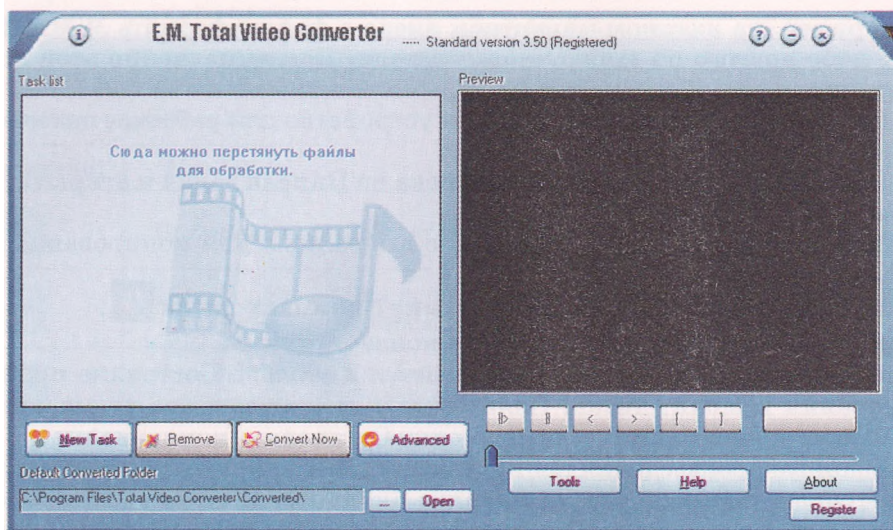
Для создания аудиодиска нужно:

1. Вставить чистый диск в устройство для работы с оптическими дисками.
2. Открыть вкладку Запись.
3. Установить метки флажков возле выбранных для копирования произведений.
4. Выбрать, если необходимо при использовании диска CD-RW, кнопку Очистка диска .
5. Выбрать, если необходимо изменить настройки записи, кнопку Обращение свойств и настроек .
6. Установить необходимые настройки скорости записи компакт-диска или уровня качества и др.
7. Выбрать в списке Элементы на устройстве команду Аудио компакт-диск.
8. Выбрать кнопку Начать запись .
9. Дождаться завершения сначала обработки (конвертации), а затем записи данных на диск.



Кроме медиаплееров, существует очень много специализированных программ для осуществления конвертации разнообразных мультимедийных файлов. Примером такой программы является программа Total Video Converter (англ. *Total Video Converter* - общий видеоконвертор). Загрузить ее можно с сайта <http://www.effectmatrix.com>.

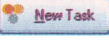

Невзирая на название, в котором речь идет о преобразовании видео, отмеченная программа осуществляет конвертацию как видеофайлов большинства форматов MP4, AVI, 3GP, MPG, MPEG, WMV, FLV, DIVX и др., так и звуковых файлов форматов CDA MP3, MP2, WAV, WMA и др.

Рис. 4.4. Окно программы **Total Video Converter**

Программа **Total Video Converter** предоставляет пользователю возможности:

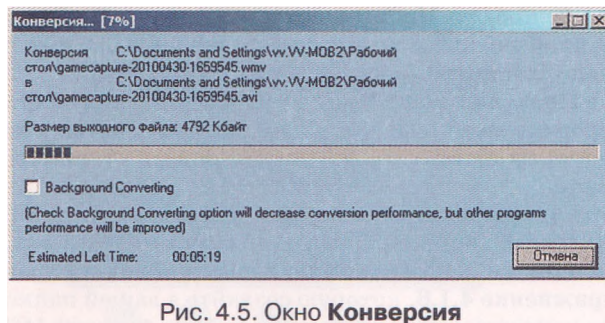
- конвертировать медиаданные из одного формата файла в другой с использованием разных типов кодеков;
- создавать потоковые презентации с использованием собственных шаблонов смены кадров или переходов между видеофрагментами;
- конвертировать несколько медиафайлов одновременно;
- объединять несколько видеороликов и аудиозаписей в одном файле;
- разделять видео- и звуковые дорожки и записывать их в разные файлы и др.

Окно программы **Total Video Converter** представлено на рисунке 4.4. Рассмотрим последовательность конвертации файлов на примере конвертации видеофайла формата MOV в файл формата AVI. Для этого нужно выполнить такие действия:

1. Выбрать кнопку **New Task**  (англ. *New Task* - новое задание (тема)).
2. Выбрать в списке команду **Import Media files** (англ. *Import Media files* - импорт медиафайлов).
3. Выбрать файл, который нужно конвертировать.
4. Выбрать в окне **Выбор формата** вкладку в соответствии с типом медиаданных (в нашем случае - вкладку **Видеофайлы**).  
Выбрать кнопку с названием формата, в который планируется конвертировать выбранный файл - AVI.
6. Выбрать в открывшемся списке название кодека, который будет использован во время конвертации, например WMV AVI.
7. Изменить при необходимости папку, в которую будет записан конвертированный файл, используя кнопку .
8. Установить при необходимости качество кодирования данных, используя список и гиперссылку **Settings** (англ. *Settings* - свойства).
9. Выбрать кнопку **Convert Now** (англ. *Convert Now* - конвертировать сейчас).
10. Дождаться завершения конвертации файла.

Процесс конвертации в зависимости от размеров файлов, алгоритмов сжатия и мощности компьютера, на котором происходит конвертация, может длиться достаточно долго. Для выполнения этой операции в фоновом режиме необходимо уста-



Рис. 4.5. Окно **Конверсия**

новить метку флажка **Background Converting** (англ. *Background Converting* - фоновое конвертирование) в окне **Конверсия** (рис. 4.5).

Аналогично можно конвертировать аудио- и видеофайлы разных форматов.

### Проверьте себя

- 1°. Какие данные называются мультимедийными? Приведите примеры.
- 2°. Назовите источники мультимедийных данных.
- 3°. Какие программы относят к программам обработки мультимедийных данных?
- 4°. Опишите классификацию программ обработки звуковых и видеоданных по назначению.
- 5°. Какие программы называют мультимедиаплеерами?
- 6°. Опишите возможности **Проигрывателя Windows Media**.
- 7°. Опишите режимы работы **Проигрывателя Windows Media**.
- 8°. Объясните, как записать **Аудио CD**, используя **Проигрыватель Windows Media**.
- 9°. Объясните, как сохранить записи с **Аудио CD** в звуковых файлах на жестком диске вашего компьютера, используя **Проигрыватель Windows Media**.
- 10°. Опишите, как создать **Список воспроизведения в Проигрывателе Windows Media**.
- 11°. Для чего используется компактный режим **Проигрывателя Windows Media**?
- 12°. Какие программы называют грабберами?
- 13°. Какие программы называют конверторами?
- 14°. Объясните, чем отличаются музыкальные и видеоредакторы от музыкальных видеостудий.
- 15°. Назовите известные вам форматы мультимедийных дисков и файлов.
- 16°. Объясните, чем отличается файл формата WAV от файла формата MP3.
- 17°. Что такое кодеки? Для чего они предназначены?
- 18°. Опишите последовательность конвертации файлов с использованием одной из программ конвертации, например **Total Video Converter**.

### Выполните задания

- 1°. Просмотрите список установленных программ на вашем компьютере. Установите, какие программы для обработки мультимедийных данных есть среди установленных. Определите, используя справку этих программ, какой их тип и для чего они предназначены.
- 2°. Проведите поиск мультимедийных файлов на вашем компьютере, используя стандартные средства операционной системы. Определите, файлы каких форматов были найдены.
- 3°. Запишите, используя **Проигрыватель Windows Media**, **Аудио CD**, включив в него звуковые файлы из папки, указанной учителем, например **Тема 4\Задания 4.1\Музыка**.

- 4. Запишите, используя Прои­грыватель Windows Media, в файлы формата WMA два произвольных произведения с Audio CD, который есть в вашей домашней коллекции.
- 5\*. Создайте в Прои­грывателе Windows Media список воспроизведения с именем, которое соответствует вашей фамилии, включив в него музыкальные произведения из папки, указанной учителем, например Тема 4\Задания 4.1\Музыка.
- 6\*. Проведите конвертацию двух первых файлов формата MP3 из папки, указанной учителем, например Тема 4\Задания 4.1\Музыка, в формат WMA, используя один из конверторов звуковых файлов. Сохраните эти файлы в папке Упражнение 4.1.6, которую создайте в вашей папке.
- 7\*. Проведите конвертацию двух первых файлов формата MOV из папки Тема 4\Задания 4.1\Видео в формат WMV, используя один из конверторов видеофайлов. Сохраните эти файлы в папке Упражнение 4.1.7, которую создайте в папке Мои документы.

## 4.2. Добавление звуковых и видеообъектов к слайдовым презентациям



1. Что такое слайдовая презентация? Какие объекты она может содержать?
2. Какие форматы звуковых и видеофайлов вы знаете?
3. Какие программы обработки звуковых и видеоданных вы знаете?
4. Какие программы называют грабберами?
5. Что такое анимация? Как добавить анимацию к объектам слайда?

Как вы уже знаете, слайдовые презентации, кроме текстовых и графических объектов, могут содержать звуковые и видеообъекты. Перед включением в презентацию в большинстве случаев эти объекты предварительно создаются и редактируются. Это связано с тем, что в программе PowerPoint 2007 в презентацию можно включить только файлы определенных типов: добавлять звуковые файлы форматов WMA, WAV MP3, MIDI, AU, AIFF и видеофайлы форматов WMV, AVI, MPEG, ASF. Поэтому перед включением в презентацию файлы других форматов следует конвертировать.

Еще одним способом подготовки звуковых и видеообъектов является захват и запись в файлы звука и видео с использованием специальных программ - грабберов.

### Запись (захват) аудио

Одной из самых простых в использовании программ для записи (захвата) звука является программа, входящая в стандартные программы операционной системы Windows, — программа Звукозапись. Для ее запуска следует выполнить *Пуск => Все программы => Стандартные => Развлечения => Звукозапись*.

Программа Звукозапись предназначена для записи, воспроизведения и редактирования звукозаписей. Она обеспечивает сохранение звукозаписи в файлах формата WAV без сжатия и соответственно без потери данных. Длительность одного сеанса записи не может превышать 60 с.

Для создания новой звукозаписи с использованием микрофона или другого подсоединенного к входу звуковой карты устройства ввода звука нужно:

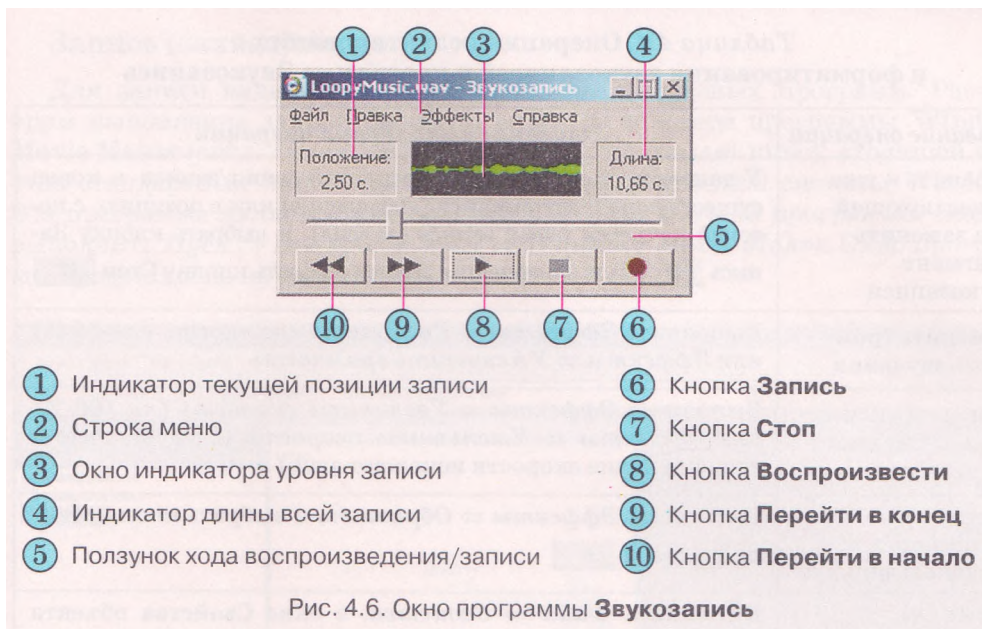






Рис. 4.6. Окно программы **Звукозапись**





- |  |                                  |
|--|----------------------------------|
| ① Индикатор текущей позиции записи     | ⑥ Кнопка <b>Запись</b>           |
| ② Строка меню                          | ⑦ Кнопка <b>Стоп</b>             |
| ③ Окно индикатора уровня записи        | ⑧ Кнопка <b>Воспроизвести</b>    |
| ④ Индикатор длины всей записи          | ⑨ Кнопка <b>Перейти в конец</b>  |
| ⑤ Ползунок хода воспроизведения/записи | ⑩ Кнопка <b>Перейти в начало</b> |

1. Выполнить *Файл* => *Создать* (рис. 4.6).
2. Выбрать кнопку **Запись** .
3. Проговорить в микрофон нужное сообщение (или воспроизвести на другом устройстве нужное звуковое сообщение).
4. Остановить процесс записи выбором кнопки **Стоп** .
5. Прослушать созданную звукозапись, выбрав кнопку **Воспроизвести** .
6. Сохранить созданную звукозапись в файле, выполнив *Файл* => *Сохранить* (*Сохранить как*).

Для создания записи длительностью более 60 секунд необходимо несколько раз повторить описанные в пунктах 2-4 действия.

Для прослушивания ранее созданной звукозаписи нужно:




1. Выполнить *Файл* => *Открыть*.
2. Выбрать файл, который нужно прослушать.
3. Выбрать кнопку **Воспроизвести** .

Воспроизведение звукозаписи можно остановить выбором кнопки **Стоп** , а затем продолжить воспроизведение с этой же позиции или с другой, выбрав ее ползунком хода воспроизведения/записи и выбрав кнопку **Воспроизвести** . Для быстрого перехода к началу звукозаписи или к ее концу следует выбрать кнопку **Перейти в начало**  или **Перейти в конец**  соответственно.

Программа **Звукозапись** предоставляет пользователю также возможности по редактированию и форматированию звукозаписей. Для выполнения операций редактирования или форматирования звукозаписи нужно открыть файл, который содержит нужную звукозапись, и выполнить действия, описанные в таблице 4.4.



**Таблица 4.4. Операции редактирования  
и форматирования звукозаписей в программе Звукозапись**

<i>Название операции</i>	<i>Описание выполнения операции</i>
Добавить к уже существующей или заменить фрагмент звукозаписи	Установить ползунок хода воспроизведения/записи в конец существующей звукозаписи (добавление) или в позицию, с которой начнется новая запись (замена), и выбрать кнопку <b>Запись</b>  . Для завершения записи выбрать кнопку <b>Стоп</b> 
Изменить громкость звучания	Выполнить <b>Эффекты</b> ⇒ <b>Увеличить громкость (на 25 %)</b> или <b>Эффекты</b> ⇒ <b>Уменьшить громкость</b>
Изменить скорость воспроизведения звукозаписи	Выполнить <b>Эффекты</b> ⇒ <b>Увеличить скорость (на 100 %)</b> или <b>Эффекты</b> ⇒ <b>Уменьшить скорость</b> (следует учесть, что изменение скорости искажает звук)
Воспроизвести звукозапись в обратном порядке	Выполнить <b>Эффекты</b> ⇒ <b>Обратить</b> и выбрать кнопку <b>Воспроизвести</b> 
Изменить качество звукозаписи	Выполнить <b>Файл</b> ⇒ <b>Свойства</b> , в окне <b>Свойства объекта «Звук»</b> выбрать кнопку <b>Преобразовать</b> и в следующем окне указать формат файла для преобразования и атрибуты
Изменить формат звукового файла	Выполнить <b>Файл</b> ⇒ <b>Сохранить как</b> , в окне <b>Сохранить как</b> выбрать кнопку <b>Изменить</b> и в следующем окне указать формат файла сохранения и атрибуты
Удалить часть звукозаписи	Переместить ползунок хода воспроизведения/записи в позицию, начиная с которой нужно удалить звукозапись, и выполнить <b>Правка</b> ⇒ <b>Удалить до текущей позиции (Удалить после текущей позиции)</b>
Добавить к существующей звукозаписи аудиоданные из другого файла	Переместить ползунок хода воспроизведения/записи в позицию, начиная с которой нужно добавить звукозапись из файла, выполнить <b>Правка</b> ⇒ <b>Вставить файл</b> и выбрать файл, звукозапись из которого нужно добавить. В ходе добавления старая звукозапись будет замещена звукозаписью из файла с текущей позиции
Смешать звукозаписи (микширование (англ. <i>mixing</i> – смешивать))	Переместить ползунок хода воспроизведения/записи в позицию, начиная с которой нужно смешать существующую звукозапись с новой, выполнить <b>Правка</b> ⇒ <b>Смешать с файлом</b> и выбрать файл, звукозапись из которого нужно добавить для смешивания. Во время смешивания одна запись налагается на другую



В операционной системе **Windows Vista** для записи и воспроизведения звука используется несколько иная программа **Звукозапись** (рис. 4.7). В ней отсутствуют средства для осуществления редактирования звукозаписей.

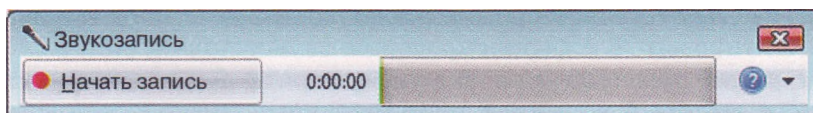


Рис. 4.7. Окно программы **Звукозапись**



## Запись (захват) видео

Для записи видео существует много разнообразных программ. Рассмотрим выполнение записи (захвата) видео на примере программы Windows Movie Maker (англ. *Movie Maker* - тот, который создает кино), входящей в состав стандартных прикладных программ операционной системы Windows. Эта программа является видеоредактором. Для запуска программы следует выполнить *Пуск => Все программы => Windows Movie Maker*. Окно программы представлено на рисунке 4.8.

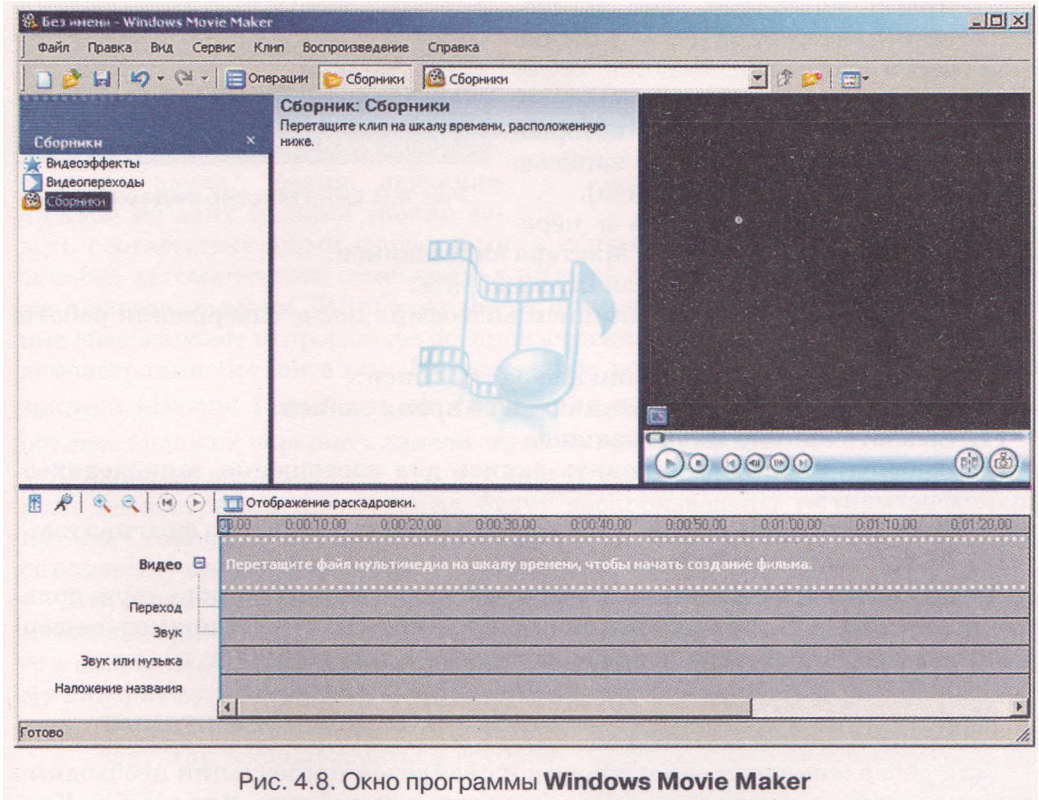


Рис. 4.8. Окно программы **Windows Movie Maker**

Во время записи видео может осуществляться запись и видео, и звука или только видео. Соответственно к компьютеру предварительно следует подсоединить устройства ввода видео- и звуковых данных или только видеоданных.

Для записи видео нужно выполнить такие действия:

1. Выполнить *Файл => Запись видео*.
2. Выбрать в поле *Доступные устройства* окна *Мастер видеозаписи* (рис. 4.9) устройство, из которого будет осуществлена запись видео.
3. Установить значения параметров устройств ввода звука: тип устройства для ввода и для обработки звука, уровень записи.
4. Выбрать кнопку *Далее* и перейти к следующему окну *Мастера видеозаписи*.
5. Ввести имя файла видеозаписи и путь к папке, в которую он будет записан.

6. Выбрать кнопку Далее и перейти к следующему окну Мастера видеозаписи.
7. Установить выбором соответствующего переключателя качество воспроизведения будущей видеозаписи.
8. Выбрать в списке при установленном значении переключателя Другие настройки одно из значений настроек видео (в нижней части окна указываются более детальные сведения о выбранных настройках и размере, который займет на диске файл с записью одной минуты видеозаписи).
9. Выбрать кнопку Далее и перейти к следующему окну Мастера видеозаписи.
10. Установлением меток флажков выбрать:
  - создавать ли из видеозаписи видеофайл после завершения работы программы мастера;
  - отключать ли динамики на время записи;
  - устанавливать ли ограничение на время записи.
11. Выбрать кнопку Начать запись.
12. Выбрать кнопку Остановить запись для завершения записи видеофрагмента.
13. Повторить шаги 10-11, если нужно записать несколько фрагментов.
14. Выбрать кнопку Готово.

Созданные видеофрагменты будут записаны в файлы в выбранную пользователем папку. В последующем их можно обрабатывать, например конвертировать в файлы другого формата, вставлять в документы и т. п.

## Добавление звуковых объектов к слайдовой презентации

Для добавления звукового объекта к слайдовой презентации необходимо выбрать слайд, в который он будет вставлен, и выполнить *Вставка => Клипы мультимедиа => Звук*. Откроется окно для вставки звука из файла. Если выбрать кнопку открытия списка команд кнопки Звук (рис. 4.10), то можно выбрать один из четырех способов вставки звукового объекта:

- Звук из файла — вставка звука из файла, хранящегося во внешней памяти или в сети;
- Звук из организатора клипов - вставка звука из файла, включенного в одну из коллекций: коллекции пользователя, коллекции Microsoft Office и Веб-коллекции;
- Запись с компакт-диска — вставка звука с музыкального компакт-диска, который находится в устройстве для работы с оптическими дисками;
- Записать звук — запись звукового комментария с микрофона и вставка его в слайд.

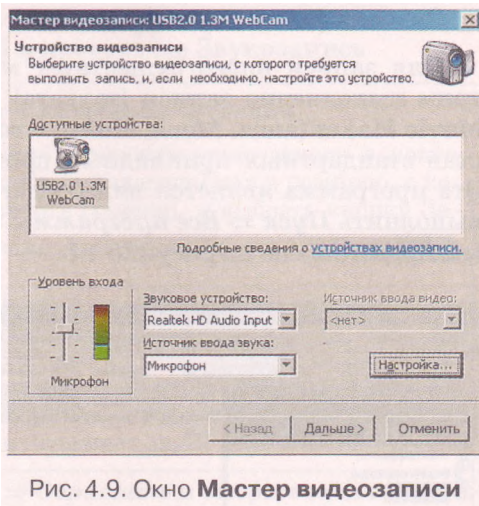


Рис. 4.9. Окно **Мастер видеозаписи**

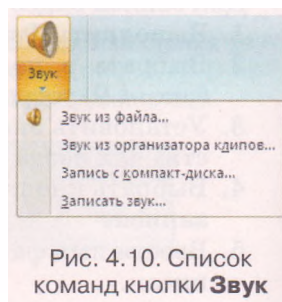








Рис. 4.10. Список команд кнопки **Звук**

Рассмотрим эти способы детальнее. Вставка звуковых объектов из файлов или из коллекции клипов происходит аналогично соответствующим операциям вставки рисунков.

Во время вставки звука с музыкального компакт-диска открывается окно установки параметров вставки (рис. 4.11). В этом окне пользователь устанавливает номер записи (звукового произведения) на аудиодиске, с которого следует начать воспроизведение звука, и номер записи, которая будет последней в воспроизведении звука. Время звучания каждой из этих записей можно задать соответствующими счетчиками, а общая длительность звучания всех записей автоматически отмечается в нижней части окна возле надписи *Время воспроизведения*. Установка метки флажка *непрерывное воспроизведение* обеспечивает непрерывное воспроизведение выбранных записей во время демонстрации презентации. Можно также установить громкость звучания записей (кнопка *Громкость звука* ) или скрыть пиктограмму звукового объекта (флажок *скрывать значок звука во время показа*).

Обращаем внимание на то, что при замене компакт-диска на другой в ходе демонстрации презентации будут воспроизведены те записи, номера которых совпадают с ранее установленными. Если диск отсутствует, то воспроизведения звука не будет.

Для вставки в слайд звукового комментария пользователя после выбора команды *Записанный звук* необходимо в окне *Звукозапись* (рис. 4.12) ввести в поле название звукового объекта и осуществить запись звука по такому алгоритму:

1. Подключить микрофон к компьютеру или воспользоваться встроенным. 
2. Выбрать кнопку *Запись* 
3. Проговорить в микрофон содержание комментария. 
4. Выбрать кнопку *Остановить* 
5. Прослушать записанный комментарий, выбрав кнопку *Воспроизвести* 
6. Если комментарий записан правильно, выбрать кнопку *ОК*, если нет — повторить команды 2-5 алгоритма.

На последних шагах операции вставки звуковых объектов на экран выводится окно установки значений параметров воспроизведения звука (рис. 4.13). Необходимо выбрать, как будет начинаться воспроизведение звука на слайде: сразу, как только начнется показ слайда (кнопка *Автоматически*), или после выбора пиктограммы звукового объекта на слайде во время

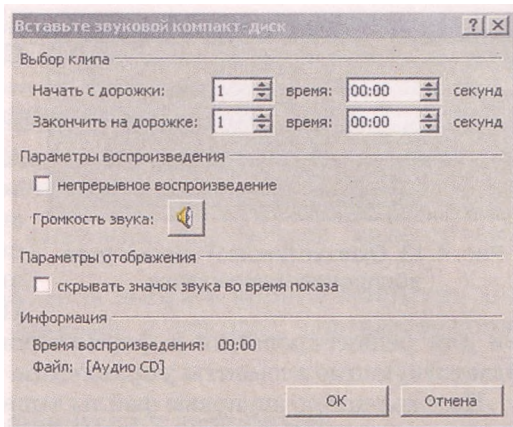


Рис. 4.11. Окно установки параметров вставки звука с компакт-диска

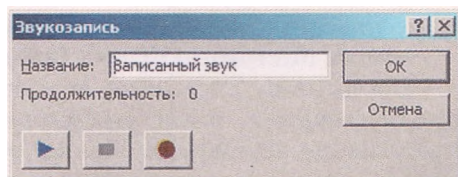


Рис. 4.12. Окно **Звукозапись**



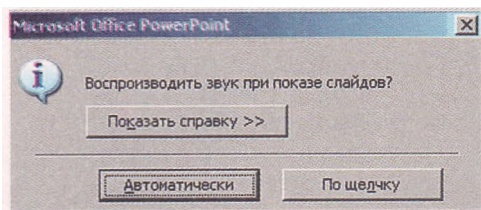



Рис. 4.13. Окно установки параметров воспроизведения звука

демонстрации презентации (кнопка По щелчку).

После вставки звукового объекта на слайде появляется его пиктограмма , если только не был избран режим ее скрытия. При выборе пиктограммы звукового объекта на Ленте появляется дополнительный раздел Работа со звуками и временная вкладка Параметры. После выбора этой вклад-

ки для редактирования и форматирования звука можно использовать соответствующие элементы управления.

По умолчанию звуковые файлы типа wav, размер которых более 100 Кбайт, автоматически связываются с файлом презентации, а не встраиваются в него. Для встроенных файлов типа wav можно указать максимальный размер до 50 000 Кбайт, но это приведет к увеличению общего размера файла презентации. Если же пользователь при вставке выберет файл большего размера, чем установлено этим параметром, то файл не будет включен в презентацию.

Воспроизведение звука происходит в режиме показа слайдов.

## Вставка и обработка видеообъектов

Для вставки видеообъекта необходимо выбрать слайд, в который он будет вставлен, и выполнить *Вставка => Клипы мультимедиа => Фильм*. Откроется окно для вставки фильма из файла. Если открыть список команд кнопки Фильм (рис. 4.14), то можно выбрать один из двух способов вставки видеообъекта:

- Фильм из файла - вставка видео из файла, что сохраняется во внешней памяти или в сети;
- Фильм из организатора клипов - вставка видео из файла, включенного в одну из коллекций: коллекции пользователя, коллекции Microsoft Office и Веб-коллекции.

Аналогично операции вставки звука во время вставки видео пользователю будет предложено установить, каким образом будет начинаться воспроизведение видео: автоматически или после выбора объекта на слайде в ходе его демонстрации.

После вставки видеообъекта на слайде появляется прямоугольная область, в которой будет воспроизводиться видеofilm. Размеры этой области после ее выделения можно изменить, используя соответствующие маркеры или установив точные размеры счетчиками Высота фигуры и Ширина фигуры группы Размер временной вкладки Формат дополнительного раздела Работа с рисунками. Можно также переместить область воспроизведения видео стандартной операцией перетягивания объекта.

После выделения видеообъекта на Ленте появляется дополнительный раздел Работа с фильмами и временная вкладка Параметры. Выбрав заглавие этой вкладки, пользователь получает доступ к элементам управления.

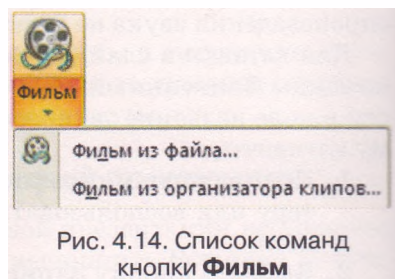


Рис. 4.14. Список команд кнопки **Фильм**



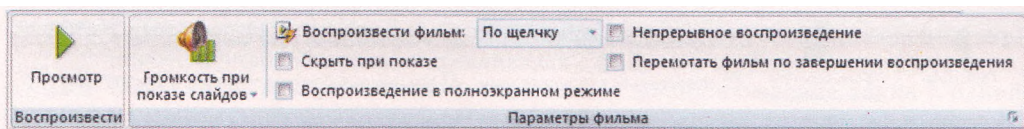


Рис. 4.15. Элементы управления временной вкладки **Параметры** раздела **Работа с фильмами**

используя которые, устанавливает значения параметров видеообъекта (рис. 4.15).

Эти же настройки можно изменить, если выбрать кнопку открытия диалогового окна группы **Параметры фильма** и открыть соответствующее окно.

### Добавление анимационных эффектов к звуковым и видеообъектам слайда

Для добавления анимационных эффектов к звуковым и видеообъектам используется та же последовательность действий, как и для текстовых или графических.

Только эффекты появления, выделения, выхода и путей перемещения применяются к значку объекта  (звуковой объект) или к окну, в котором воспроизводится видеообъект.

Список **Добавить эффект** для этих объектов содержит еще одну команду **Действия со звуком** или **Действия с фильмами** (рис. 4.16). В обоих случаях в открывшемся списке после выбора команды можно выбрать одну из команд: **Воспроизведение**, **Приостановка** или **Остановка**. Для установки значений параметров воспроизведения эффектов анимации следует выбрать кнопку открытия списка команд текущего эффекта и открыть окно, имеющее заглавие в зависимости от эффекта, параметры которого настраиваются. Для видеообъекта, например, это могут быть такие окна: **Фильм: воспроизведение**, **Фильм: пауза** или **Фильм: остановка**.

На вкладке **Эффект** окна **Фильм: воспроизведение** (рис. 4.17) пользователь может установить значения таких параметров:

- **начало воспроизведения:** *С начала*, *С последней позиции* или *По времени*;
- **завершение воспроизведения:** *по нажатию левой кнопки мыши (По щелчку)*, *После текущего* или *слайда с заданным номером*;
- **действия после анимации** - аналогично действиям для текстовых и графических объектов.

На вкладке **Параметры фильма** окна **Фильм: воспроизведение** пользователь может установить значения таких параметров:

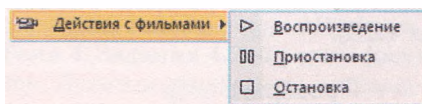


Рис. 4.16. Список команд **Действия с фильмами**

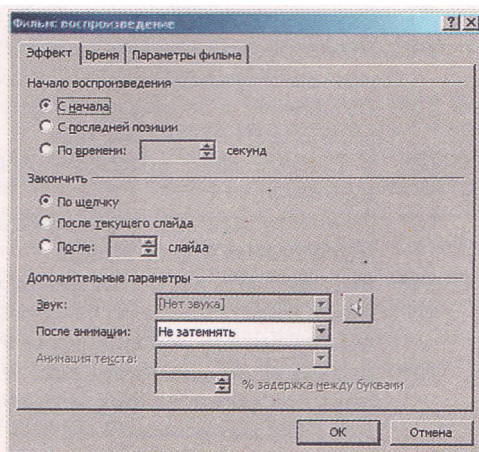


Рис. 4.17. Вкладка **Эффект** окна **Фильм: воспроизведение**

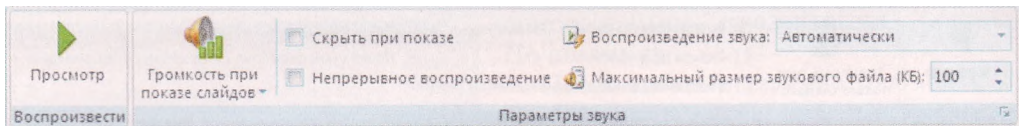


Рис. 4.18. Группы элементов управления **Параметры звука** и **Воспроизвести** вкладки **Параметры** дополнительного раздела **Работа со звуками**

- громкость звука видеофильма;
- скрытие при отсутствии воспроизведения;
- отображение видео на весь экран.

Настройка эффектов звуковых объектов аналогична настройке эффектов видео, за исключением отображения на весь экран.

Настроить параметры воспроизведения видео- (звуковых) объектов можно также с использованием элементов управления групп **Параметры фильма** (**Параметры звука**) и **Воспроизвести** вкладки **Параметры** дополнительного раздела **Работа с фильмами** (**Работа со звуками**) (рис. 4.15 и 4.18).



## Проверьте себя

- 1°. Звуковые файлы каких форматов можно добавлять к презентации в PowerPoint 2007?
- 2°. Видеофайлы каких форматов можно добавлять к презентации в PowerPoint 2007?
- 3°. Для чего предназначена программа Звукозапись? Как ее запустить?
- 4°. Объясните операции, которые можно выполнить с использованием программы Звукозапись.
- 5°. Опишите интерфейс программы Звукозапись.
- 6°. Укажите последовательность действий при создании звукозаписи в программе Звукозапись.
- 7°. Укажите последовательность действий при прослушивании звукозаписи в программе Звукозапись.
- 8°. В каком формате файлов сохраняет звукозаписи программа Звукозапись?
- 9°. Опишите, как подготовить звукозапись длительностью 3 мин в программе Звукозапись.
- 10°. Опишите, какие возможности по редактированию и форматированию звукозаписи предоставляет пользователю программа Звукозапись.
- 11\*. Объясните, как смешать одну звукозапись с другой. Для каких случаев используют эту операцию?
- 12°. К какому виду программ относится программа Windows Movie Maker? Как ее запустить?
- 13°. Опишите выполнение операции записи видео с использованием программы Windows Movie Maker.
- 14°. Как добавить звуковой или видеообъект к слайду презентации?
- 15°. Объясните особенности вставки звука с компакт-диска. Какие виды оптических дисков используются в этом случае?
- 16\*. Сравните операции записи аудио с использованием программы Звукозапись и вставки в слайд звукового комментария пользователя после выбора команды Записанный звук. В каких случаях лучше использовать каждую из этих операций?
- 17°. Опишите, какие параметры воспроизведения видеообъекта во время демонстрации презентации можно установить и как это сделать.

18. Опишите отличия вставки и установки параметров воспроизведения звукового и видеообъектов, вставляемых с файлов в слайдовую презентацию.
19. Объясните, как установить непрерывное воспроизведение звука с одной из дорожек компакт-диска на протяжении демонстрации всей презентации.



### Выполните задания

- 1°. Запишите в собственном исполнении декламацию стихотворения И. Франко «Сикстинська Мадонна» (Тема 4\Задания 4.2\Франко.<i>i</i>osx) с использованием программы Звукозапись. Сохраните документ в собственной папке в файле с именем упражнение 4.2.1.wav.
- 2°. Откройте в программе Звукозапись указанный учителем файл, например Тема 4\Задания 4.2\образец 4.2.2.wma. Допишите к существующей звукозаписи декламацию стихотворения И. Франко «Котляревський» (Тема 4\Задания 4.2\Франко.<i>c</i>osx) в собственном исполнении. Сохраните звукозапись в собственной папке в файле с именем упражнение 4.2.2.wav.
3. Запишите в собственном исполнении декламацию стихотворения И. Франко «Сонети - се раби. У форми пута...» (Тема 4\Задания 4.2\Франко.<i>c</i>osx) с использованием программы Звукозапись. Допишите еще одну декламацию - стихотворения И. Франко «Колись в сонетах Данте і Петрарка...» (Тема 4\Задания 4.2\Франко.<i>c</i>osx). Сохраните документ в папке Моя музыка в файле с именем упражнение 4.2.3.wav.
4. Откройте в программе Звукозапись указанный учителем файл, например Тема 4\Задания 4.2\образец4.2.4^та. Установите ползунок хода воспроизведения/записи в место, где заканчивается первый куплет стихотворения И. Франко «Сикстинська Мадонна». Удалите часть звукозаписи с этого места до конца звукозаписи. Допишите в эту звукозапись первый куплет стихотворения И. Франко «Котляревський» (Тема 4\Задания 4.2\Франко.<i>doc</i>x) в собственном исполнении. Сохраните документ в собственной папке в файле с именем упражнение 4.2.4.wav.
- 5\*. Используя справочную систему программы Звукозапись, определите, что такое уровень записи. Как его изменить?
6. Подключите к компьютеру веб-камеру и микрофон. Выполните с использованием программы Windows Movie Maker видеозапись декламации произвольного фрагмента произведения М. Франко «Сучасна приказка» (Тема 4\Задания 4.2\Франко^osx). Сохраните видеозапись в формате WMV в собственной папке в файле с именем упражнение 4.2.6.wmv.
7. Подготовьте вместе с одноклассниками групповую декламацию одного из произведений, которые вы изучаете в курсе зарубежной литературы. Выполните с использованием программы Windows Movie Maker видеозапись подготовленной декламации. Сохраните видеозапись в формате WMV в папке Мои видеозаписи в файле с именем упражнение 4.2.7.wmv.
- 8°. Откройте указанный учителем файл (например, Тема 4\Задания 4.2\образец 4.2.8.pptx). Добавьте к первому слайду звуковой объект из файла, например Тема 4\Задания 4.2\звук 4.2.8.mid. Установите непрерывное воспроизведение этого объекта во время всей демонстрации презентации. Сохраните презентацию в собственной папке в файле с именем упражнение 4.2.8. pptx.
9. Откройте указанный учителем файл, например Тема 4\Задания 4.2\образец 4.2.9.pptx. Добавьте:
  - а) к слайду с рассказом о белке вместо ее фотографии видеозапись из файла, например Тема 4\Задания 4.2\белка.<i>au</i>i. Установите скрытие видеозаписи в начале воспроизведения и автоматическое воспроизведение

после появления заглавия. Сохраните презентацию в собственной папке в файле с именем упражнение 4.2.9. pptx;



- б) к слайду с рассказом о еже вместо его фотографии видеозапись из файла, например Тема 4\Задания 4.2\еж.ауі. Установите воспроизведение этого объекта через 1 с после появления текста и показ на весь экран. Сохраните презентацию в папке Мои документы в файле с именем упражнение 4.2.9\_1.pptx.

- 10\*. Откройте указанный учителем файл, например Тема 4\Задания 4.2\образец 4.2.10.pptx. Просмотрите слайды презентации, посвященной устройствам компьютера. Подготовьте краткие сообщения о свойствах современных устройств, которые бы сопровождали демонстрацию презентации. Запишите речевое сопровождение для слайдов с изображениями:

- а) **монитора** и **планшетного сканера**. Установите воспроизведение звукового сопровождения сразу после перехода к слайдам, на которых размещены объекты. Время показа слайдов не сохранять. Сохраните презентацию в собственной папке в файле с именем упражнение 4.2.10 pptx;



- б) **процессора**. Установите воспроизведение звукового сопровождения после нажатия левой кнопки мыши. Время показа слайда сохранить. Сохраните презентацию в папке Мои документы в файле с именем упражнение 4.2.10\_1.pptx.

## 4.3. Создание и редактирование видеofilьмов (поточковых презентаций)



1. Что, по вашему мнению, отличает кинофильм от видеofilьма? Что нужно иметь для просмотра видеofilьма?
2. Что такое сценарий? Где он используется?
3. Для чего можно использовать программу Windows Movie Maker? Как ее запустить?
4. Как выполнить запись (захват) звука, видео? Какие программы осуществляют подобные операции? Как называется этот тип программ?

### Общие требования к созданию видеofilьма

Термин видеofilьм происходит от термина кинофильм (англ. *cine-film* - кинопленка, кинофильм), который в широком понимании означает производство киноискусства. Видеofilьм отличается от кинофильма способом фиксации данных: кинофильм - на фотографических пленках с использованием кинокамер, а видеofilьм - на видеокассетах и видеодисках с использованием видеокамер. Для небольших по размеру кино- и видеofilьмов употребляют термин клип (англ. *clip* - сжимать, обрезать), или видео-клип. В последнее время видеоклипы широко используются в музыкальной индустрии. Их называют музыкальными видеоклипами.

По своей структуре и технологии создания потоковые презентации фактически являются видеofilьмами (видеоклипами), предназначенными для представления чего-либо нового: идей, проектов, продукции, товара и т. п. Для создания потоковых презентаций используют видеоредакторы.

На первом этапе, перед началом создания видеofilьма, необходимо разработать его сценарий. Сценарий — это детально разработанный план проведения какого-либо мероприятия, осуществления каких-либо дей-



ствий. Сценарий даже небольшого видеоклипа должен иметь внутреннюю структуру (композицию). Сценарий должен быть детальным, с посекундным планированием всех действий.

На втором этапе происходит съемка всех эпизодов в соответствии со сценарием, готовятся фотографии, рисунки, подбирается музыка, записывается звуковое сопровождение. Часто видеофильм создается на основе ранее отснятого видеоматериала. Например, фильм об определенном периоде истории страны, фильм-эссе о жизни выдающихся людей, даже фильм о вашем классе, по-видимому, будет создаваться на основе уже существующих документальных кадров, фотографий, документов и т. п.

На третьем этапе осуществляется монтаж — процесс обработки и структурирования отдельных объектов (кадров) для получения целостного произведения - видеофильма. В процессе монтажа отснятый материал редактируется, уточняется сценарий, накладывается звук и титры - текстовые комментарии к кадрам и т. п.

## Разработка сценария видеофильма

Рассмотрим процесс создания сценария видеофильма о птицах, которые живут у рек. Его можно использовать в ходе изучения темы Виды животных из курса «Я и Украина» 3-го класса. Проанализируем имеющийся видео- и фотоматериал (табл. 4.5).

Таблица 4.5. Файлы для использования в видеофильме

Имя файла	Папка размещения	Содержание	Длительность, с
лысуха.avi	Тема 4\Видео	Видео	42
анст.avi	Тема 4\Видео	Видео	31
лысуха01-jpg	Тема 4\Фото	Фотографии	-
лысуха02.jpg	Тема 4\Фото	Фотографии	-
анст01.jpg	Тема 4\Фото	Фотографии	-
анст02.jpg	Тема 4\Фото	Фотографии	-
анст.wav	Тема 4\Звук	Звук	4
лысуха.wav	Тема 4\Звук	Звук	6,75

В соответствии с программой начальной школы и требованиями ко времени демонстраций на уроке для учеников 3-го класса составим сценарий нашего видеофильма. Сценарий состоит из сценарных кадров — описания смысловых фрагментов видеофильма и тех событий, которые в них происходят.

### 1. Кадр 1. Титульная страница:

- заголовок - *Птицы Украины*, подзаголовок - *Для учащихся 3-х классов;*

- цвет фона - *зеленый*, цвет букв - *белый*;
  - длительность показа - *4 с*;
  - звук - *отсутствует*;
  - видеоэффект - *зернистость*;
  - эффект перехода на следующий кадр - *глаз*.
2. Кадр 2. Фотография аиста:
- файл - *аист02.jpg*;
  - титры - *Аист*; подзаголовок (*аист, бусел, черногуз*);
  - длительность показа - *4 с*;
  - звук — файл *аист.wav*; длительность звучания — *4 с*;
  - эффект перехода на следующий кадр - *вихрь*.
3. Кадр 3. Фотография аиста:
- файл -*аист01.jpg*;
  - длительность показа - *4 с*;
  - звук - *отсутствует*;
  - эффект перехода на следующий кадр - *бабочка, вертикальная*.
4. Кадр 4. Видео Аиста:
- файл - *аист.avi*;
  - длительность показа - *26,14 с* (общая длительность - *31 с*, необходимо удалить в начале *4,86 с*; оригинальный звук видеозаписи нужно удалить);
  - звук - *дикторский текст*; длительность - *22,14 с*;
  - эффект перехода на следующий кадр - *вставить, слева снизу*;
  - дикторский текст: *Вы увидели птицу и услышали ее голос, точнее стук клюва. Это аист, или черногуз, бусел. Имеет белое перо, а на крыльях черное. У аиста длинные ноги и длинный клюв красного или желто-красного цвета. В Украине аист считается вестником весны.*
5. Кадр 5. Фотография лыски:
- файл - *лысуха02.jpg*;
  - титры - *Евразийская, или обычная, лыска (лысуха)*;
  - длительность показа - *4 с*;
  - звук — файл *лысуха.wav*; начало звучания через *2 с* после начала кадра, длительность звучания - *6,75 с* (звук продолжается в следующем кадре);
  - эффект перехода на следующий кадр - *вставить, слева сверху*.
6. Кадр 6. Фотография лыски:
- файл - *лысуха01.jpg*;
  - длительность показа — *4 с*;
  - звук - файл *лысуха.wav*; началось звучание на предыдущем кадре, завершается вместе с завершением показа фотографии;
  - эффект перехода на следующий кадр - *диафрагма*.
7. Кадр 7. Видео лыски:
- файл - *лысуха.avi*;
  - длительность показа - *30 с* (общая длительность - *42 с*, необходимо удалить в конце *12 с*; оригинальный звук видеозаписи сделать фоном для дикторского текста);

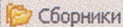
- эффект перехода на следующий кадр - диагональ, ниша;
- звук - дикторский текст; длительность - 30 с;
- дикторский текст: *Евразийская, или обычная, лыска, которую еще называют лысуха - небольшая водоплавающая птица, которая гнездится у рек и озер на всей территории Украины. Ее легко узнать по белому клюву и белому пятну на лбу. Значительную часть времени лыска проводит на воде, за что получила еще одно народное название: болотная или водяная курица.*



#### 8. Кадр 8. Заключительные титры:

- титры - *Житомирский лицей № 1, видеостудия «Будущее»:*  
О Оператор - Михайленко Степан;  
О Автор текста - Коваленко Оксана;  
О Звук - Иваненко Николая;  
О Ученики 11 А класса для учеников 3-х классов;
- цвет фона - желто-зеленый, цвет букв - белый;
- длительность показа — 10 с.

### Создание видеофильма

Перейдем к созданию видеофильма в программе Windows Movie Maker. Для этого рассмотрим подробнее интерфейс этой программы (рис. 4.19).

В левой части окна после выбора кнопки Сборники  отображается одноименная панель (рис. 4.19, 3) с тремя стандартными папками Видеоэффекты, Видеопереходы и Сборники. При выборе папки Видеоэффекты в Области отображения содержимого папок (рис. 4.19, 4) отобразится список видеоэффектов (отображение в оттенках серого, отображение в виде акварельного рисунка, отображение сверху вниз и т. п.), которые можно применить к отдельным кадрам или к фильму в целом. Папка Видеопереходы содержит список эффектов перехода от одного кадра к другому. Папку Сборники пользователь заполняет сам разными объектами - графическими, звуковыми и видеофайлами, которые потом сможет включить в свои фильмы.

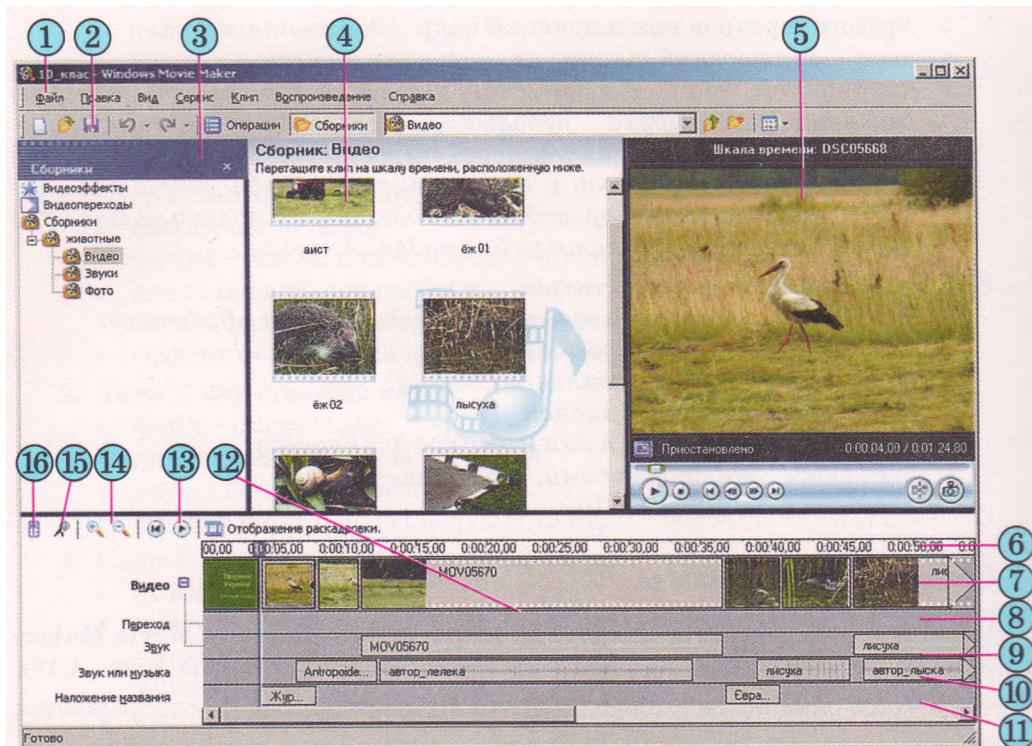
В правой части окна размещено окно проигрывателя (рис. 4.19, 5) для предварительного просмотра результатов монтажа. Нижнюю часть окна занимает Область монтажа (рис. 4.19, 12). Она отображается в одном из двух режимов: в режиме раскадровки или в режиме отображения шкалы времени. Для переключения режимов Области монтажа используются кнопки Отображение шкалы времени  Отображение шкалы времени и Отображение раскадровки  Отображение раскадровки.

Область монтажа в режиме отображения шкалы времени содержит: звуковую дорожку (рис. 4.19, 9), видеодорожку (рис. 4.19, 7), звуковую или музыкальную дорожку (рис. 4.19, 10), дорожку титров (рис. 4.19, 11).

В верхней части области размещена шкала времени (рис. 4.19, 6).

В режиме раскадровки в Области монтажа отображаются кадры - те видео- или графические фрагменты, из которых состоит видеофильм.

Режим отображения шкалы времени используется в основном для вставки мультимедийных объектов и определения длительности их показа или




- |   |  |
|---|--|
| ① Строка меню                           | ⑨ Звуковая дорожка   |
| ② Панель инструментов                   | ⑩ Звуковая или музыкальная дорожка                               |
| ③ Панель <b>Сборники</b>                | ⑪ Дорожка титров   |
| ④ Область отображения содержимого папок | ⑫ <b>Область монтажа</b>   |
| ⑤ Окно проигрывателя                    | ⑬ Кнопки управления предварительным просмотром                   |
| ⑥ Шкала времени                         | ⑭ Кнопки управления масштабом шкалы времени                      |
| ⑦ Видеодорожка                          | ⑮ Кнопка включения режима записи комментария                     |
| ⑧ Дорожка переходов                     | ⑯ Кнопка включения режима регулирования (установки уровня) звука |

Рис. 4.19. Окно программы **Windows Movie Maker**


звучания, а режим раскадровки - для применения к кадрам видеоэффектов и эффектов перехода между кадрами.

На первом этапе видеомонтажа необходимо создать новый проект. Для этого следует выполнить *Файл => Создать проект* или выбрать кнопку

Создать, проект  на панели инструментов. Проект содержит ссылку на мультимедийные объекты, которые входят в состав фильма, значения их свойств в конкретном фильме и другие данные, которые нужны для работы



над монтажом фильма. Для сохранения проектов программа Windows Movie Maker использует специальный формат файлов MSWMM (англ. *Microsoft Windows Movie Maker*). Пользователю нужно периодически сохранять файл проекта для того, чтобы в случае сбоев в работе компьютера результаты монтажа не были утеряны.



Файлы объектов, которые будут входить в фильм, следует поместить в один из сборников. Для упорядочения объектов желательно предварительно создать в сборнике папки, например, как представлено на рисунке 4.19: папка животные, в которой созданы папки Видео, Звуки, Фото. Для создания папки в выбранной необходимо воспользоваться кнопкой Создать папку сборников  или соответствующей командой контекстного меню папки Сборники.

На *втором этапе* следует вставить мультимедийные объекты из сборника в область монтажа перетягиванием. Это можно выполнить так:

1. Установить режим раскадровки выбором соответствующей кнопки.
2. Перетянуть по очереди в соответствии с разработанным сценарием из папок Видео и Фото видео- и графические объекты в большие прямоугольники (кадры) Области монтажа (рис. 4.20).
3. Перейти в режим отображения шкалы времени.
4. Перетянуть по очереди в соответствии с разработанным сценарием звуковые объекты из папки Звуки в Область монтажа в те точки на шкале времени, с которых начнется воспроизведение звука.
5. Отредактировать длительность показа вставленных объектов.



Рис. 4.20. Область монтажа в режиме раскадровки

Если вставлять видеообъект в режиме отображения шкалы времени, то во время перетягивания возле указателя появляется метка , а на видеодорожке и на звуковой дорожке - метка , которая указывает на место вставки объекта (рис. 4.21, 1-2). Возле указателя также появляется прямоугольник с сообщениями о моменте времени в общей длительности фильма, с которого начнется демонстрация объекта (например, 0 ч 0 мин 10,80 с), об имени объекта, возле которого вставляется выбранный объект (например, Аист), и о длительности показа объекта (например, 0 ч 0 мин 25,39 с).

При вставке графических объектов программой устанавливается стандартная длительность показа - 4 с. Для изменения этой длительности необходимо перетянуть правую границу объекта на видеодорожке (рис. 4.22). Можно как увеличить, так и уменьшить длительность показа графического объекта.

Для видеообъектов длительность их показа будет равняться установленной длительности в соответствующем видеофайле. Изменение длительности



- ① Метка места вставки объекта на звуковой дорожке
- ② Метка места вставки объекта на видеодорожке
- ③ Метка места начала текущего объекта на шкале времени
- ④ Момент времени в общей длительности фильма, с которого начнется демонстрация объекта
- ⑤ Имя объекта, рядом с которым вставляется объект
- ⑥ Длительность демонстрации вставляемого объекта

Рис. 4.21. Сообщения и метки при вставке объекта в **Область монтажа**

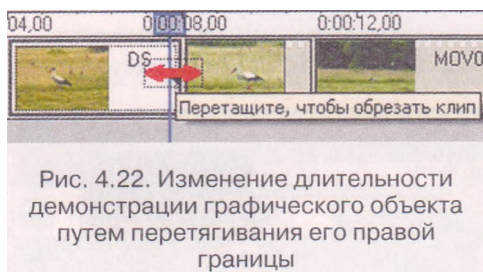



Рис. 4.22. Изменение длительности демонстрации графического объекта путем перетягивания его правой границы

показа видеообъекта осуществляется аналогично изменению длительности показа графического объекта, только можно перетягивать как левую, так и правую границу видеообъекта на видеодорожке. При этом фрагмент видеообъекта не будет воспроизводиться от его начала или конца до момента времени, определенного положением границы.

Для более точного редактирования объекта в Области монтажа можно увеличить/уменьшить масштаб отображения объектов. Для этого используют кнопку Увеличить/уменьшить шкалу времени .

Во время перетягивания границ в окне проигрывателя отображается текущий видеокادر, что дает возможность сориентироваться, к какому месту следует перетянуть границу. Это процесс обратимый и можно в дальнейшем восстановить длительность показа видеообъекта, перетянув границу в обратном направлении. Вместе с тем увеличить длительность показа больше, чем это предусмотрено свойствами файла, невозможно.

Звуковые объекты вставляются и редактируются аналогично видеообъектам, только перетягиваются они на звуковую или музыкальную дорожку.

В ходе вставки звуковых и видеообъектов и при их редактировании (изменении длительности, изменении момента времени начала воспроизведения) происходит *синхронизация* звукового и видеоряда.

Результаты монтажа следует периодически просматривать в окне проигрывателя. На следующем *третьем этапе монтажа видеофильма* вставляются титры или текстовое сопровождение. Пример начальных титров представлен на рисунке 4.23.

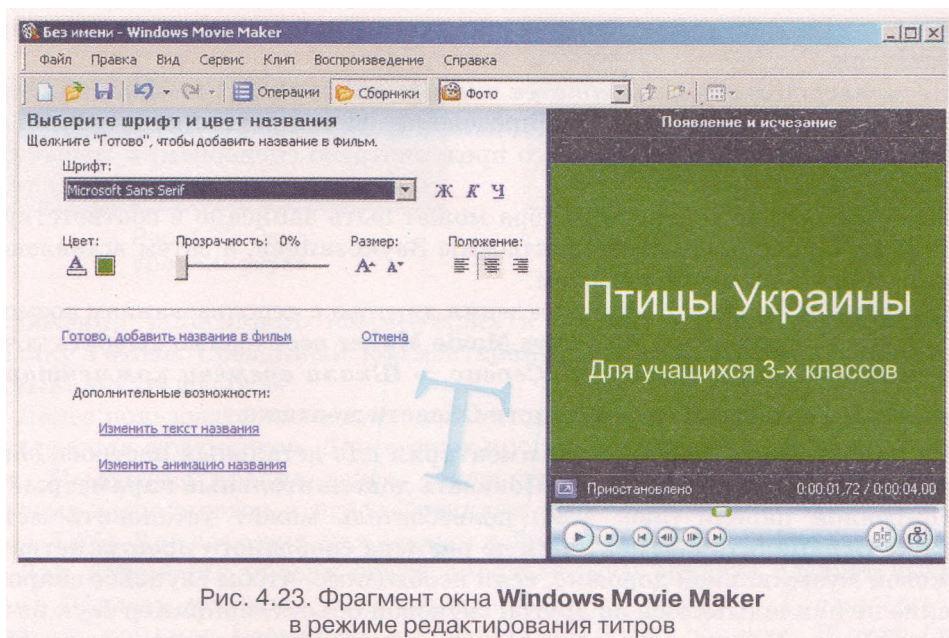


Рис. 4.23. Фрагмент окна **Windows Movie Maker** в режиме редактирования титров

Для добавления текстовых надписей к видеofilmу нужно выполнить такую последовательность действий:

1. Выполнить *Сервис => Названия и титры*.
2. Выбрать на панели *Где добавить название?* одну из гиперссылок, которая соответствует месту вставки надписи. Различают начальные и заключительные титры, а также титры, которые размещаются на графических или видеообъектах или перед ними.
3. Ввести текст надписей в соответствующие поля на панели *Введите текст названия*.
4. Изменить при необходимости для надписей, которые размещаются не на графических или видеообъектах, цвет фона, цвет букв, шрифт и начертание, а также эффект анимации (гиперссылка *Изменить шрифт и цвет текста* и *Изменить анимацию текста*).
5. Просмотреть, как будет смотреться надпись в видеofilmе, для чего выбрать кнопку *Воспроизвести* в окне проигрывателя.
6. Выбрать гиперссылку *Готово*.

Созданные надписи будут вставлены либо на видеодорожку, либо на дорожку титров. На видео дорожку будут вставлены все надписи, за исключением тех, которые вставляются на определенном кадре (графическом или видео).

Для вставки титров и надписей внутри видеofilmа и заключительных титров следует повторить приведенный алгоритм нужное количество раз.

Длительность демонстрации вставленных титров по умолчанию составляет 3,5 с. Изменить ее можно таким же образом, как и длительность демонстрации графических объектов. Для титров, которые размещаются внутри фильма, можно перетягивать как левую, так и правую границу.


В Области монтажа для большинства объектов можно изменять последовательность их просмотра путем перетягивания условных обозначений вставленных объектов (прямоугольников) в пределах соответствующей дорожки.



Результаты монтажа следует периодически просматривать в окне проигрывателя.

На четвертом этапе монтажа видеofilьма, когда размещение и длительность показа всех объектов определены, фильм дополняется звуковым сопровождением диктора (если это предусмотрено сценарием) и музыкальным сопровождением.

Звуковое сопровождение диктора может быть записано в соответствующий файл с использованием программы Звукозапись, а затем вставлено в фильм описанным выше способом.

Для записи звукового сопровождения диктора с использованием возможностей самой программы Windows Movie Maker необходимо выбрать точку на шкале времени и выполнить *Сервис => Шкала времени комментария* или выбрать кнопку  в левой части Области монтажа.

На панели Шкала времени комментария для детальных настроек записи следует выбрать гиперссылку Показать дополнительные параметры. На расширенной панели (рис. 4.24) пользователь может установить метку флажка Ограничить комментарий до размера свободного пространства на звуковой музыкальной дорожке, если необходимо, чтобы звуковое сопровождение не накладывалось на другой звуковой объект, например звук из видеофрагмента. При этом программа автоматически прекратит запись, если звуковые фрагменты начнут накладываться.

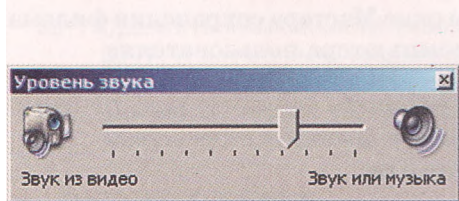
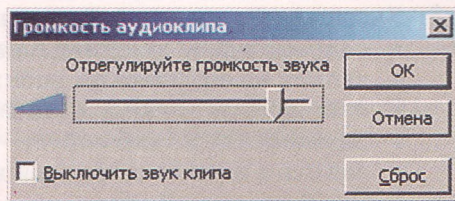
Установление метки флажка Выключить динамики дает возможность избежать возможного искажения звука от наложения звуковых фрагментов. Следует также опробовать уровень сигнала от микрофона и отрегулировать его, используя ползунок Уровень входа.

Для начала записи комментария необходимо выбрать кнопку Запустить, а для приостановки или завершения записи - кнопку Остановить. Для



Рис. 4.24. Панель Шкала времени комментария с дополнительными параметрами настройки



Рис. 4.25. Окно **Уровень звука**Рис. 4.26. Окно **Громкость аудиоклипа**

включения записанного комментария к фильму следует выбрать гиперссылку Готово. Созданный комментарий будет вставлен на звуковую или музыкальную дорожку.



Далее пользователю следует отредактировать и выровнять уровни звука для разных фрагментов. Для этого в программе есть несколько инструментов. Изменение уровня звука одновременно всех звуковых фрагментов фильма выполняется с использованием диалогового окна **Уровень звука** (рис. 4.25). Для его открытия следует выполнить *Сервис* => *Уровень звука*.

Для изменения уровня звука определенного объекта следует выбрать его в *Области монтажа* и выполнить *Клип* => *Аудио* => *Громкость* или выбрать команду **Громкость** в контекстном меню объекта. В окне **Громкость аудиоклипа** (рис. 4.26) установить ползунком нужный уровень звука объекта.

Для полного выключения звука фрагмента (например, звука фрагмента видео) необходимо выбрать фрагмент в *Области монтажа* и выполнить *Клип* => *Аудио* => *Выключить* или выбрать команду **Выключить** в контекстном меню объекта.

После редактирования и синхронизации звукового и видеоряда следует посмотреть результаты монтажа в окне проигрывателя.

*Пятый этап монтажа видеофильма* заключается в настройке видеоэффектов и эффектов перехода между кадрами фильма. Для этого лучше перейти к режиму раскадровки в *Области монтажа*.

Для вставки видеоэффекта достаточно на панели *Сборники* выбрать папку **Видеоэффекты** и перетянуть нужный эффект на кнопку  в нижнем левом углу нужного кадра. К каждому кадру можно установить несколько видеоэффектов. Для предварительного просмотра результата применения эффекта следует воспользоваться проигрывателем. Для удаления эффекта следует выбрать кнопку  на нужном кадре и нажать клавишу **Delete**.

Для вставки эффекта перехода между кадрами необходимо на панели *Сборники* выбрать папку **Видеопереходы** и перетянуть нужный эффект на прямоугольную область между нужными кадрами. Результат применения эффекта следует посмотреть с использованием проигрывателя. Для удаления эффекта перехода следует выбрать прямоугольную область между нужными кадрами и нажать клавишу **Delete**.

После применения всех эффектов следует проверить результаты монтажа с использованием проигрывателя программы.

Последний этап работы над видеофильмом - сохранение видеофильма в файл. Для этого нужно:

1. Выполнить *Файл* => *Сохранить файл фильма*.

2. Указать действие над файлом в первом окне Мастера сохранения фильма:
  - сохранение во внешней памяти компьютера пользователя;
  - запись на оптический диск;
  - отправка по электронной почте;
  - размещение на веб-сайте;
  - запись в память цифровой видеокамеры.
3. Выбрать кнопку Далее.
4. Если выбрано сохранение во внешней памяти компьютера, то следует в следующем окне указать имя файла и папку, в которую он будет сохранен, и выбрать кнопку Далее.
5. Установить значения параметров качества видеofilма и выбрать кнопку Далее.
6. Дождаться завершения кодирования фильма с выбранными параметрами качества и сохранения в файл.


Файл будет сохранен в формате WMV и в дальнейшем его можно использовать как обычный видеофайл: копировать, воспроизводить с использованием видеопроигрывателей, конвертировать, вставлять в слайдовые презентации и т. п.

### Проверьте себя

- 1°. Что такое видеofilm? Чем он отличается от кинофильма?
- 2°. Объясните, для чего разрабатываются сценарии.
- 3°. Опишите этапы создания видеofilmа. Какие операции выполняются на каждом из них?
- 4°. Объясните, какие действия следует выполнить в ходе подготовки сценария видеоклипа о вашей семье.
- 5°. Что такое сценарный кадр? Какие данные указываются в сценарном кадре?
- 6°. Что такое монтаж видеofilmа?
- 7°. Опишите, какие подготовительные действия следует выполнить в программе Windows Movie Maker перед началом монтажа видеofilmа.
- 8°. Для чего в программе Windows Movie Maker используются папки Сборники, Видеопереходы, Видеоэффекты?
- 9°. Какие режимы отображения Области монтажа есть в программе Windows Movie Maker? Как их переключать?
- 10\*. Объясните назначение дорожек **Области монтажа** в режиме отображения шкалы времени.
- 11°. Что такое файл проекта в программе Windows Movie Maker? Какие данные он содержит?
- 12°. Опишите последовательность действий для включения видео- и графических объектов в видеofilm.
- 13°. Опишите последовательность действий для включения звуковых объектов в видеofilm.
- 14°. Как выключить воспроизведение звукового фрагмента?
- 15°. Объясните способы изменения длительности воспроизведения графических, видео-, звуковых и текстовых объектов.
- 16°. Опишите последовательность действий при вставке текстовых объектов в **Область монтажа**. Какие есть варианты их размещения относительно других объектов видеofilmа?
- 17°. Как изменить формат текстовых объектов в программе Windows Movie Maker?
- 18°. Опишите порядок добавления дикторского звукового сопровождения к фильму.
- 19°. Опишите последовательность сохранения файла видеofilmа в программе Windows Movie Maker.

- 20\*. Сравните возможности слайдовых презентаций, созданных в PowerPoint 2007, и потоковых, созданных в Windows Movie Maker, на примере работы с титульными страницами.
- 21\*. Можно ли создать в PowerPoint 2007 презентацию, по своим свойствам похожую на видеофильм (потоковую презентацию)? Ответ обоснуйте.

### Выполните задания

- 1°. Создайте видеофильм «Птицы Украины» по сценарию, который описан в этом пункте. Материалы для включения в видеофильм размещены в папках Видео, Звук, Фото папки Тема 4\Задания 4.3. Сохраните видеофильм в собственной папке в файле с именем упражнение 4.3.1.wmv.
2. Составьте сценарий и создайте по нему в Windows Movie Maker потоковую презентацию о Лесе Украинке к курсу украинской литературы 10-го класса. В презентации используйте фрагмент из видеофильма «Великі українці. Леся Українка», фотографии Леси Украинки и памятников ей из папки Тема 4\Задания 4.3\Украинка. Подготовьте и включите в видеофильм титульный слайд и титры сопровождения. Самостоятельно подберите музыку, которая должна звучать на протяжении всего видеофильма. Сохраните видеофильм в собственной папке в файле с именем упражнение 4.3.2.wmv.
-  3. Подготовьте в Windows Movie Maker потоковую презентацию о руководителях украинского государства начала XX века, используя материал учебника истории Украины 10-го класса и материалы, размещенные в папках Тема 4\Задания 4.3\Скоропадский и Тема 4\Задания 4.3\Петлюра. Дикторский текст и музыкальное сопровождение подберите самостоятельно. Сохраните видеофильм в собственной папке в файле с именем упражнение 4.3.3.wmv.

### Практическая работа № 15. Создание видеоклипа (потоковой презентации)

**Внимание!** Во время работы с компьютером соблюдайте правила безопасности и санитарно-гигиенические нормы.

1. Создайте, используя программу Windows Movie Maker, видеоклип (потоковую презентацию) по вопросу охраны окружающей среды.
2. Разработайте сценарий презентации, предусмотрев:
  - Кадр 1. Титульная страница:
    - О заголовок - *Сохраним природу родного края*, подзаголовок - *Учебная программа*;
    - О цвет фона - *желтый*, цвет букв — *темно-зеленый*;
    - О анимация - *видео*, в тексте;
    - О длительность показа - *4,40 с*;
    - О звук - *отсутствует*;
    - О эффект перехода на следующий кадр - *круги*.
  - Кадр 2. Фотография мусора:
    - О файл - *Мусор01.jpg*;
    - О титры на слайде - *Что нужно сделать, чтобы такого не было?*
    - О начало показа титров - *через 2 с после начала показа фотографии*;
    - О длительность показа - *3 с*;
    - О анимация — *Класс!*;
    - О звук - файл *FINCL 02.mid*; длительность звучания - *5 с*;
    - О эффект перехода на следующий кадр - *вихрь*.
  - Отобразите в следующих кадрах основные проблемы загрязнения окружающей среды и пути борьбы за ее чистоту. Общая длительность видеоклипа - от 3 до 5 мин.

3. Смонтируйте видеоклип, включив в него нужные видео-, звуковые, графические и текстовые объекты из папки Тема 4\Задания 4.3\Экология или из других источников.
4. При необходимости отредактируйте объекты и синхронизируйте видео- и аудиоряды.
5. Сохраните презентацию в своей папке в файле с именем Практическая работа 15.wmv.

#### 4.4. Основы создания компьютерных публикаций. Программа Microsoft Office Publisher 2007



1. Что вы знаете о полиграфии? Какие виды печатных изданий выпускают полиграфические предприятия?
2. Когда и где появились первые печатные издания?
3. Как классифицируют программы обработки текста? Что такое издательские системы?
4. Что такое шаблон? Как используются шаблоны в Word 2007 и PowerPoint 2007?

#### Публикации. Компьютерные публикации

Публикацией (лат. *publico* - объявлять публично) называют процесс издания определенного произведения. Изданное (обнародованное) произведение также называют публикацией.

В зависимости от способа предания огласке произведения различают *печатные публикации* и *электронные*. К печатным публикациям относят книги, брошюры, журналы, газеты, бюллетени, буклеты, открытки и т. п. К электронным публикациям относят документы, которые распространяются с использованием электронных носителей данных (компакт-диски, дискеты, флэш-накопители и т. п.) или компьютерных сетей.

Длительное время печатные публикации изготовлялись исключительно в типографии. И этим занималась целая отрасль производства - полиграфия (греч. *πολύ* - много и *γραφειν* - писать). А подготовкой материалов к печати занималось издательство. В многотиражном производстве так происходит и сегодня: издательство готовит макеты книг, журналов, газет и т. п., а предприятия полиграфии — печатают и при необходимости сшивают их.

Вместе с широким распространением персональных компьютеров и высококачественных лазерных и струйных принтеров, копировальных аппаратов приобрели распространения так называемые *настольные издательские системы*. Эти системы имеют аппаратную и программную составляющие. Аппаратная составляющая включает персональный компьютер, устройства печати (принтер, копировальный аппарат, плоттер и т. п.) и устройства ввода данных (сканеры, графические планшеты и т. п.). А к программной составляющей относятся специальные программы подготовки макета публикации. Все это может разместиться на столе пользователя, и поэтому эти системы получили название *desktop publishing* (англ. *desktop publishing* — настольные публикации), или настольные издательские системы.

Вы уже знаете с курса информатики 9-го класса, что системы обработки текста, предназначенные для создания макетов полиграфических изданий и подготовки их к печати, также называются *издательскими системами*. К таким программам относят QuarkXPress, Corel Ventura, Adobe PageMaker, Adobe InDesign и т. п. Это профессиональные программы, которые ис-



пользуют, как правило, для подготовки к печати больших по объему публикаций массовым (от сотен до сотен тысяч) тиражом.

Как издательские системы для малых предприятий и организаций, учебных заведений, а также для создания публикаций дома используют текстовые процессоры и специализированные программы настольных издательских систем с упрощенным интерфейсом и набором функций - Microsoft Publisher, Page Plus, Avery DesignPro Paraben's Label Builder, Publish-It! и т. п.

В отличие от текстовых процессоров, издательские системы ориентированы не на ввод и редактирование текста и других объектов, а на верстку.



**Верстка** — это процесс компоновки текстовых и графических объектов для создания страниц издания в соответствии с принципами дизайна и техническими требованиями.

По назначению различают *книжно-журнальную верстку*, *газетную верстку* и *акцидентную* (лат. *accidentia* - случай, случайность) *верстку*. Акцидентная верстка применяется для верстки отдельных, как правило, небольших по количеству страниц и по тиражу объектов: объявлений, бланков, афиш, буклетов, проспектов, визиток и т. п.

Публикации, которые осуществлены с использованием компьютерных технологий, называют *компьютерными публикациями*.

## Основные возможности настольных издательских систем

Настольные издательские системы обеспечивают:

- автоматизацию процессов размещения блоков текста и графических изображений в соответствии с назначением публикации (*макетирование*);
- импортирование фрагментов текста и графических изображений, созданных в разных прикладных программах;
- обработку фрагментов текста, используя стили и расширенные возможности по установке расстояния между символами и между строками;
- автоматизацию создания содержания и ссылок;
- использование расширенного набора шрифтов в соответствии с требованиями полиграфии;
- воспроизведение публикации на экране монитора с учетом особенностей устройства, на котором будет осуществлена печать публикации;
- сохранение файлов публикации в форматах, которые используются в полиграфии;
- возможность печати публикаций каждым основным цветом отдельно (цветоделение)

и др.

Набор предоставляемых программами операций может меняться в зависимости от назначения программы, от полиграфической базы, на которую программа ориентирована.

## Общая характеристика издательской системы Microsoft Office Publisher 2007

Настольная издательская система Microsoft Office Publisher 2007 (далее Publisher 2007), как уже отмечалось, предназначена для создания компьютерных публикаций малыми тиражами и используется в небольших пред-

приятных и организациях, а также для обеспечения личных потребностей пользователя.

Программа Publisher 2007 имеет упрощенную по сравнению с профессиональными издательскими системами функциональность. Она обеспечивает создание и акцидентную верстку публикаций на основе разнообразных шаблонов содержания и может работать в двух основных режимах - создание и редактирование печатной и электронной публикации.

Стандартным форматом файлов для сохранения публикаций, подготовленных с использованием Publisher 2007, является формат PUB. Следует учитывать, что сохранение публикаций в файлах предыдущих версий программы Publisher (1998 и 2000 года) может привести к потере ряда особенностей оформления публикации.

Основным объектом в Publisher 2007 является публикация как совокупность страниц. На страницах печатных публикаций могут размещаться текстовые и графические объекты, а на страницах электронных публикаций еще и звуковые, видеообъекты и гиперссылки. Свойства объектов публикаций практически не отличаются от свойств аналогичных объектов, рассмотренных в Word 2007 и PowerPoint 2007. Над каждым из объектов Publisher 2007 можно выполнять определенный набор операций по редактированию и форматированию.

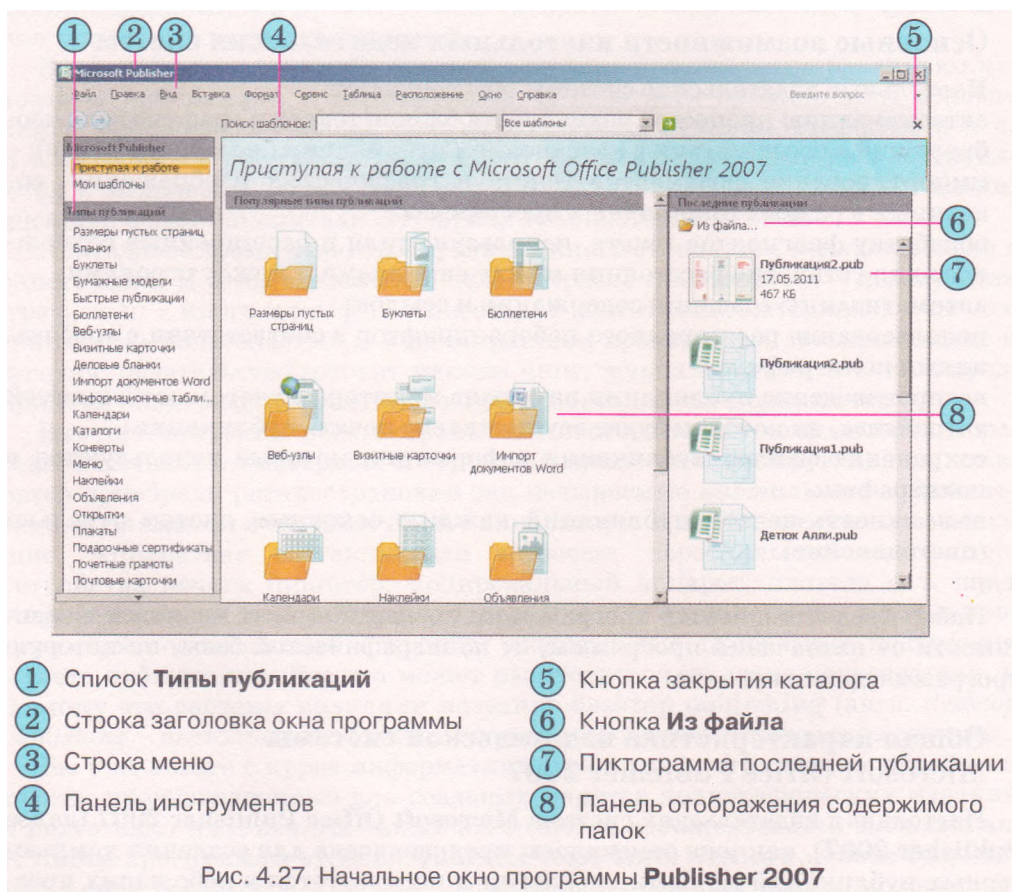



Рис. 4.27. Начальное окно программы **Publisher 2007**

## Запуск Publisher 2007. Окно программы

Программу Publisher 2007 можно запустить на выполнение несколькими способами. Приведем самые распространенные из них:

выполнить Пуск => Все программы => Microsoft Office => Microsoft Office Publisher 2007;

- использовать ярлык программы на Рабочем столе или Панели быстрого запуска;
- дважды щелкнуть на значке файла публикации Publisher, например таким — .

Если выбран один из первых двух вариантов запуска программы, то на экране появится начальное окно программы, разделенное на три части (рис. 4.27)


В левой части окна размещен список типов публикаций (рис. 4.27, 1), которые можно создать в Publisher 2007. При выборе в этом списке одного из объектов, например Буклеты, в центральной части отображаются эскизы шаблонов публикации этого типа. При этом в правой части вместо панели Последние публикации отобразится панель с элементами управления, используя которые устанавливаются значения свойств будущей публикации.

Поиск нужных шаблонов публикаций по их названиям можно осуществить, используя поле Поиск шаблонов и список, в котором можно выбрать место поиска: на этом компьютере, на узле Microsoft Office Online или в обоих этих местах (Все шаблоны).

### Создание новой публикации

Создание новой публикации можно начать с начального окна программы, которое описано выше.

После выбора типа публикации и одного из шаблонов на панели настройки значений свойств (рис. 4.28) можно изменить цветовую схему, схему шрифтов (или оставить эти свойства со значениями из шаблона по умолчанию), указать данные об организации или пользователе, которые автоматически будут вставляться во все шаблоны публикаций, и выбрать форму для осуществления регистрации, проведения опроса и т. п. После выбора кнопки Создать, размещенной под панелью установки значений свойств выбранного шаблона, программа перейдет в режим редактирования и форматирования публикации.

Можно пропустить эти действия, для чего в начальном окне программы следует выбрать кнопку Закрывать каталог  (рис. 4.27, 5). Начальное окно будет закрыто, и откроется основное окно программы (рис. 4.29), которое имеет несколько другой вид, чем окна рассмотренных программ пакета Microsoft Office 2007. При этом в рабочей области будет размещен пустой лист.

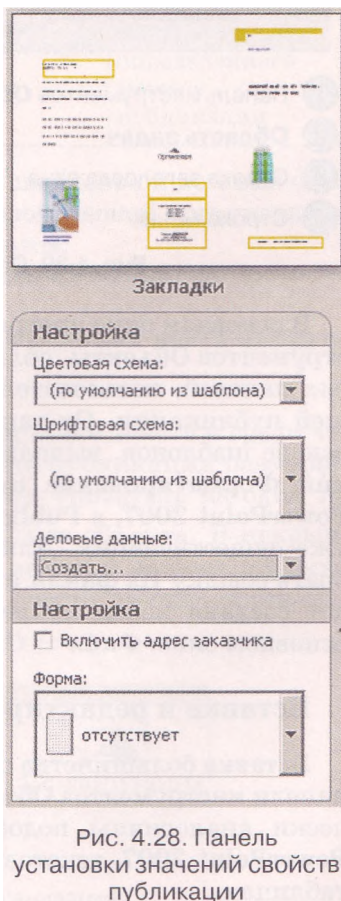
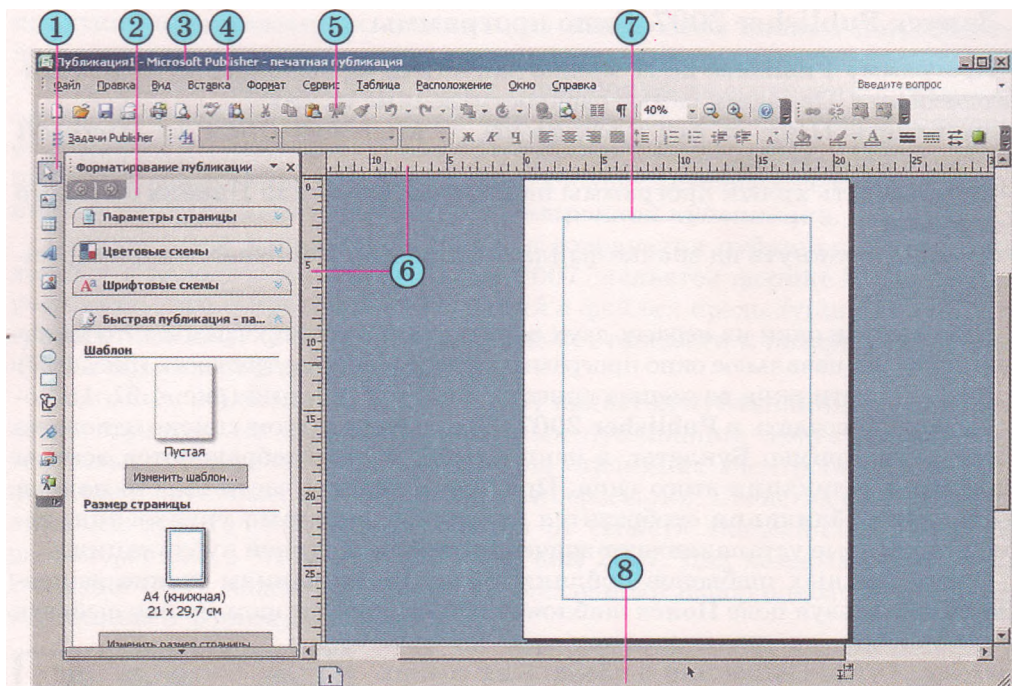


Рис. 4.28. Панель установки значений свойств публикации





- |                                      |                       |
|--------------------------------------|-----------------------|
| ① Панель инструментов <b>Объекты</b> | ⑤ Панели инструментов |
| ② <b>Область задач</b>               | ⑥ Линейки             |
| ③ Строка заголовка окна              | ⑦ Рабочая область     |
| ④ Строка меню                        | ⑧ Строка состояния    |

Рис. 4.29. Основное окно программы **Publisher 2007**

В основном окне программы, используя элементы управления панели инструментов **Объекты**, области заданий **Форматирование публикации** и других панелей, пользователь может самостоятельно разработать макет будущей публикации. Он также может вернуться к созданию публикации на основе шаблонов, выполнив *Файл => Создать* или выбрав в области заданий **Форматирование публикации** кнопку *Изменить шаблон*. Как и в **PowerPoint 2007**, в **Publisher 2007** можно создавать публикации на основе уже существующих. Для этого в начальном окне программы следует выбрать ссылку *Из файла* и указать файл, на основе содержания которого будет создана новая публикация. Такое же действие можно выполнить в основном окне: *Файл => Открыть*.

## Вставка и редактирование объектов

Вставка большинства объектов осуществляется с использованием кнопок панели инструментов **Объекты** (табл. 4.6). Операции над объектами практически аналогичны подобным операциям в **PowerPoint 2007**. Как и в **PowerPoint 2007**, текстовые объекты размещаются в надписях, фигурах и таблицах.




Таблица 4.6. Кнопки панели инструментов Объекты и их назначение

Кнопка	Название	Назначение	Кнопка	Название	Назначение
	Выбор объектов	Для выбора объектов публикации		Надпись	Для вставки текстового объекта (надписи)
	Вставить таблицу	Для вставки таблицы		Вставить объект WordArt	Для вставки объекта WordArt
	Рисунок	Для вставки графического объекта		Линия	Для вставки линии
	Стрелка	Для вставки стрелки		Овал	Для вставки овала
	Объект библиотеки макетов	Для открытия окна библиотеки макетов публикаций		Авто-фигуры	Для вставки одной из фигур библиотеки автофигур
	Элемент из библиотеки содержимого	Для открытия области заданий Библиотека содержимого		Закладка	Для вставки закладки для быстрого поиска определенного места в публикации
	Прямоугольник	Для вставки прямоугольника			

Рассмотрим последовательность использования шаблонов и вставки объектов на примере создания листа молодежной экологической организации. Для начала работы следует выполнить такой алгоритм:

1. Запустить на выполнение программу Publisher 2007.
2. Выбрать в списке типов шаблонов Бланки.
3. Выбрать нужный шаблон (например, шаблон Цветная лента из группы Современные макеты).
4. Выбрать кнопку Создать.

Будет создан макет публикации с местами для вставки текста и графических объектов. В верхней правой части этого макета публикации размещены несколько объектов (рис. 4.30). Следует навести указатель на один из объектов и выбрать кнопку , появляющуюся возле объекта. В открывшемся списке следует выбрать команду Изменить деловые данные и в окне Редактирования нового набора деловых данных ввести данные организации, например такие:


Адрес основного места работы .	Телефон: (555)555-55-55	 Организация
Адрес, строка 2 .	Факс: (555)555-55-55	
Адрес, строка 3 .	Эл. почта:	
Адрес, строка 4	proverka@example.com	

Рис. 4.30. Объекты шаблона бланка организации

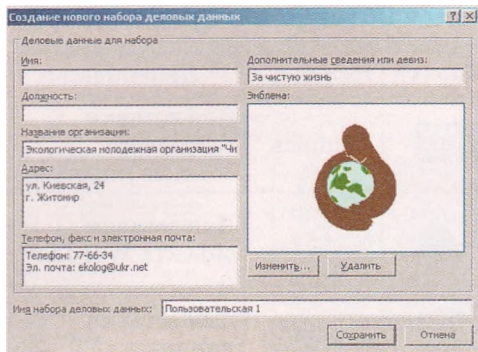


Рис. 4.31. Окно Редактирования нового набора деловых данных

После завершения редактирования данных окно будет выглядеть так, как это представлено на рисунке 4.31. Далее следует выбрать кнопку Сохранить и в следующем окне выбрать кнопку Обновить публикацию.

Служебные сведения сохраняются и в дальнейшем будут автоматически вставляться в шаблоны других публикаций.

Для ввода текста с данными об адресате нужно в правой верхней части публикации, под эмблемой, вставить надпись и ввести текст, например такого содержания: *Директору малого предприятия «Автохим»*. На основном поле листа следует также вставить надпись и ввести текст, например такого содержания:

*Уважаемый господин директор!*

*В течение прошлого месяца сотрудники малого предприятия «Автохим» постоянно сваливали мусор в зеленой зоне непосредственно у забора предприятия. Это ухудшает экологию нашего микрорайона и является недопустимым. Требуем немедленно принять меры и убрать мусорную свалку.*

*З надеждой на понимание и сотрудничество в деле сохранения природы родного края.*

*Фотографию мусорной свалки прилагаем.*

*Председатель организации*

*Петренко Николай*




Рис. 4.32. Макет письма

Под текстом письма следует вставить фотографию свалки мусора, например из файла Тема 4\Задания 4.4\Мусор.jpg. Для этого нужно на панели инструментов Объекты выбрать кнопку Рисунок . В открывшемся списке выбрать команду Рисунок из файла и выделить прямоугольную область на поле публикации - место размещения рисунка. В окне Вставка ри-



сунка, которое откроется автоматически, указать нужный файл и выбрать кнопку **Вставить**.

Под рисунком следует сделать надпись, например такую: *Мусорная свалка возле малого предприятия «Автохим»*, и сгруппировать ее с рисунком. Для группировки рисунка и надписи следует их выделить и выбрать кнопку **Сгруппировать объекты** , которая появляется возле выделенных объектов.

Общий вид подготовленной публикации (макет) представлен на рисунке 4.32.

## Сохранение и печать публикации

Для сохранения публикации в файле следует выполнить *Файл => Сохранить* и в окне **Сохранить** как указать папку для сохранения, имя файла, его формат и выбрать кнопку **Сохранить**.

Настройка параметров и сам процесс печати практически не отличаются от аналогичной операции в Word 2007. Для установки значений свойств печати можно предварительно открыть окно **Настройка печати** (*Файл => Настройка печати*). Подобные настройки можно установить и в окне **Печать** (*Файл => Печать*) (рис. 4.33).

Отличия в настройках окон **Печать** в программах Word 2007 и Publisher 2007 заключаются в наличии поля предварительного просмотра и измененных элементах управления для установки печати разного количества страниц на листе.

Учитывая, что программа Publisher 2007 имеет средства связывания объектов, которые размещены в отдельных файлах, при перемещении публикации на другие компьютеры эти объекты могут не воспроизводиться

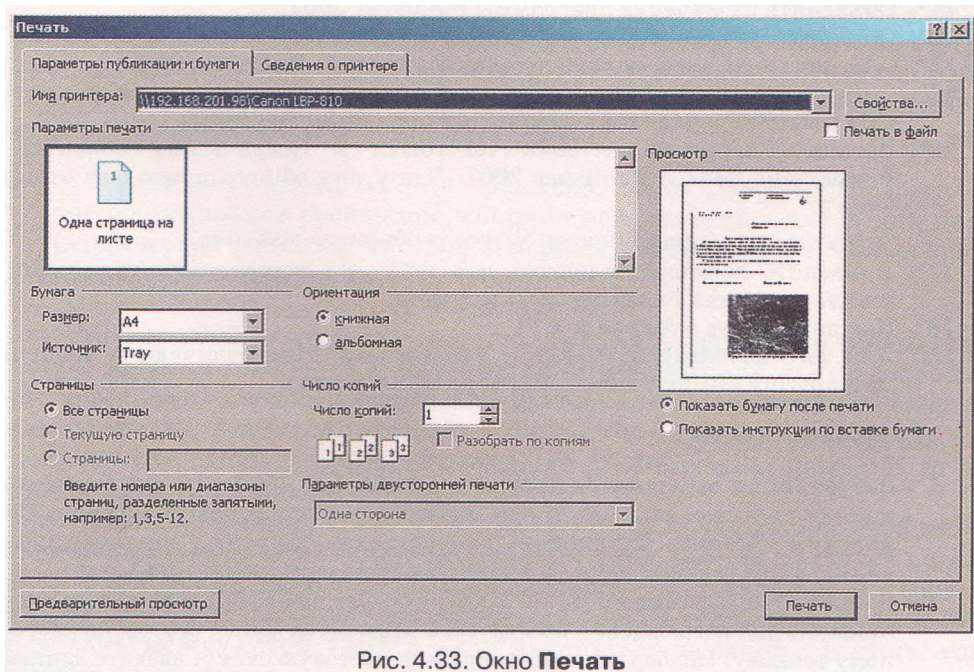


Рис. 4.33. Окно **Печать**

ся. Для корректного перемещения публикации на другие компьютеры с целью последующего просмотра или печати используется средство упаковки публикации. Для его использования следует выполнить *Файл => Упаковать => Для другого компьютера/Для передачи в печать*. В обоих случаях откроется окно Мастер упаковки. Последовательно указывая значения свойств упаковки (место сохранения, объекты, которые следует включить и связать) на каждом шагу работы мастера упаковки, пользователь создает архивный файл формата ZIP. Отличие в упаковке для другого компьютера и для печати заключается в том, что во время упаковки для другого компьютера не предлагается включать в архивный файл шрифты и файлы связанных объектов.

### Проверьте себя

- 1°. Что такое публикация?
- 2°. Что такое печатная публикация? Приведите примеры печатных публикаций.
- 3°. Что такое электронная публикация? Приведите примеры электронных публикаций.
4. Какая роль в создании публикаций отводится предприятиям полиграфии; издательствам?
- 5°. Что такое настольная издательская система?
6. Что входит в состав настольной издательской системы? Для чего предназначена каждая составляющая?
- 7°. Что такое верстка?
8. Какие существуют виды верстки по назначению? Где их используют?
- 9°. Что такое компьютерная публикация? Приведите примеры компьютерных публикаций.
10. Опишите возможности настольных издательских систем.
11. Объясните назначение программы Publisher 2007.
12. Опишите интерфейс начального окна программы Publisher 2007.
13. Опишите интерфейс основного окна программы Publisher 2007.
- 14°. Опишите последовательность создания новой публикации с использованием элементов управления начального окна Publisher 2007.
15. Сравните операции вставки текстовых и графических объектов в PowerPoint 2007 и Publisher 2007. Что у них общего и чем они отличаются?
16. Для чего используют кнопку ® возле объектов шаблона?
- 17\*. Приведите последовательность действий для сохранения публикации в стандартном для Publisher 2007 формате.
- 18°. Как напечатать публикацию?
- 19°. Для чего используют упаковку публикаций? Чем отличаются упаковки **Для другого компьютера** и **Для передачи в печать**?

### Выполните задания

- 1°. Запустите на выполнение программу **Publisher 2007**. Используя начальное окно программы, создайте новую публикацию на основе шаблона **Объявления — Маркер**. Установите цветовую схему **Сирень**. Сохраните публикацию в собственной папке в файле с именем **упражнение 4.4.1.pub**.
2. Создайте в программе **Publisher 2007** на основе шаблона **Грамота** новую публикацию для награждения победителей первенства вашей школы по настольному теннису. Подберите самостоятельно цветовую схему, введите данные о



вашей школе, тексты для подписей директора и учителя физического воспитания. Сохраните публикацию в собственной папке в файле с именем упражнение 4.4.2.pub.

3. Создайте в программе Publisher 2007 на основе шаблона Меню Особые блюда новую публикацию - меню праздничного ужина к дню рождения. Подберите самостоятельно цветовую схему, введите названия 5-6 блюд и их краткое описание. Сохраните публикацию в папке Мои документы в файле с именем упражнение 4.4.3.pub.
4. Создайте в программе Publisher 2007 на основе одного из шаблонов собственную визитную карточку. Включите в карточку ваше фото и данные о месте учебы. Распечатайте визитные карточки, разместив на одном листе 10 копий. Сохраните публикацию в папке Мои документы в файле с именем упражнение 4.4.4.pub.
- 5\*. Разработайте фирменный бланк вашей школы на основе шаблона Бланк Геометрия. Включите в бланк название школы, ее адрес и телефоны, эмблему (герб). Цвета подберите в соответствии с цветовой гаммой флага школы или флага региона. Сохраните публикацию в собственной папке в файле с именем упражнение 4.4.5.pub.

## 4.5. Особенности работы с текстовыми и графическими объектами в Publisher 2007. Создание бюллетеня и буклета



1. Что является основой дизайна?
2. Какие требования к цветовому оформлению вы знаете?
3. Объясните, что такое стили.
4. Как создать новую публикацию в Publisher 2007?

### Понятие о бюллетене

Как уже отмечалось, в публикациях могут размещаться объекты двух типов — тексты и графические изображения. Рассмотрим особенности работы с ними при создании публикаций в Publisher 2007. Сделаем это на примере создания информационного бюллетеня.



Бюллетень (франц. *bulletin* - записка, листок, документ) — короткое официальное сообщение, сборник официальных актов или периодическое издание, которое содержит сообщения по определенному кругу вопросов.

В Publisher 2007 бюллетень - это публикация из одной или нескольких страниц, которая содержит сообщения по определенному кругу вопросов, например анализ экономического или финансового положения предприятия за определенный период, подборка нормативных актов по определенному вопросу или за определенный период, обзор спортивных новостей, отчет о проведенных исследованиях и т. п.

Как правило, бюллетень имеет титульную и заключительную страницы. На каждой странице может быть несколько статей. Материал статей размещается в несколько колонок. Каждая колонка - это надпись, которую в Publisher 2007 еще называют текстовым полем. Текстовые поля одной статьи связаны между собой. Статьи могут начинаться на одной странице, а заканчиваться на другой.

Один из главных объектов публикации - заголовок бюллетеня. По своей структуре заглавие бюллетеня похоже на газетное. Оно содержит текст заглавия и данные об издателе, может содержать девиз, эмблему, для периодических изданий — номер и дату выпуска.

Основное место на титульной странице занимает передовая или главная статья. В бюллетене она играет роль своеобразной программной статьи - то есть статьи, в которой раскрыты цели данного издания или описаны проблемы, которые в следующих статьях будут обсуждаться, детализироваться и уточняться.

Последняя страница бюллетеня содержит данные о редакторах и дизайнерах бюллетеня, а также некоторые служебные данные: тираж публикации, реквизиты редакции и типографии и т. п.

## Создание бюллетеня на основе шаблона

Процесс создания бюллетеня, как и рассмотренный раньше процесс создания презентации, начинается с разработки структуры и дизайна данного вида публикации. Прежде всего следует ознакомиться с шаблонами публикаций данного типа, выяснить тематику и перечень вопросов, которые будут освещаться в публикации, подобрать материалы для публикации: название бюллетеня, заголовки и тексты статей, иллюстрации к ним, элементы оформления и т. п. Далее следует перейти к непосредственному созданию бюллетеня.

Для создания бюллетеня необходимо в начальном окне Publisher 2007 в списке типов публикаций выбрать Бюллетень. Потом выбрать шаблон бюллетеня и на соответствующей панели (рис. 4.34) установить значения свойств публикации: выбрать цветовую схему и схему шрифтов, указать служебные сведения. На этой самой панели в поле Размер страницы устанавливается количество листов публикации. Следует учитывать, что предусматривается двусторонняя печать, и если выбрать один лист публикации, то мы будем иметь две страницы бюллетеня, а если два листа - то четыре.

После завершения начальной установки значений свойств публикации следует выбрать кнопку Создать и перейти в режим редактирования и форматирования шаблона бюллетеня.

В Строке состояния окна программы отображаются ярлыки страниц публикации (рис. 4.35). Для просмотра нужной страницы публикации следует выбрать ярлык с ее номе-

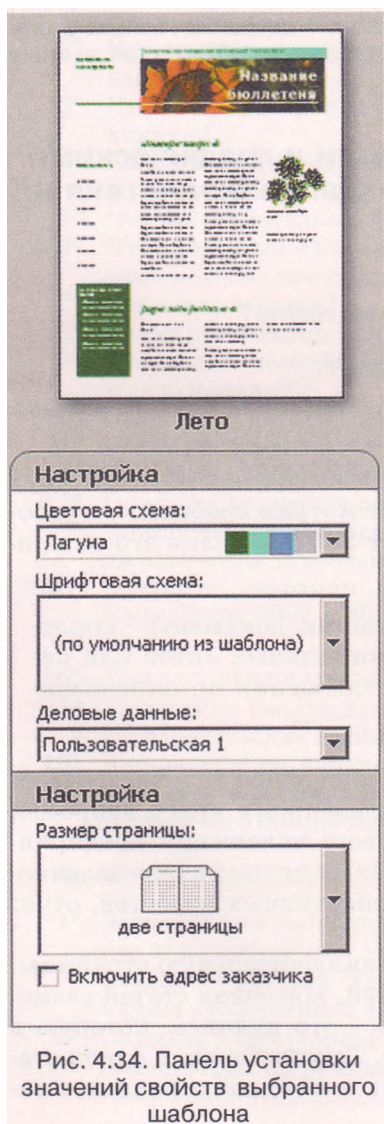




Рис. 4.34. Панель установки значений свойств выбранного шаблона

ром. Внутренние страницы (например, 2 и 3) отображаются попарно. Ярлык текущей страницы имеет более темный цвет.

### Настройка параметров страницы

Для настройки параметров страницы следует открыть, выбрав в области заданий **Форматирования публикации**, кнопку открытия панели  **Параметры страницы** (рис. 4.36). В поле **Столбцы** можно установить количество колонок текста на текущей или на всех страницах публикации. Для этого следует выбрать кнопку открытия списка (рис. 4.37) и команду, которая соответствует выбранной области применения. Выбор кнопки **Комбинация**  позволяет устанавливать разное количество столбцов для разных статей.

В списке **Цветовая схема области заданий Форматирования публикации** можно изменить выбранную цветовую схему публикации, а в списке **Шрифтовая схема** — указать новый набор шрифтов для заголовков и основного текста.

В списке **Бюллетень** — параметры этой же области заданий, кроме изменения шаблона публикации, можно установить и размеры страницы публикации. Для этого следует выбрать кнопку **Изменить размер страницы** и в окне **Параметры страницы** выбрать один из стандартных размеров или установить свои значения. В этом же окне устанавливают и размеры полей страницы.

### Редактирование заголовка, содержания и боковой панели

Структура заголовка бюллетеня стандартная для всех шаблонов. На рисунке 4.38 изображено оформление заголовка бюллетеня с шаблона *Лето*, а на рисунке 4.39 - измененный пользователем вариант заголовка этого же шаблона для бюллетеня *Альманах литературного творчества «Подснежник»*. Для создания заголовка на основе существующего шаблона следует ввести в соответствующие надписи новые тексты, заменить рисунки.

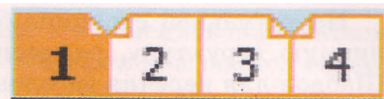


Рис. 4.35. Ярлыки страниц публикации

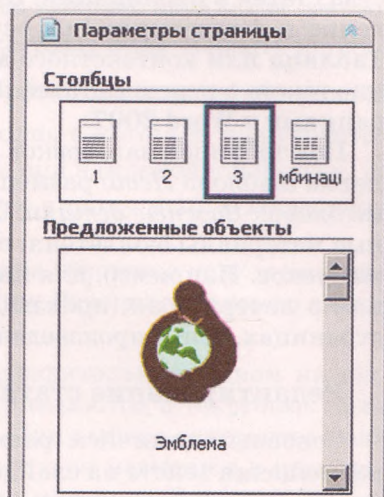


Рис. 4.36. Панель **Параметры страницы**

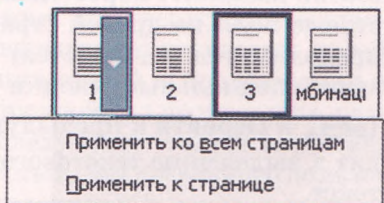


Рис. 4.37. Кнопки установки количества столбцов



Рис. 4.38. Заголовок бюллетеня с шаблона *Лето*

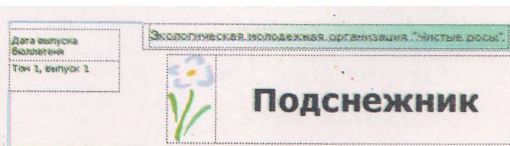


Рис. 4.39. Заголовок альманаха «Подснежник»

На титульной странице бюллетеня, особенно если он имеет многостраничную структуру, размещают перечень заголовков внутренних статей. Шаблон для перечня заголовков (рис. 4.40), как правило, размещается сбоку, рядом с первой или второй статьей бюллетеня. Хотя может быть и другое размещение, в зависимости от макета шаблона бюллетеня. Пользователь может изменить количество строк в перечне, в зависимости от количества статей в публикации. Этот объект является таблицей, и его редактирование и форматирование осуществляется с использованием команд меню Таблица или контекстного меню объекта. Редактирование и форматирование текста в перечне статей происходит в соответствии с аналогичными операциями в Word 2007.

Под таблицей заголовков статей на титульной странице бюллетеня с макетом шаблона *Лето* размещена Боковая панель (рис. 4.41), которая имеет заголовок *Важные детали*. Она, как правило, содержит ссылку на интересные материалы бюллетеня, сведения об авторах, анонсах статей следующих выпусков. Например, для литературного альманаха это могут быть сообщения о литераторах, произведения которых в первый раз появились на его страницах, или о произведениях уже известных и популярных авторов.

### Редактирование статей бюллетеня

Основным отличием размещения текста на страницах публикации от размещения текста на слайде презентации или в текстовом документе является возможность автоматического продолжения текста из одной колонки статьи в другую. Колонки с текстом в программе называют текстовыми полями, а автоматическое продолжение текста в следующих текстовых полях статьи называют перетеканием текста. Перетекание возможно также в текстовые поля на другой странице. Для подсказки, в каком текстовом поле продолжается (начинается) текст из выбранного текстового поля, используют специальные кнопки - *Перейти к следующему текстовому полю* и *Перейти к предыдущему текстовому полю*. Их выбор приводит к выделению текстового поля, в котором продолжается или начинается текст.

В шаблонах уже установлено автоматическое перетекание текста между текстовыми полями статьи. Если же пользователь самостоятельно создает макет публикации или вносит существенные изменения в существующий,

В этом выпуске:

<i>Карбань Леся</i> <i>Україна</i>	2
<i>Салимовская Инна</i> <i>«Только моя»</i>	2

Рис. 4.40. Шаблон таблицы для размещения заголовков статей




- Наши авторы:
- Ученица 10-Б класса  
Шаповал Ольга;
  - Ученица 10-А класса  
Королёва Надежда;
  - Ученица 11-В класса


Рис. 4.41. Боковая панель со списком авторов поэзий




то ему нужно уметь самостоятельно устанавливать и отменять перетекание текста из одного текстового поля в другое. Эта операция называется *связыванием текстовых полей*.

Перед началом выполнения связывания текстовых полей необходимо установить отображение панели инструментов Связать текстовые поля. Для этого следует выполнить Вид => Панели инструментов => Связать поля.

Для связывания текстовых полей необходимо выбрать начальное текстовое поле и на панели инструментов Связать текстовые поля выбрать кнопку Создать связь с надписью . Указатель приобретет вид чашки со стрелкой вниз . Его следует подвести к текстовому полю, с которым планируется установить связь (указатель должен приобрести вид наклоненной чашки, из которой выливаются буквы ) , и нажать левую кнопку мыши. Поле, с которым создается связь, должно быть пустым.

Для разрыва связи необходимо выбрать нужное текстовое поле и на панели инструментов Связать надписи выбрать кнопку Разорвать связь со следующей .

Тексты статей, как правило, готовятся предварительно в одном из текстовых процессоров, например Word 2007, и вставляются в текстовые поля публикации с использованием Буфера обмена. При выборе текстового поля шаблона выделяется весь текст статьи, даже если он размещен в нескольких колонках (надписях). И после выполнения команды Вставить новый текст заменяет шаблонный. Если текст не вмещается в одно или несколько связанных текстовых полей, отведенных для статьи в данном шаблоне, то программа откроет диалоговое окно (рис. 4.42) с соответствующим сообщением и запросом на выполнение автоперетекания. Подтверждение выполнения действия автоперетекания приведет к связыванию текстовых полей текущей статьи с текстовыми полями следующих статей и продолжения в них вставляемого текста. Пользователю будет предложено подтвердить размещение текста в текстовых полях одной из следующих статей шаблона, если они еще не заполнены текстом. Если же пустых, без введенного пользователем текста, нет, то программа предложит создать новые текстовые поля.

Если же пользователь откажется от выполнения операции автоперетекания, то под последним полем будет выведена метка Текст в области переполнения . Это значит, что часть текста не отображается на экране, но остается в памяти, и его можно будет воспроизвести.

Для того чтобы в текстовых полях статьи отображался весь текст, пользователю следует выполнить одно из действий:

- отредактировать текст, удалив несущественные его фрагменты;
- отформатировать текст, изменив значения свойств символов или абзацев;

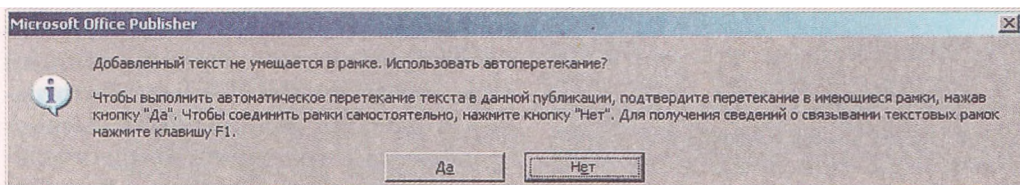


Рис. 4.42. Диалоговое окно с сообщением о переполнении текстового поля

- изменить размеры текстовых полей;  
связать текстовые поля статьи с другими на этой или другой странице.

Изменение размеров текстовых полей осуществляется с использованием маркеров изменения размеров выбранного текстового поля, а перемещение - стандартной операцией перетягивания после выбора границы поля.

Редактирование и форматирование графических объектов в публикациях Publisher 2007 осуществляется аналогично операциям в Word 2007 и PowerPoint 2007. Только если графические объекты из файлов в Word 2007 по умолчанию вставляются с обтеканием В тексте, а в PowerPoint 2007 - с обтеканием Перед текстом, то в Publisher 2007 — с обтеканием Вокруг рамки.

В ходе создания публикаций следует выполнять требования по соблюдению авторских прав. Все статьи и графические материалы должны иметь ссылку на авторов. При использовании материалов с других печатных или электронных публикаций следует получить разрешение на это их авторов.

### Создание буклета

Буклет (франц. *bouclette* - кольцо) - публикация, выполненная на одном листе и сложена сгибанием в несколько страниц так, что ее можно просматривать, не разрезая страниц, а раскрывая. В форме буклета изготавливаются путеводители, проспекты, программы концертов и представлений, рекламы продукции и услуг и т. п. Как правило, буклет состоит из одного листа формата А4, размещенного в альбомной ориентации и разделенного на три равные части. По границам этих частей буклет сгибается. Печать осуществляется, как правило, с обеих сторон листа.

Создание буклета в Publisher 2007, вставка текстовых и графических объектов не отличается от аналогичных операций с бюллетенем. Для создания буклета следует в начальном окне программы среди типов публикаций выбрать Буклет, а затем конкретный шаблон. Если пользователь работает с

основным окном программы, то необходимо для открытия типа публикации выполнить *Файл => Создать*.

Основные отличия в работе с буклетом определяются его назначением и формой.

Основное назначение буклета - реклама: учебного заведения, предприятия, организации, лица, нового товара, новой акции и т. п. А учитывая, что все данные должны быть размещены на одном листе, буклет должен содержать небольшие по объему фрагменты текста, которые точно передают основные данные об объекте, а также фотографии и рисунки, который иллюстрируют эти данные. Фотографии и рисунки должны быть высокого качества и содержать незначительное количество объектов, которые четко различаются.




Рис. 4.43. Примеры титульных страниц буклетов

Как правило, буклет, имеет титульную и заключительную страницы. На титульной странице указывают заголовок буклета, его основную тему и фотографию или рисунок из этой же темы. Могут указываться данные о том, кто выпустил этот буклет. На рисунке 4.43 представлены образцы оформления титульной страницы буклета.

На заключительной странице указываются контактные данные той организации или лица, которое выпустило этот буклет.

Примеры оформления буклетов можно просмотреть в Интернете. Для этого следует в строке поиска ввести ключевое слово *буклет* и выбрать команду (или гиперссылку, в зависимости от поисковой системы) **Изображения**.

### Проверьте себя

- 1°. Что такое бюллетень?
- 2°. Опишите последовательность создания бюллетеня в программе Publisher 2007.
- 3°. Как установить параметры страницы при работе с шаблоном бюллетеня?
- 4°. Как просмотреть содержание определенной страницы бюллетеня?
- 5°. Опишите последовательность установки разного количества столбцов текста на странице бюллетеня.
- 6°. Какие данные, как правило, отображаются в заголовке бюллетеня? Как внести изменения в заголовок?
- 7°. Для чего в бюллетене используют **Боковую панель**, что на ней размещают?
- 8°. Объясните, как изменить размеры или переместить столбец на другое место в публикации.
- 9°. Как добавить новую надпись в публикацию?
- 10°. Как определить, продолжается ли текст статьи в других надписях?
- 11\*. Что такое **перетекание текста** и как установить или отменить перетекание текста с одной надписи в другую?
- 12\*. О чем свидетельствует метка  возле столбца с текстом? Какие действия должен выполнить пользователь, чтобы ее убрать?
- 13\*. Объясните, что такое **автоперетекание** и в каких случаях оно используется.
- 14\*. Объясните, для чего выполняется группирование графических объектов и подписей под ними.
- 15°. Что такое буклет? Для чего он предназначен?
- 16\*. Опишите отличия между бюллетенем и буклетом. Для каких целей в вашей школе могут изготавливаться эти публикации?

### Выполните задания

- 1°. Запустите на выполнение программу **Publisher 2007**. Используя начальное окно программы, создайте новый бюллетень на основе шаблона *Школа*. Установите цветовую схему *Морская*, схему шрифтов *Архивная*, количество страниц - *одна*. Сохраните публикацию в собственной папке в файле с именем **упражнение 4.5.1.pub**.
- 2°. Создайте в программе **Publisher 2007** буклет на основе шаблона *Геометрия*. Установите цветовую схему *Плющ*, схему шрифтов *Стандартная 1*, количество страниц - *одна*. Вставьте в буклет форму *Бланк для ответов*. В заголовок бланка для ответов введите: *Укажите ваши контактные данные*, удалите надписи, которые касаются проведения мероприятий, цены и способов оплаты. Расположите заголовок бланка и шаблон для заполнения контактных данных так, чтобы они занимали нижнюю часть

той же страницы. Сохраните публикацию в собственной папке в файле с именем упражнение 4.5.2.pub.

- 3\*. Создайте в программе Publisher 2007 буклет, посвященный вашему классу. Шаблон, цветовую схему и схему шрифтов подберите самостоятельно. На титульной странице поместите фотографию школы и ее название, название населенного пункта и год издания. Сохраните публикацию в папке Мои документы в файле с именем упражнение 4.5.3.pub.
- 4\*. Создайте в программе Publisher 2007 на основе одного из шаблонов бюллетень, посвященный одному из писателей, творчество которого вы изучаете на уроках украинской или зарубежной литературы. Количество листов - *один*. Включите в бюллетень на титульной странице фото писателя и его короткую биографию, а в следующих статьях - отрывки из стихотворных или прозаических произведений, которые вам больше всего понравились. На последней странице укажите собственные данные как издателя. Сохраните публикацию в папке Мои документы в файле с именем упражнение 4.5.4.pub.
- 5\*. Откройте указанный учителем файл, например Тема 4\Задания 4.5\образец 4.5.5.pub. Используя материалы из файла Тема 4\Задания 4.5\образец 4.5.5.doc и фотографии весенней природы из файлов папки Тема 4\Задания 4.5, завершите создание бюллетеня *Альманах литературного творчества*.
- 6\*. Создайте в программе Publisher 2007 бюллетень, посвященный учебе в школе по вашему профилю. Подберите текстовый и иллюстративный материалы. Обоснуйте в статьях целесообразность изучения именно этого профиля. Сохраните публикацию в собственной папке в файле с именем упражнение 4.5.6.pub.

### Практическая работа № 16. Создание информационного бюллетеня и буклета

**Внимание!** Во время работы с компьютером соблюдайте правила безопасности и санитарно-гигиенические нормы.

1. Создайте, используя программу Publisher 2007, бюллетень на тему «Семь чудес света и Украины». Во время его создания можете использовать шаблон *Лист* или разработать собственный макет. Образцы первой и второй страниц бюллетеня (рис. 4.44) размещены в папке Тема 4\Практическая 16 в файлах Практическая 16\_1.jpg и Практическая 16\_2.jpg соответственно.
2. Для создания бюллетеня используйте текстовые и графические материалы из файлов папки Тема 4\Практическая 16.
3. Сохраните бюллетень в своей папке в файле с именем Практическая работа 16-1.pub.
4. Создайте буклет об одной из историко-культурных или естественных достопримечательностей Украины, которые вошли в перечень семи чудес Украины. В ходе его создания можете использовать шаблон *Затмение* или разработать собственный макет. Образцы первой и второй страниц буклета (рис. 4.45) размещены в папке Тема 4\Практическая 16 в файлах Практическая 16\_3.jpg и Практическая 16\_4.jpg соответственно.
5. Для создания буклета используйте текстовые и графические материалы из файлов папки **Тема 4\Практическая 16**.
6. Сохраните буклет в своей папке в файле с именем Практическая работа 16.2.pub.





а) б) Рис. 4.44. Образец оформления бюллетеня

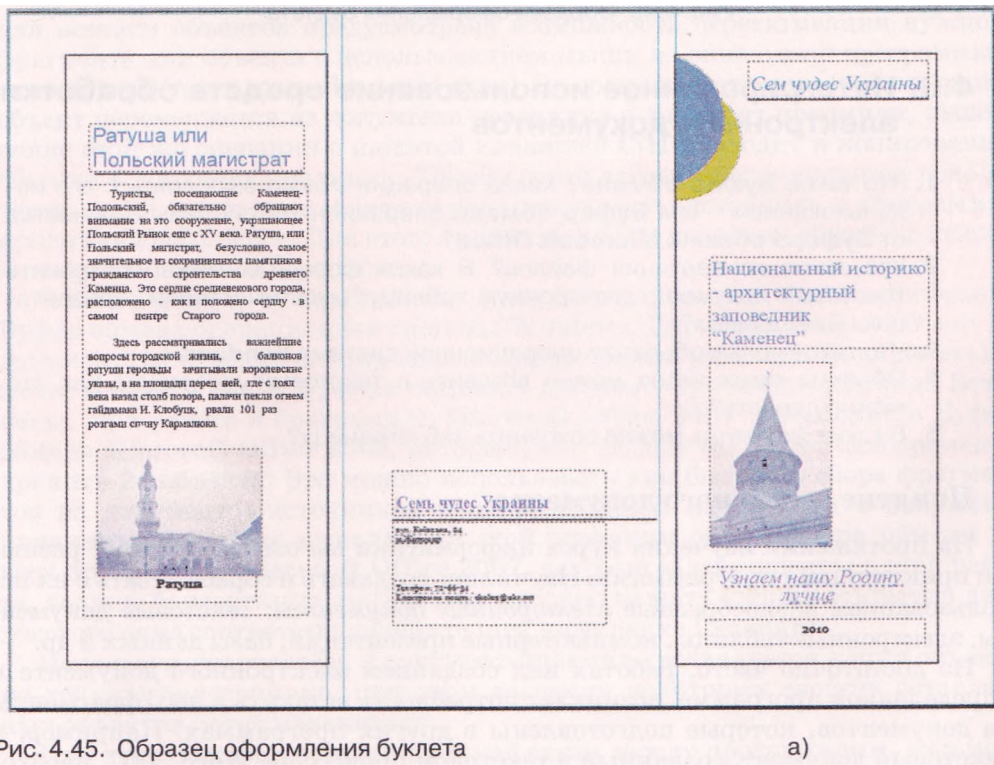


Рис. 4.45. Образец оформления буклета

а)

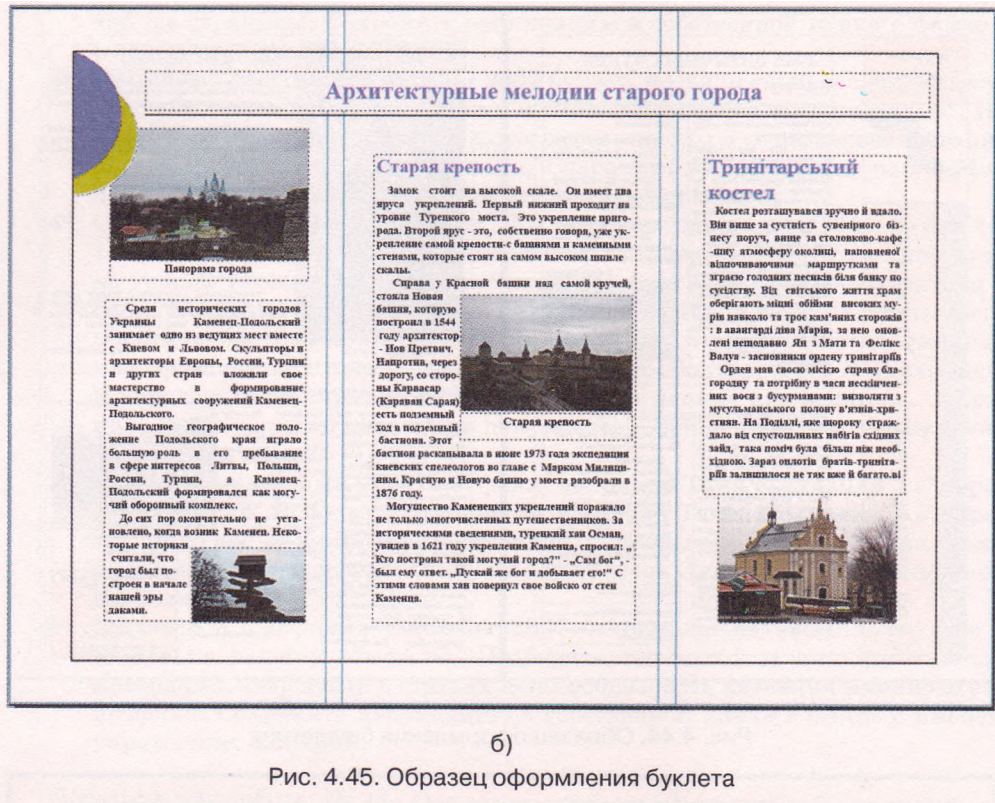


Рис. 4.45. Образец оформления буклета

## 4.6. Интегрированное использование средств обработки электронных документов



1. Что такое **Буфер обмена**? Какие операции можно выполнять с его использованием? Чем **Буфер обмена** операционной системы отличается от **Буфера обмена Microsoft Office**?
2. Что такое конвертация файлов? В каких форматах можно сохранить текстовый документ; электронную таблицу; мультимедийную презентацию; базу данных?
3. Что такое ярлык объекта в операционной системе **Windows**?
4. Объекты каких типов можно вставить в текстовый документ? Как это можно сделать?
5. В каких форматах можно сохранить веб-страницу?

### Понятие составного документа

На протяжении изучения курса информатики вы ознакомились с разными прикладными программами. Научились создавать и обрабатывать с их использованием разнообразные *электронные документы*: текстовые документы, электронные таблицы, компьютерные презентации, базы данных и др.

Но достаточно часто, работая над созданием электронного документа в определенной программе, возникает потребность вставить в него фрагменты из документов, которые подготовлены в других программах. Например, в текстовый документ, созданный в текстовом процессоре Word 2007, необхо-



можно вставить растровый рисунок, созданный в графическом редакторе Paint, а также фрагмент электронной таблицы, которая разработана в табличном процессоре Excel 2007, и таблицу из базы данных Access 2007. Либо, например, при заполнении базы данных в СУБД Access 2007 нужно использовать уже подготовленные данные из электронной таблицы Excel 2007, либо в мультимедийную презентацию нужно вставить видеоролики и аудиозаписи, которые были созданы предварительно. В результате таких действий пользователь получает документ, который включает в себя фрагменты, созданные с помощью различных программных сред. Подобный документ получил название *составного*, или *интегрированного*, документа.



*Составной (интегрированный, лат. *integrum* - целое) документ* — это документ, в который вставлены фрагменты с данными разных типов, созданные в различных прикладных программах.

Программа (документ, файл), из которой проводится вставка фрагмента в составной документ, называется *источник*, а программа (документ, файл), куда вставляют объекты из других источников, — *приемник*.

### Вставка фрагментов данных из разных программ

Операционная система Windows предоставляет пользователю несколько способов вставки данных из разных программ в составной документ. С некоторыми из них вы уже ознакомились раньше — это технология *drag-and-drop* (англ. *перетянуть и бросить*) и использование Буфера обмена.

Во многих программах, которые работают под руководством ОС Windows, для вставки объектов предусмотрена возможность перетягивания нужного фрагмента или объекта с использованием мыши из окна одной программы в окно другой (технология *drag-and-drop*). Во время выполнения этой операции объект перемещается из документа-источника в документ-приемник, выполнение этой же операции с нажатой клавишей Ctrl приводит к копированию объекта в документ-приемник. Такой способ вставки рекомендуется использовать, если расстояние между окнами программы-источника и программы-приемника небольшое. Для этого окна этих программ целесообразно расположить рядом в оконном режиме.

Также вставить нужный объект в документ-приемник можно, используя Буфер обмена операционной системы Windows. Для этого следует объект из документа-источника скопировать в Буфер обмена (команда Копировать), а затем вставить его из Буфера обмена в документ-приемник (команда Вставить). При работе в программах Microsoft Office 2007 используется Буфер обмена Microsoft Office 2007, который, как вы знаете, может одновременно хранить 24 объекта. Это можно использовать для быстрого сбора фрагментов из документов-источников и одновременной их вставки в документ-приемник. Во время выполнения этой операции окно Буфера обмена во всех программах Microsoft Office 2007, которые задействованы в этой работе, должно быть открытым (для этого нужно нажать кнопку открытия диалогового окна соответствующей группы на вкладке Главная).

Используя эти способы, в документ-приемник вставляется копия объекта из документа-источника, при этом значения некоторых свойств вставленных объектов могут измениться.

При этих способах вставки никакой связи между программами, документами и объектами не устанавливается. И в дальнейшем обработка вставлен-

ного объекта обеспечивается средствами программы, в которой осуществляется работа с составным документом. Например, вставив в текстовый документ рисунок из программы Paint, пользователь уже не сможет использовать для изменения рисунка инструменты графического редактора, такие как Кисть, Ластик, Распылитель, не сможет осуществить инвертирование цветов, выделение произвольной области и т. п. Он может воспользоваться только средствами, которые предоставляет текстовый процессор для работы с графическими изображениями.

Отсутствие связи между документом-источником и документом-приемником приводит к тому, что никакие изменения объекта-оригинала не отображаются в документе-приемнике. Например, если в текстовый документ вставлена диаграмма из электронной книги Excel, то она вставляется как рисунок и изменение числовых данных в документе-источнике никак не повлияет на изображение диаграммы, которая вставлена в составной документ.

### Технология OLE

Кроме перечисленных выше способов, которые не всегда являются целесообразными, вставить объекты в составной документ можно и способами, которые обеспечиваются технологией OLE.

Технология OLE (англ. *Object Linking and Embedding* - связывание и внедрение объектов) дает возможность вставлять объекты в составной документ, обеспечивая разные способы связи этого объекта с документом-источником и с программой-источником.



*Технология OLE* — технология вставки объектов разных типов в документы других типов.

Технология OLE предусматривает два способа вставки объектов в документ:

- *внедрение объекта* - обеспечивает вставку в документ-приемник копии объекта с автоматическим установлением связи с программой-источником, но связь с документом-источником теряется. При таком способе никакие изменения оригинального объекта не влияют на вид его копии в составном документе, но для работы со вставленным объектом можно пользоваться не только средствами программы-приемника, но и средствами программы-источника.
- *связывание объекта* — в документ-приемник вставляется не сам объект, а только ссылка на него в виде образа этого объекта с автоматическим установлением связи и с документом-источником, и с программой-источником. При таком способе любые изменения объекта в документе-источнике приводят к автоматическому изменению объекта в документе-приемнике. Для обработки объекта можно использовать средства программы-источника.

Отличить во время просмотра и печати составного документа, с помощью какого способа вставлены объекты в документ, почти невозможно (кроме вставленных в виде значка). Способ вставки объектов влияет только на возможность использования различных средств для обработки вставленных объектов (как было указано выше) и на размер составного документа. Поскольку встроенные объекты располагаются внутри документа-приемника,



то размер файла составного документа возрастает за счет размера этих объектов. При связывании объектов размер составного документа почти не увеличивается, поскольку в документ вставляется не сам объект, а ссылка на него, которая занимает очень мало места.

Технологию OLE в полном объеме поддерживают не все прикладные программы. Некоторые из них можно использовать только как приемник, другие - только как источник, но есть и такие программы, которые могут выступать в обеих ролях. Например, прикладные программы пакета Microsoft Office 2007 полностью обеспечивают работу по технологии OLE, а графический редактор Paint — только как источник объектов.

## Внедрение и связывание объектов в программах Microsoft Office 2007

Внедрение или связывание объектов в программах Microsoft Office 2007 из открытого документа-источника осуществляется через Буфер обмена командой Специальная вставка и выполняется по такому алгоритму:

1. Выделить в документе-источнике объект (фрагмент документа), который вставляется, и выполнить *Главная => Буфер обмена => Копировать*.
2. Разместить курсор в нужном месте документа-приемника и выполнить *Главная Буфер обмена => Вставить => Специальная вставка*.
3. Указать в диалоговом окне Специальная вставка способ вставки объекта, выбрав нужный переключатель:
  - для внедрения - переключатель Вставить;
  - для связывания - переключатель Связать (рис. 4.46).
4. Выбрать в списке Как тип объекта, в формате которого объект-копия будет вставлен в документ-приемник.

В дальнейшем в зависимости от выбранного типа пользователю будут предоставляться соответствующие средства обработки объекта-копии. Набор возможных вариантов для выбора типа объекта зависит от типа самого объекта-оригинала. Для вставки объекта с теми же свойствами, которые он имеет в документе-источнике, рекомендуется выбирать тип *Объект*.

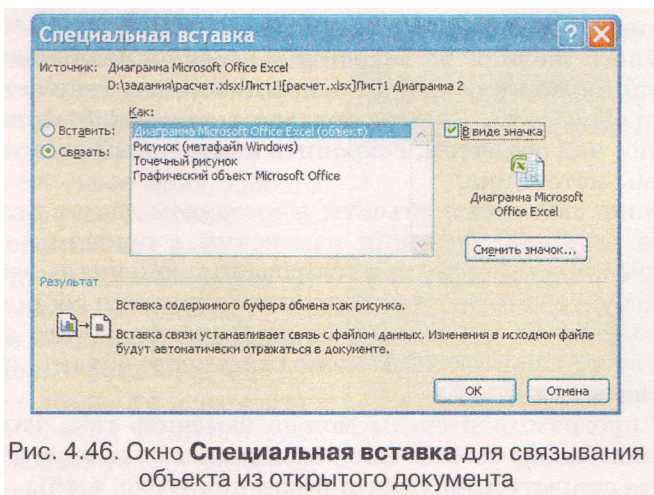


Рис. 4.46. Окно **Специальная вставка** для связывания объекта из открытого документа

5. Выбрать способ отображения объекта-копии в документе-приемнике, установив при необходимости метку флажка В виде значка:
  - метка флажка не установлена - объект вставляется в таком же виде, что и в документе-источнике;
  - метка флажка установлена - в документ вместо объекта вставляется значок данного типа объекта (его можно изменить, выбрав кнопку Сменить значок), который является ссылкой на оригинальный объект. Для открытия объекта-оригинала нужно будет дважды щелкнуть на вставленном значке.
6. Выбрать кнопку ОК.

Используя связывание объектов, следует помнить:

- программа, которая была использована для создания объекта-оригинала, должна быть обязательно установлена на том компьютере, где осуществляется просмотр и обработка составного документа;
- файлы документа-источника и документа-приемника нужно предварительно сохранить;
- при изменении места записи составного документа или документов-источников может произойти *разрыв (потеря) связей* и тогда в составном документе вставленные объекты не будут отображаться.

### Обработка внедренных и связанных объектов в программах Microsoft Office 2007

Работа с внедренными и связанными объектами в составном документе имеет свои отличия.

При двойном щелчке на *внедренном объекте* в окне программы-приемника откроется дополнительное окно, в котором отобразится внедренный объект и средства этой программы, с помощью которых можно осуществлять обработку объекта.

На рисунке 4.47 представлен внешний вид окна программы Word 2007, в котором планируется обработка диаграммы табличного процессора Excel 2007, встроенной в текстовый документ. Как видно, на Ленте появились новые вкладки *Формулы*, *Данные*, *Средства для диаграмм*, с помощью которых и будет осуществляться редактирование диаграммы. После завершения работы для возвращения в окно составного документа следует выбрать произвольное место вне внедренного объекта.

После двойного щелчка на *связанном объекте* открывается окно программы-источника, в котором будет открыт весь документ-источник для последующей его обработки. Дальше пользователю нужно выполнить необходимые операции над объектом, сохранить измененный документ и закрыть окно программы-источника.

По умолчанию связанные объекты в составном документе обновляются автоматически во время внесения изменений в оригинал объекта. Если документ-приемник на момент редактирования оригинала закрыт, то изменения в копию будут внесены при первом же открытии составного документа. Если обновление не происходит автоматически, то это можно сделать вручную. Для этого в контекстном меню связанного объекта нужно выбрать команду *Обновить связь*.

Настройку программной среды можно изменить так, чтобы связанные объекты не обновлялись вообще или обновлялись только по запросу пользователя во время открытия документа-приемника (рис. 4.48).

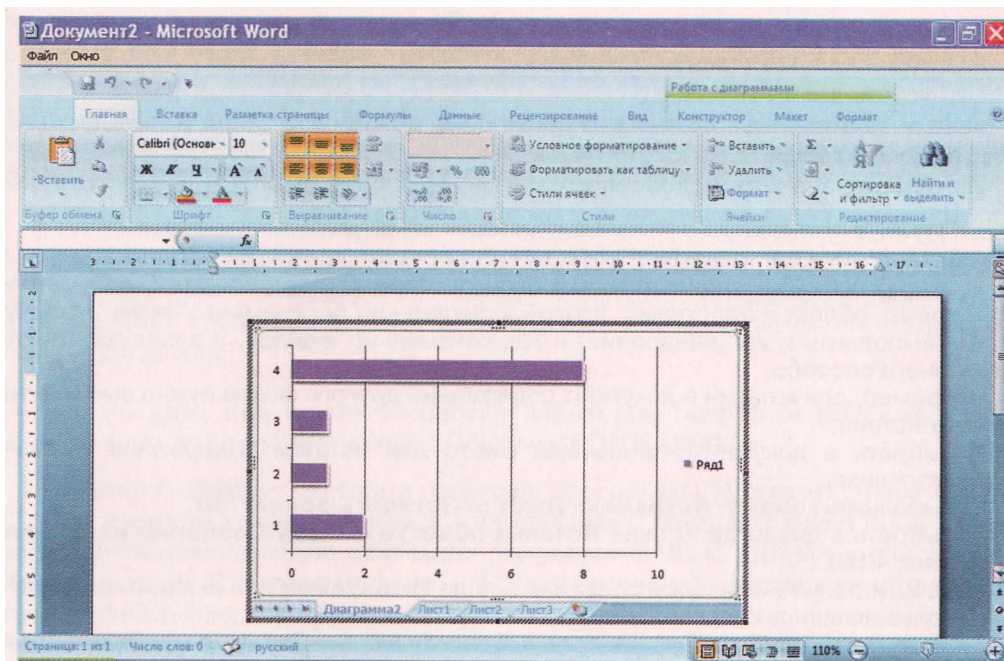


Рис. 4.47. Дополнительное окно программы-источника во время обработки внедренного объекта

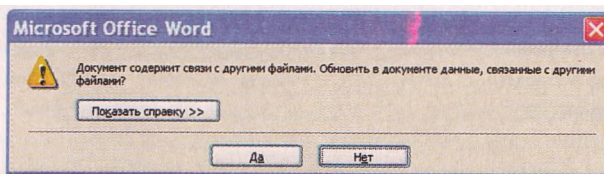


Рис. 4.48. Запрос пользователю об обновлении связанных объектов при открытии составного документа

Также пользователю нужно контролировать связи документов на случай их разрыва. Все изменения связей составного документа, их удаление, просмотр, возобновление или настройку можно осуществить в диалоговом окне **Связи** (рис. 4.49), которое можно открыть двумя способами:

открыть контекстное меню любого связанного объекта и выполнить **Связанный объект => Связи**;  
 выполнить **Office => Подготовить => Изменить ссылки на файлы**.

В окне **Связи** содержится перечень файлов, с которыми установлены связи для данного составного документа. При выборе из списка нужного связанного файла можно узнать значения таких его свойств:

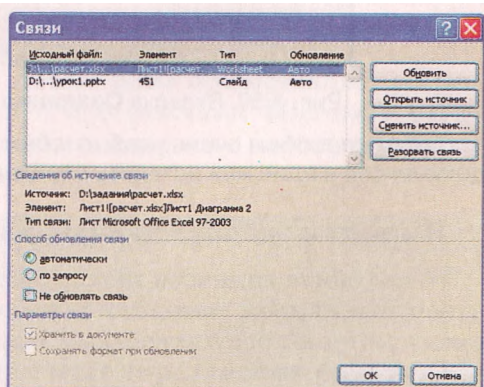



Рис. 4.49. Окно **Связи**




- полное имя файла-источника;
- вид объекта в документе-источнике;
- тип установленной связи;  
режим обновления объекта в составном документе - *автоматическое* или *по запросу*.

Изменение значений свойств можно осуществить выбором соответствующих кнопок и переключателей в этом диалоговом окне.

 Технология **OLE** также дает возможность вставить в документ содержание некоторого файла, созданного в другой программе, или создать в документе новый объект средствами другой прикладной программы. Такую вставку можно выполнить и как внедрение, и как связывание файлов, в зависимости от выбранного способа.

Например, для вставки в документ содержания другого файла нужно выполнить такие действия:

1. Выбрать в документе-приемнике место для вставки содержания файла-источника.
2. Выполнить команду *Вставка => Текст => Вставить объект* .
3. Выбрать в диалоговом окне **Вставка объекта** вкладку **Создание из файла** (рис. 4.50).
4. Ввести полное имя нужного файла в поле **Имя файла** или выбрать его, воспользовавшись кнопкой **Обзор**.
5. Установить при необходимости метки флажков **Связь с файлом** (для связывания файлов) и **В виде значка**.
6. Выбрать кнопку **ОК**.

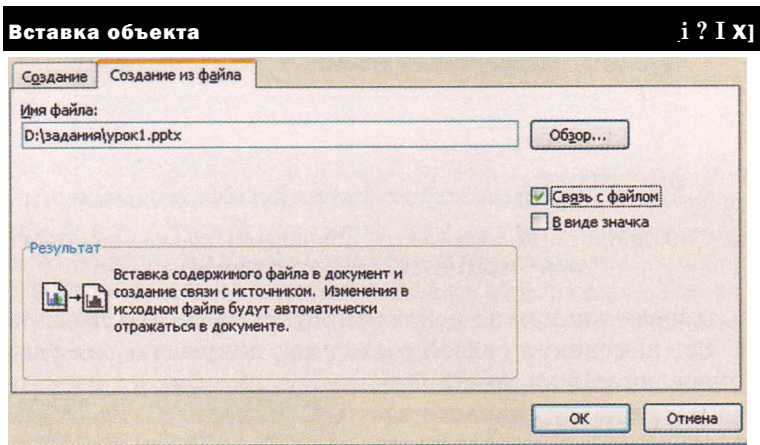


Рис. 4.50. Вкладка **Создание из файла** окна **Вставка объекта**

Таким способом очень удобно и быстро можно объединить несколько текстовых документов в один или вставить слайды презентации в текстовый документ.

### Импорт и экспорт данных из файлов

Часто обмен данными между разными программами можно осуществить с использованием операций *импорта* и *экспорта*, если такие операции предусмотрены возможностями этих программ.

Операция импорта предусматривает считывание программой данных из файлов, подготовленных в других программах и формат которых не является основным для данной программы. После открытия таких файлов (*импор-*



жирование данных) программа обрабатывает полученные данные как стандартные для своей работы. Экспорт заключается в сохранении данных в файле, формат которого не является основным для данной программы. В сущности, во время осуществления операций импорта и экспорта происходит *конвертация данных*, при этом превращение данных может происходить с частичной потерей данных, элементов форматирования и т. п.



**Импорт** (лат. *importo* — ввоз) файла — открытие в программе файла, созданного в другой программной среде.

**Экспорт** (лат. *export* - вывоз) данных — сохранение данных в файле, предназначенном для использования в какой-либо другой программе.

Рассмотрим, как осуществляются некоторые операции импорта и экспорта в прикладных программах Microsoft Office 2007.

#### Пример 1. Импорт тестовых файлов в программы Microsoft Office 2007.

Для импорта текстовых файлов в программу Microsoft Excel 2007 нужно выполнить стандартную операцию открытия файла, выбрав в списке Тип файла формат *Текстовые файлы*. Импортированию подлежат только текстовые файлы формата TXT.

После этого запускается *Мастер текстов (импорт)*, выполняя шаги которого, нужно указать, используются ли разделители в начальном документе, какой именно знак-разделитель использован, какое кодирование символов выбрано, какой тип и формат отдельных столбцов данных нужно установить и т. п. В итоге текст будет конвертирован и вставлен в Excel 2007 в виде таблицы.

Примерно так же осуществляется импортирование текстовых данных в программе Access 2007, только мастер импорта в данном случае имеет название *Связь с текстовым файлом* и при его работе дополнительно можно указывать язык и способ кодирования импортируемого текста.

В программу Power Point 2007 импортировать можно текстовые файлы разных форматов (TXT, RTF, DOCX). В итоге будет создана презентация из некоторого количества слайдов, на каждом из которых *как заголовок* будет вставлен текст отдельных строк импортируемого документа.

В Publisher 2007 импортировать текстовые данные можно двумя способами - открыть текстовый файл выбранного формата (TXT, RTF, DOCX) или импортировать документ Word.

#### Пример 2. Импорт и экспорт файлов в Access 2007.

Система управления базами данных Access 2007 предоставляет особенные возможности для обмена данными с программами Microsoft Office 2007. Все необходимые для этого инструменты находятся на вкладке **Внешние данные** (рис. 4.51).

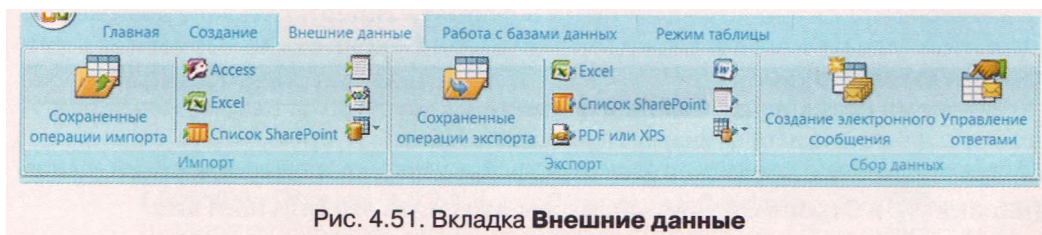









Рис. 4.51. Вкладка **Внешние данные**

Используя группу элементов управления Импорт, можно в текущую БД импортировать данные из другой базы данных Access 2007 (кнопка ) , из электронной таблицы Excel 2007 (кнопка ) , из текстового документа формата RTF (кнопка ) . Можно экспортировать данные (группа Экспорт) из текущей БД в электронную таблицу (кнопка ) или в текстовый документ формата Word 2007 (кнопка ) и др.

Например, чтобы импортировать в текущую БД Access 2007 данные из книги Excel 2007, нужно на вкладке Внешние данные в группе Импорт выбрать кнопку Excel  . Далее следовать указаниям мастера, который дает возможность импортировать в Access 2007 содержание выбранного рабочего листа или именованного диапазона ячеек.

Чтобы экспортировать данные из Access 2007 в Excel 2007, в окне базы данных необходимо выбрать таблицу, запрос, форму или отчет, а затем выполнить *Внешние данные => Экспорт => Excel*  . Программа сохранит объект базы данных в виде файла электронной книги в выбранной папке. Имя файла по умолчанию будет совпадать с именем сохраненного объекта базы данных.

Аналогично происходит экспортирование данных и в текстовый документ.

### Пример 3. Экспорт данных в программе Power Point 2007.

Созданную презентацию можно экспортировать в текстовый документ, выполнив *Office => Опубликовать => Создать выдачи в Microsoft Office Word* . При этом можно выбрать два способа вставки слайдов в текстовый документ:

в виде *выдач*, когда каждый слайд презентации вставляется в текстовый документ в виде графического эскиза слайда, рядом с которым отводится дополнительное место для заметок. Количество слайдов на листе и место расположения заметок указывают дополнительно;


в виде *структуры* презентации, когда текстовые данные с каждого слайда размещаются в документе в виде последовательного текста.

Также удобным видом экспортирования является сохранение презентации в виде набора графических изображений, где каждый слайд представлен отдельным файлом заданного формата (*Office => Сохранить как* и выбрать тип файла *Рисунок*) . Все изображения слайдов будут собраны в одну папку, имя которой будет совпадать с именем файла презентации.

### Веб-публикация документов

Чтобы быстро подготовить данные из программ Microsoft Office для публикации в Интернете, не обязательно знать теги языка разметки HTML, достаточно сохранить созданный файл в формате гипертекстового документа. Для этого следует воспользоваться командой Сохранить как и выбрать тип файла *Веб-страница* .

Во время сохранения файла в формате веб-страницы неминуемо теряются некоторые элементы форматирования. Поэтому, прежде чем выполнять экспорт данных, следует просмотреть документ в режиме Веб-документ

(кнопка  в Строчке состояния), чтобы выяснить его будущий вид.

Следует помнить, что вставленные в начальный документ графические изображения будут сохранены в отдельной папке в отдельных файлах, а при сохранении презентации в формате веб-страницы каждый слайд будет сохранен в отдельном файле.

Полученную в результате конвертации данных веб-страницу можно просматривать, используя програму-браузер, и в дальнейшем ее можно разместить в сети Интернет.

### Проверьте себя

- 1°. Какой документ называют составным?
- 2°. Приведите примеры задач, которые решаются с помощью нескольких программ обработки электронных документов.
- 3°. Объекты каких типов можно вставить в текстовый документ; в мультимедийную презентацию; в электронную таблицу?
- 4°. Назовите способы, которыми можно вставить объекты в электронный документ. Объясните их суть.
- 5°. Какой документ во время вставки объектов называют источником, а какой - приемником?
- 6°. Объясните, как осуществляется вставка фрагментов документов и отдельных объектов с использованием Буфера обмена Microsoft Office 2007.
- 7°. В чем заключается суть технологии OLE? Какое ее назначение?
- 8°. Объясните, чем внедрение объекта отличается от связывания объекта.
- 9°. Объясните, чем вставка объектов через Буфер обмена отличается от вставки объекта по технологии OLE.
- 10°. Какая ситуация называется разрывом (потерей) связи? Когда это может произойти? Как исправить такую ситуацию?
- 11\*. В корпоративных компьютерных сетях создают некоторые стандартизированные объекты (бланки документов, логотипы, шаблоны, заготовки файлов и др.), которые используются в качестве связанных объектов всеми сотрудниками компании. В чем преимущество такого способа использования стандартизированных объектов? Какие возможные неудобства при этом?
- 12\*. Во время вставки диаграммы в публикацию вместо объекта в документе появилась только его пиктограмма. Объясните, почему так произошло и как исправить эту ситуацию.
- 13\*. Три фрагмента одного документа были сохранены в разных файлах. Как их быстро объединить?
14. Чем отличается редактирование связанных объектов от внедренных?
15. В чем суть операций импортирования и экспортирования данных?
16. В чем заключается веб-публикация документа?


### Выполните задания

1. Сравните два способа вставки объектов в электронные документы, заполнив таблицу:


<i>Характеристика .</i>	<i>Внедрение объекта</i>	<i>Связывание объекта</i>
<b>Размер файла составного документа</b>		
<b>Возможность потери связи</b>		
<b>Средства для редактирования</b>		

2. Откройте указанные учителем файлы (например, из папки Тема 4\Задания 4.6 файлы Образец 4.6.2-1.docx, Образец 4.6.2-2.docx). Просмотрите их и сравните содержание. Сравните размеры этих файлов. Объясните отличие.


<i>Имя файла</i>	<i>Размер файла</i>	<i>Способ вставки объектов</i>
<b>Образец 4.6.2-1.docx</b>		
<b>Образец 4.6.2-2.docx</b>		

 3°. Укажите преимущества и недостатки внедрения и связывания объектов.

<i>Характеристика</i>	<i>Внедрение объекта</i>	<i>Связывание объекта</i>
Преимущества		
Недостатки		


 4°. Откройте по очереди указанные программы пакета **Microsoft Office** и определите возможные форматы электронных документов для экспорта и импорта данных.

<i>Программа</i>	<i>Форматы файлов для экспорта данных</i>	<i>Форматы файлов для импорта данных</i>
<b>Word 2007</b>		
<b>Excel 2007</b>		
<b>Power Point 2007</b>		
<b>Access 2007</b>		
<b>Publisher 2007</b>		

 5°. Откройте указанные учителем файлы - текстовый документ и электронную книгу (например, из папки **Тема 4\Задания 4.6** файлы **текст.docx** и **таблица.xlsx**). Вставьте в текстовый документ диаграмму из электронной книги с использованием **Буфера обмена**. Сохраните текстовый документ в собственной папке в файле с именем **упражнение 4.6.5.docx**. Отредактируйте произвольным образом диаграмму в файле электронной книги и просмотрите текстовый документ. Произошли ли какие-либо изменения в текстовом документе? Объясните результат.

6°. Откройте указанные учителем файлы - мультимедийную презентацию и рисунок (например, из папки **Тема 4\Задания 4.6** файлы **рисунок.Бтп** и **презентация.pptx**). На слайд № 1 вставьте рисунок через **Буфер обмена**, на слайд № 2 внедрите рисунок, на слайде № 3 свяжите рисунок. Сохраните презентацию в собственной папке в файле с именем **упражнение 4.6.6.pptx**. Отредактируйте рисунок в документе-источнике. Просмотрите презентацию и объясните полученный результат.

7°. Откройте указанный учителем файл текстового документа (например, **Тема 4\Задания 4.6\текст 4.6.7.docx**). Попробуйте выполнить редактирование вставленных в него объектов. Укажите, каким способом были вставлены эти объекты в документ. Объясните, почему вы так считаете.

 8°. Откройте мультимедийную презентацию, указанную учителем (например, **Тема 4\Задания 4.6\образец 4.6.8.pptx**). Сохраните файл в форматах HTML, JPG, RTF в собственной папке. Опубликуйте презентацию в виде файла выдач. Просмотрите созданные файлы.

9°. Откройте указанный учителем файл текстового документа (например, файл **Тема 4\Задания 4.6\текст 4.6.9.docx**). Просмотрите таблицу связей данного документа. Удалите все связи в этом документе. Сохраните файл в форматах HTML, RTF, TXT в собственной папке. Просмотрите созданные файлы.



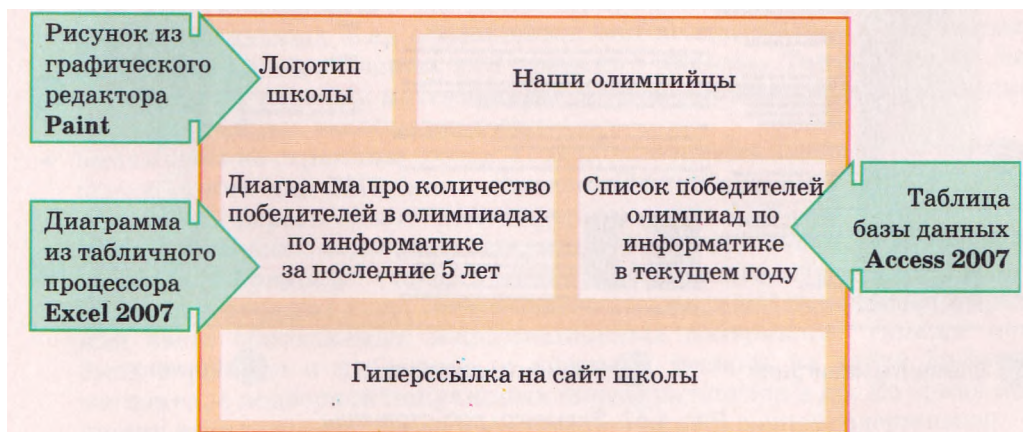
10. Создайте базу данных **Задание 4.6.10** и импортируйте в нее электронную таблицу (например, **Тема 4\Задания 4.6\таблица 4.6.10.xlsx**). Сохраните созданный файл в собственной папке в файле с именем **упражнение 4.6.10.accdb**.
11. Откройте указанный учителем файл базы данных (например, **Тема 4\Задания 4.6\база 4.6.11.accdb**). Экспортируйте таблицу базы данных в электронную таблицу, а отчет - в текстовый файл. Сохраните созданные файлы в собственной папке с именами **упражнение 4.6.11.xlsx** и **упражнение 4.6.11.docx**.
- 12\*. Импортируйте указанный текстовый документ (например, **Тема 4\Задания 4.6\текст 4.6.12.rtf**) в базу данных и электронную таблицу. Сохраните созданные файлы в собственной папке с именами **упражнение 4.6.12.xlsx** и **упражнение 4.6.12.accdb**.

### Практическая работа № 17. Обработка данных в нескольких программных средах

**Внимание!** Во время работы с компьютером соблюдайте правила безопасности и санитарно-гигиенические нормы.

1. Откройте в графическом редакторе **Paint** рисунок из файла **Тема 4\Задания 4.6\Практическая 17\Цветок.jpg**.
2. Откройте в табличном процессоре **Excel 2007** диаграмму из файла **Тема 4\Задания 4.6\Практическая 17\Количество.xlsx**.
3. Откройте в СУБД **Access 2007** таблицу базы данных из файла **Тема 4\Задания 4.6\Практическая 17\Олимпийцы.accdb**.
4. Откройте программу **Power Point 2007** и создайте слайд, на который вставьте объекты из открытых документов (согласно рисунку): графическое изображение как внедренный объект, диаграмму как связанный объект, список учеников как объект, вставленный через **Буфер обмена**.

Слайд Microsoft PowerPoint



5. Сохраните презентацию в собственной папке в файлах с именем **практическая 17** в форматах PPTX, RTF, JPG, HTML.
6. Экспортируйте базу данных **Олимпийцы.accdb** в электронную таблицу и текстовый документ.
7. Сохраните созданную электронную таблицу и текстовый документ в собственной папке в файлах с именами **Олимпийцы.xlsx** и **Олимпийцы.txt**.
8. Закройте окна всех программ.

## 4.7. Типы веб-страниц. Классификация веб-сайтов



1. Какой документ называют составным?
2. Что такое веб-страница, веб-сайт, веб-портал? В чем состоит отличие между этими понятиями?
3. Какие языки используются для создания веб-страниц?
4. Сообщения каких видов по способу представления могут содержаться на веб-страницах?
5. Из чего состоит доменное имя Интернет-ресурса? Какова структура URL-адреса?

## Типы веб-страниц

Примером составного документа является веб-страница, поскольку она может включать данные разных типов: графические изображения, анимацию, видео и музыкальные фрагменты. Веб-страницы являются информационными ресурсами службы World Wide Web, и в 9-м классе вы ознакомились с тем, как осуществлять поиск и просмотр этих ресурсов. Теперь ознакомимся со способами их создания. При создании веб-страниц важно учитывать их структуру и типы, а также виды сайтов, в состав которых будут входить эти страницы.

Рассмотрим пример веб-страницы (рис. 4.52). Среди элементов, которые на ней отображаются, можно выделить такие группы:

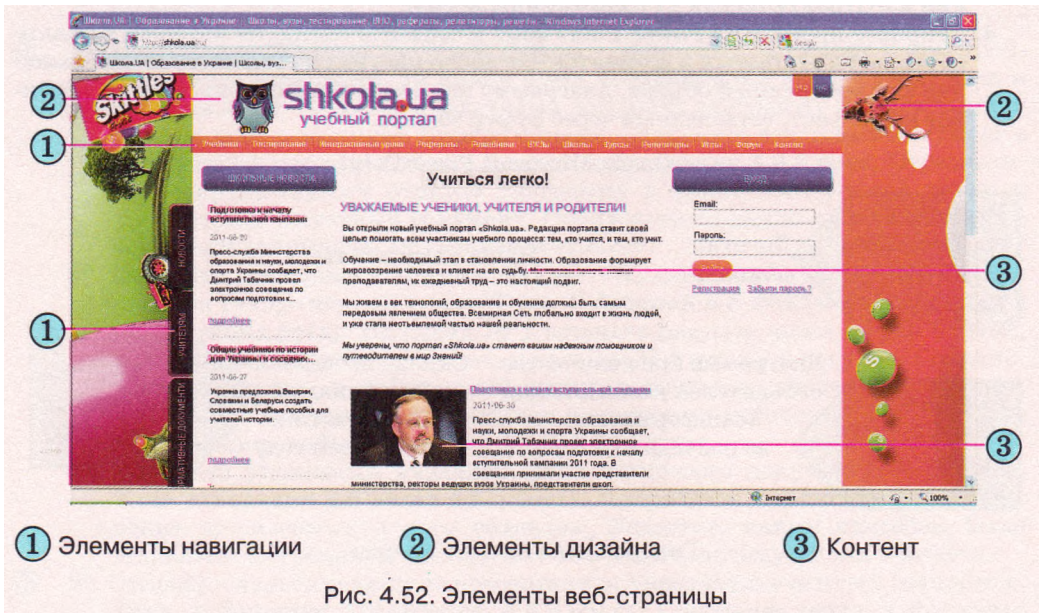


Рис. 4.52. Элементы веб-страницы

**контент** (англ. *content* - содержание) - содержательное наполнение веб-страницы, доступное пользователю: тексты, изображения, видео-, звуковые данные и др.;

**элементы навигации** - средства для перехода на другие веб-страницы;

**элементы дизайна** — элементы структуризации контента и его форматирования, оформление страницы.

В зависимости от назначения можно выделить такие типы веб-страниц: главная (домашняя) страница - страница, с которой начинается просмотр веб-сайта при переходе на сайт по его URL-адресу. На ней, как правило, раскрывается тематика сайта, его назначение, приводятся данные о разработчиках, описывается, какие материалы можно найти на других страницах сайта;

информационные страницы (страницы тематических разделов) содержат тексты, изображения и сообщения других видов, которые раскрывают тему сайта или некоторого его раздела. Главная страница сайта является частным случаем информационной страницы;

страницы-контейнеры содержат списки ссылок на ресурсы данного или других сайтов:

- веб-каталоги - ссылки на веб-ресурсы (веб-страницы, веб-сайты);
- каталоги файлов - ссылки на файлы, которые могут быть загружены пользователем;
- коммуникационные (интерактивные) страницы предназначены для предоставления пользователям сайта средств общения и обратной связи с разработчиками сайта:
  - страницы форума предназначены для организации общения на форуме;
  - страницы чата предназначены для организации общения в чате;
  - страницы гостевой книги предназначены для предоставления возможности посетителям сайта оставлять свои комментарии, как правило, для авторов сайта;
  - страницы формы предназначены для проведения опросов, определения точки зрения посетителей, осуществления выбора товаров или услуг на коммерческих сайтах и др.

В зависимости от технологий, использованных при создании веб-страниц, можно выделить такие типы страниц:

- статические страницы созданы, как правило, средствами языка разметки гипертекста HTML. Содержимое страниц остается неизменным для всех посетителей сайта. На таких страницах не предполагается частое изменение данных и их оформления, они не нуждаются в специальном программном обеспечении для хранения данных. Такими могут быть страницы с учебными материалами, историческими сведениями, описаниями музейных экспонатов и др.;
- динамические страницы созданы с использованием языков программирования, таких как PHP (англ. *Personal Home Page Tools* - инструменты разработки персональных домашних страниц), ASP (англ. *Active Server Pages* - активные серверные страницы), PERL (англ. *Practical Extraction and Report Language* - практический язык для извлечения данных и составления отчетов) и др. Такие страницы предназначены для отображения часто обновляемых информационных материалов. Такими могут быть страницы с новостями, с перечнем товаров на сайте Интернет-магазина, с подборкой популярных видеофрагментов и др. Во время посещения таких страниц пользователи могут видеть разное содержимое;
- флэш-страницы (англ. *flash* - вспышка) создаются с использованием технологии разработки анимационных изображений Adobe Flash, а потому такие страницы привлекательные, яркие, содержат много анимации и звуковых эффектов. Вид веб-страницы изменяется в результате воспроизведения флэш-анимации и в зависимости от положения указателя. Нетипичный дизайн привлекает посетителей, поэтому флэш-



страницы часто размещаются на сайтах дизайнеров, художников, фотографов и т.п., работы которых эффектнее выглядят в динамическом оформлении. В то же время флэш-страницы трудно изменять, поэтому для них почти не предполагается обновление данных и других элементов. Объемы файлов, в которых сохраняются флэш-страницы, очень большие, такие страницы медленно загружаются. Для их просмотра должны быть установлены плагины для воспроизведения флэш-анимации.

Классификация веб-страниц может быть представлена схемой, приведенной на рисунке 4.53.



Рис. 4.53. Схема классификаций веб-страниц

## Классификация веб-сайтов

Разные веб-сайты могут иметь разную структуру - количество веб-страниц и их типы, внутреннюю тематическую организацию, совокупность внутренних связей. Общим для всех сайтов является наличие в структуре главной страницы, которая связана со всеми тематическими разделами сайта. Разделы могут иметь собственные начальные страницы, которые связаны с главной страницей и информационными страницами. Количество веб-страниц и их типы определяются объемом и характером материалов сайта.

Классифицировать сайты можно по значениям разных свойств. Приведем несколько возможных классификаций.

В соответствии с основной технологией, использованной для создания веб-страниц, веб-сайты делятся на статические (все страницы статические), динамические (все страницы динамические), флэш-сайты (все страницы построены по флэш-технологии) и смешанные.

По назначению выделяют такие типы сайтов:

сайты, предоставляющие информационные материалы: информационно-тематические сайты, сайты новостей, электронные библиотеки,



энциклопедии, словари, каталоги, хранилища файлов разных видов, медиатеки и др.;

• сайты для он-лайн контактов и общения: форумы и веб-чаты, сайты социальных сетей, блоги, сайты знакомств, сетевых игр и др.;

сайты электронной коммерции: Интернет-магазины и аукционы, системы электронных платежей, сайты банков, бирж, сайты, предлагающие разнообразные платные услуги - изучение иностранных языков, консультации психолога и др.;

• сайты он-лайн сервисов: сайты электронной почты, поисковых систем, перевода слов и текстов. Кроме того, он-лайн сервисами, для которых созданы соответствующие сайты, являются сервисы:

- предоставления хостинга (англ. *hosting* - оказание гостеприимства) - выделение аппаратных и программных ресурсов сервера для размещения файлов пользователя, обеспечения доступа к ним, обработки запросов;
- он-лайн офисы - сервис создания он-лайн документов: текстовых, презентаций, электронных таблиц и др.;
- сервис автоматизированной разработки веб-сайтов. На сайтах, разработанных для поддержки этого сервиса, размещают системы управления контентом CMS (англ. *Content Management System* - системы управления содержанием, контентом) - программы, которые используются для создания, редактирования и управления содержанием веб-сайта

и др.

Приведенная выше классификация сайтов представлена на рисунке 4.54.

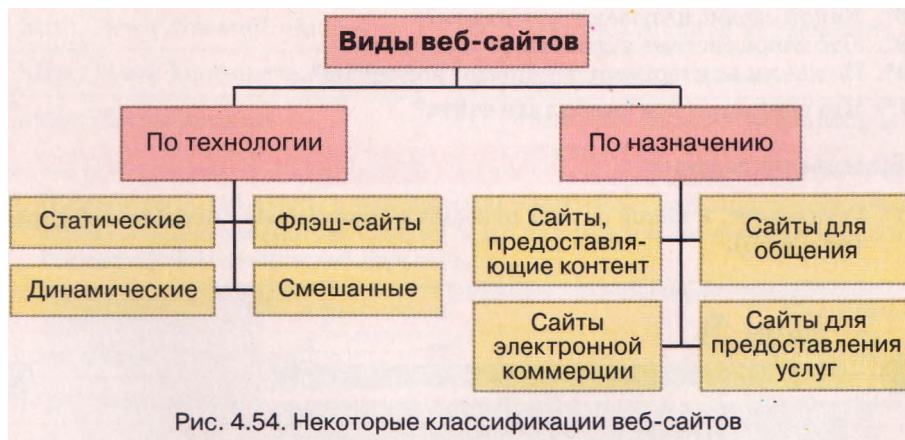


Рис. 4.54. Некоторые классификации веб-сайтов



### Оценивание веб-сайтов

Осуществляя навигацию по ресурсам Интернета, вы, наверно, обращали внимание на то, как по-разному воспринимаются сайты одинаковой тематики и направленности. На одних сайтах вы задерживались надолго, с некоторых сразу переходили на другой ресурс. Длительность просмотра зависит от привлекательности дизайна сайта, удобства поиска нужных материалов. Для посетителя при оценивании веб-сайта важными показателями являются:

- степень информативности веб-сайта, возможность получения полезных материалов, не повторяющих сведения с других ресурсов;
- частота обновления данных на сайте, что свидетельствует о внимании разработчика к собственному проекту;

наличие интерактивных средств, возможность проведения обсуждений, получения ответов, обращения к разработчику за дополнительными сведениями и комментариями;

языки, на которых доступны материалы на веб-сайте;

визуальная привлекательность сайта, дизайн, стиль оформления;

**юзабилити сайта** (англ. *usability* - удобство), то есть понятность, удобство навигации, легкость нахождения нужных данных, структурированность материала, все то, что обеспечивает простоту, комфортность и эффективность в процессе использования сайта

и др.

Длительность пребывания посетителей на сайте, повторное обращение к его ресурсам определяют популярность сайта и его рейтинг в базах данных поисковых систем.

**Проверьте себя**

- 1°. Какие группы элементов можно выделить на веб-странице? Охарактеризуйте каждую группу.
- 2°. Что такое контент веб-страницы?
- 3°. Объясните особенности разных видов веб-страниц в зависимости от их назначения.
- 4°. Чем отличается домашняя страница веб-сайта от других информационных страниц?
- 5°. Объясните особенности разных видов веб-страниц в зависимости от технологий, использованных при их создании.
- 6°. По каким признакам можно классифицировать веб-сайты?
- 7°. Объясните особенности разных видов веб-сайтов по рисунку 4.54.
- 8°. Какой сервис называют хостингом?
- 9°. Что такое система управления контентом?
- 10°. По каким критериями оценивают веб-сайты?
- 11°. Что понимают под юзабилити сайта?

**Выполните задания**

- 1°. Установите, к какой группе относятся отмеченные элементы веб-страницы (рис. 4.55).



Рис. 4.55

<i>Составляющие веб-страницы</i>	<i>Контент</i>	<i>Элементы навигации</i>	<i>Элементы дизайна</i>
Номера элементов			



2. Откройте веб-страницы с указанными URL-адресами и укажите их тип.

<i>URL-адрес страницы</i>	<i>Тип веб-страницы</i>	
	<i>По назначению</i>	<i>По технологии разработки</i>
<a href="http://ukrainancomputing.info/ada_u.html">http://ukrainancomputing.info/ada_u.html</a>		
<a href="http://www.museum-ukraine.org.ua/index.php">http://www.museum-ukraine.org.ua/index.php</a>		
<a href="http://www.litforum.net.ua/showthread.php?t=144&amp;page=70">http://www.litforum.net.ua/showthread.php?t=144&amp;page=70</a>		
<a href="http://mncat.mikanewton.info">http://mncat.mikanewton.info</a>		
<a href="http://children.kmu.gov.ua/history/persons/1358.html">http://children.kmu.gov.ua/history/persons/1358.html</a>		

3. Просмотрите веб-сайты с указанными адресами и установите соответствие между URL-адресами и типами веб-сайтов по их назначению.

	<i>URL-адрес сайта</i>
1	<a href="http://www.profeo.ua">http://www.profeo.ua</a>
2	<a href="http://www.freemail.ukr.net/ua">http://www.freemail.ukr.net/ua</a>
3	<a href="http://www.kancelarka.com.ua">http://www.kancelarka.com.ua</a>
4	<a href="http://formula.co.ua">http://formula.co.ua</a>

	<i>Тип сайта</i>
А	Сайт, предоставляющий информационные материалы
Б	Сайт электронной коммерции
В	Сайт он-лайн сервиса
Г	Сайт для он-лайн контактов и общения



4. Просмотрите веб-сайты с указанными адресами и установите соответствие между URL-адресами веб-сайтов и их типами в соответствии с основной технологией создания веб-страниц.

	<i>URL-адрес сайта</i>
1	<a href="http://veselka-mebli.com">http://veselka-mebli.com</a>
2	<a href="http://www.icfcst.kiev.ua/museum/museum-map_u.html">http://www.icfcst.kiev.ua/museum/museum-map_u.html</a>
3	<a href="http://top.bigmir.net/show">http://top.bigmir.net/show</a>
4	<a href="http://www.brevis.kiev.ua">http://www.brevis.kiev.ua</a>

	<i>Тип сайта</i>
А	Смешанный
Б	Динамический
В	Флэш-сайт
Г	Статический

5. Укажите, каким типам сайтов соответствуют описания:

- сайт содержит 1-5 страниц, включает общие сведения о владельце сайта и его контактных данных;
- сайт содержит каталог товаров и средства для их заказа и оплаты через Интернет;
- сайт содержит большое количество сведений с разной тематикой, сервисов, средств для общения;
- сайт содержит новости автора, его идеи, другие актуальные сведения и может сопровождаться комментариями посетителей.

6\*. Откройте веб-сайты с указанными URL-адресами и укажите их тип.

URL-адрес	Тип сайта	
	По технологии	По назначению
http://www.chl.kiev.ua		
http://www.onlandia.org.ua/ukr		
http://ticapac.pp.ua		
http://www.man.gov.ua		

7\*. Выскажите свое мнение о веб-сайтах с указанными URL-адресами.

Критерии	URL-адрес			
	http:// www.nenc. gov.ua	http:// www.Auth. gov.ua/index. htm	http:// WWW. osvita.com	http:// mamajeva sloboda.ua
Насколько информативным для вас оказался веб-сайт?				
Когда в последний раз были обновлены данные на сайте?				
Имеются ли на сайте интерактивные элементы?				
На каких языках доступны материалы веб-сайта?				
Насколько визуально привлекательным вы считаете сайт?				
Насколько удобным и понятным для вас оказалось использование сайта?				

## 4.8. Этапы создания веб-сайтов. Основы веб-дизайна



1. Какие вы знаете этапы решения задач с использованием компьютера?
2. Что такое дизайн? Каковы основные принципы дизайна презентации и слайдов?
3. Что такое стиль?
4. В чем заключаются законы композиции; колористики; эргономики?
5. Что такое верстка? Каково ее значение при создании публикаций?

### Этапы создания веб-сайтов

Наличие собственного сайта предприятия, организации, учебного заведения и т. п. становится сегодня актуальным вопросом. Сайт может быть полезным в ходе организации учебы, общения, бизнеса и др. Разработку сайта можно заказать в дизайн-студиях или выполнить самостоятельно. В любом случае разработка состоит из нескольких этапов. Эти этапы подобны этапам решения задач с использованием компьютера.



1. **Постановка задания.** На этом этапе определяется цель создания сайта, его основная тематика, выбирается тип сайта, осуществляется анализ существующих сайтов такой же или похожей тематики.

В итоге разработчик должен знать:

- цель, с которой создается сайт;
  - тематику сайта;
  - тип сайта: домашняя страница, форум, Интернет-магазин, портал и т. п.;
  - отличия сайта от других сайтов такой же тематики;
  - аудиторию потенциальных посетителей сайта: возраст посетителей, пол, круг интересов и др.;
- перечень сервисов для размещения на сайте: форум, чат, поисковая система, веб-каталог, электронная почта и др.;
- перспективы развития сайта.

2. **Определение структуры сайта и его отдельных страниц.** На этом этапе важно составить перечень разделов сайта для формирования системы навигации, список страниц, определить связи между ними. Количество страниц будет зависеть от того информационного наполнения, которое планируется на нем разместить. Результатом должна стать карта сайта - диаграмма, визуально отображающая иерархию страниц сайта, схему связей и переходов между ними, то есть внутреннюю структуру сайта.

Например, для сайта-визитки вашего класса, основными посетителями которого станут ученики класса и их родители, карта сайта может быть такой, как предложено на рисунке 4.56.

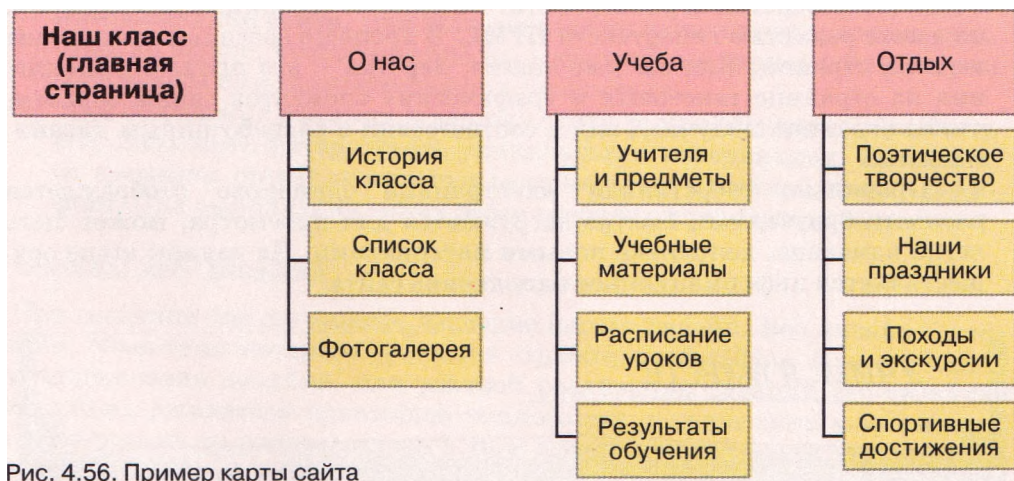


Рис. 4.56. Пример карты сайта

Вторым заданием этапа является разработка так называемой внешней структуры сайта, которая определяет внешний вид веб-страниц. Поскольку для большинства страниц сайта рекомендуется применять единый стиль оформления, то нужно определить схему расположения на страницах основных блоков: как будет расположен основной материал, дополнительные информационные и рекламные блоки, анонсы, меню, счетчик посетителей и т. п.

Как правило, на веб-страницах предполагается размещение:

верхнего блока - заголовка, в котором содержатся логотип и название сайта;

- меню для перехода к основным разделам сайта;
- информационного блока с основным материалом, который занимает центральную часть страницы;
- нижнего блока - подвала, для размещения контактных данных, сообщения об авторских правах и т. п.

Схему внешней структуры называют модульной сеткой (рис. 4.57). Если разработка страниц сайта будет осуществляться автоматизированными средствами, то внешняя структура может быть предложена в шаблоне страницы.

3. Разработка дизайн-макета страниц сайта. Дизайн-макет будет опираться

на предварительно разработанную внешнюю структуру страниц сайта. Дизайн-макет страниц включает набор значений свойств текстовых и графических объектов страницы: цветовой гаммы страниц, элементов графического оформления, набора шрифтов и др., то есть определяет стиль сайта (рис. 4.58). Важно, чтобы стиль соответствовал назначению сайта, особенностям основной аудитории, на которую рассчитан сайт, был ориентирован на предоставление наибольших удобств для восприятия основного материала.

4. Создание и верстка страниц сайта. Создаются страницы, как правило, на языке разметки гипертекста HTML. В процессе создания происходит верстка страниц. Как вы уже знаете, верстка - это процесс размещения на странице текстовых и графических элементов таким образом, чтобы страница получила вид в соответствии с разработанным дизайн-макетом.

Правильно сверстанная веб-страница одинаково отображается разными браузерами, быстро загружается для просмотра, может быть легко изменена, дополнена новыми материалами. На данном этапе осуществляется информационное наполнение сайта.

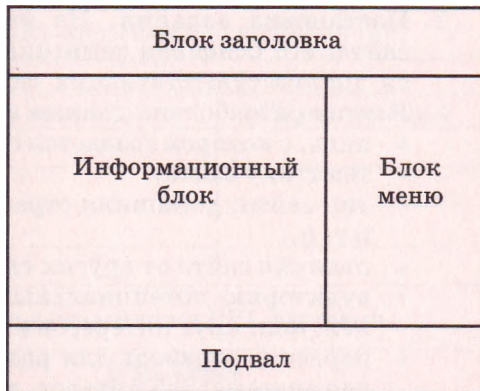


Рис. 4.57. Пример схемы внешней структуры

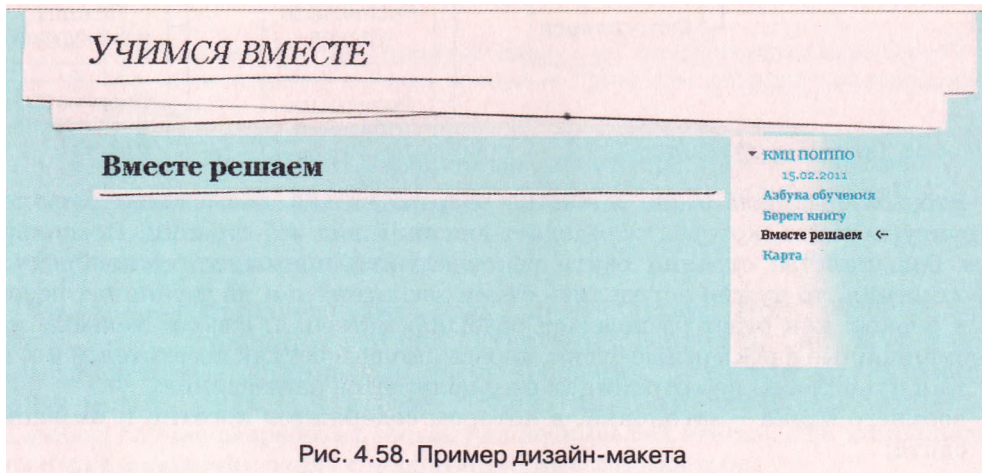


Рис. 4.58. Пример дизайн-макета

5. Если на этапе постановки задания предусматривалось размещение на сайте интерактивных элементов, таких как системы поиска, голосования, форум и др., то нужен еще и этап программирования сайта.
6. Размещение (публикация) сайта в Интернете. В ходе предыдущих этапов созданные веб-страницы могли сохраняться на локальном компьютере разработчика. На этом этапе сайт размещают на сервере, который предоставляет услуги хостинга. Во время публикации в Интернете сайту предоставляется доменное имя. После этого сайт становится доступным для просмотра всеми желающими, если он или его часть не имеют ограничений на доступ.  
Организации, предоставляющие услуги хостинга, называют хостинг-провайдерами. Существуют серверы, которые предоставляют бесплатный хостинг. При этом, как правило, на вашем сайте будет размещаться посторонняя реклама и ограничиваться использование некоторых интерактивных средств. Можно разместить сайт на сервере платного хостинга без рекламы и со всеми нужными программными средствами.
7. После публикации сайта в Интернете его разработка не считается завершенной. Определенное время будет длиться тестирование сайта для выявления недостатков верстки.
8. Популяризация и поддержка сайта. Для того чтобы ваш сайт начали посещать пользователи Интернета, желательно зарегистрировать его в поисковых системах и каталогах, разместить ссылку на него на других сайтах. Этот процесс называют популяризацией, раскруткой или продвижением сайта. Для популяризации используют и другие средства, но важно, чтобы материалы, доступные на вашем сайте, стоили того, чтобы ими заинтересовались посетители.
9. Для поддержки интереса к вашему сайту важно регулярно обновлять его, дополнять интересными уникальными материалами. Возможно со временем станет желательным изменение дизайна сайта - редизайн. При выполнении таких работ говорят о сопровождении сайта.

## Основы веб-дизайна

При создании сайтов важное значение играет дизайн. Веб-дизайн - это дизайн, объектами которого являются страницы веб-сайтов. Как и в каждом направлении дизайна, для удачной реализации заданий веб-дизайна нужно придерживаться принципов композиции, колористики и эргономики, с которыми вы ознакомились в 10-м классе при изучении темы «Компьютерные презентации».

Как уже отмечалось, важное значение в веб-дизайне имеет стиль. Дизайн сайта может быть спроектирован в одном из приведенных стилей (рис. 4.59).

Выбирая стиль дизайна по цветовой гамме, стоит учитывать особенности восприятия цвета посетителями разного возраста и пола:

- темный стиль предполагает, что в оформлении сайта преобладают темные цвета. Такие сайты хорошо воспринимает молодежь, но они кажутся слишком мрачными, тревожными лицам старшего возраста;
- светлый стиль с преобладающими светлыми цветами легко воспринимается всеми категориями посетителей, но кое-кто считает светлые сайты слишком простыми;

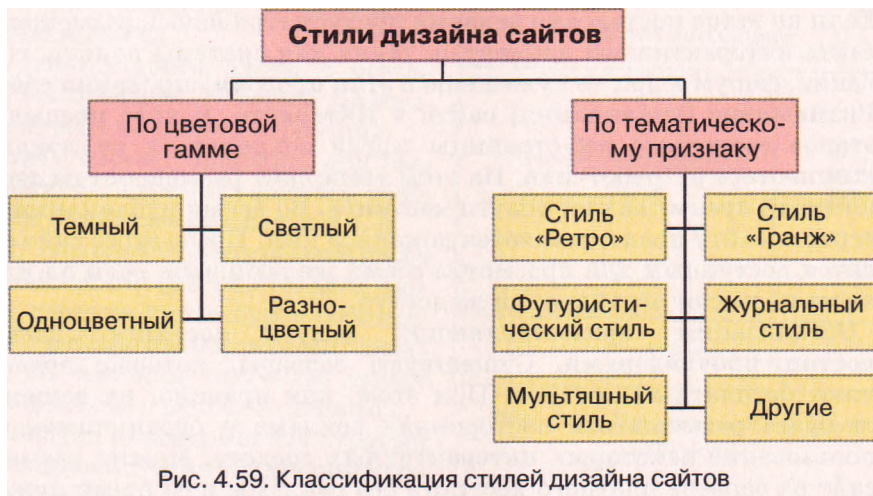


Рис. 4.59. Классификация стилей дизайна сайтов

- одноцветный стиль — это стиль оформления, в котором преобладает один цвет, отличающийся от черного и белого. Часто выбранный цвет является элементом имиджа владельца;
- разноцветный стиль предполагает использование двух или более цветов, которые занимают наибольшую площадь на сайте. Разноцветный стиль часто выбирают при создании сайтов для детей.

Наибольшей является классификация стилей дизайна по тематическому признаку. Приведем несколько примеров стилей дизайна этой классификации:

- стиль «Ретро» - в оформлении страниц используют элементы декора, предметы интерьера и другие атрибуты, присущие прошлым десятилетиям;
- стиль «Гранж» (амер. разговор, *grunge* - нечто неприятное) - асимметричное размещение элементов на странице создает впечатление хаотической композиции, выполненной якобы небрежно (рис. 4.60);
- футуристический стиль - в оформлении страницы используют графические элементы, которые являются атрибутами предполагаемого будущего: роботы, высокотехнологичные механизмы и т. п.;
- журнальный стиль - веб-страницы оформлены по принципам дизайна гляцевых журналов: большие заголовки, крупные высококачественные фотографии и др.;
- мультяшный стиль - оформление страниц стилизовано под мультфильмы и комиксы.

По тематическому признаку стилей дизайна намного больше, поскольку разнообразной является тематика информационных материалов и каждая тема может найти свое отражение в дизайне.

Особое значение при выборе дизайна играет назначение веб-сайта, а не только его эстетическое восприятие. Любой посетитель сайта благодаря дизайну должен быстро понять, чему посвящен сайт, легко найти полезные для себя информационные материалы, интуитивно сориентироваться в навигации по страницам сайта. Анализ самых популярных веб-сайтов доказывает, что чем проще дизайн, тем легче пользователю ориентироваться на сайте. Большое количество графических элементов может отвлекать внимание от действительно полезных материалов, ради которых посетитель попал на сайт.



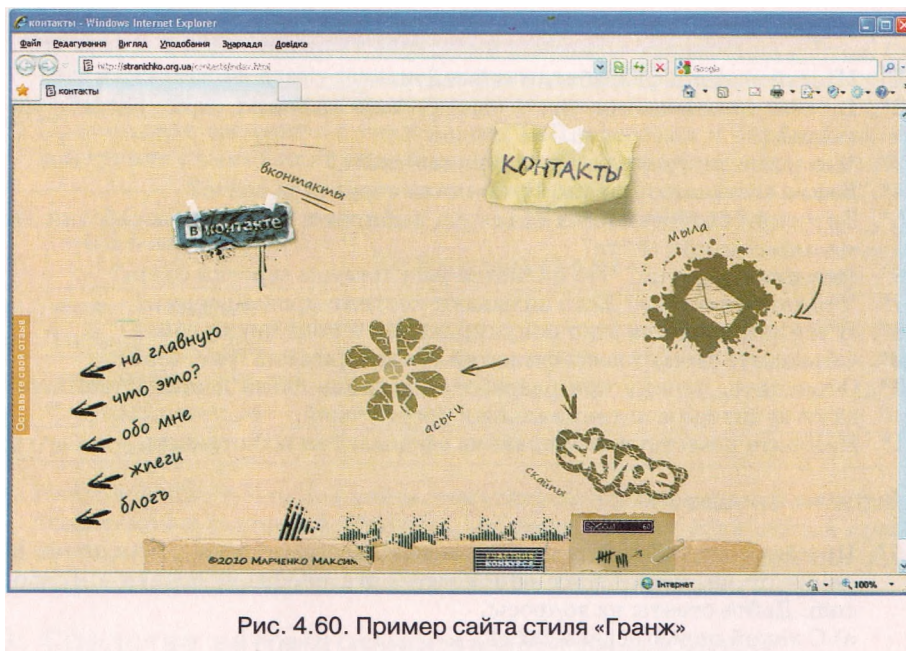


Рис. 4.60. Пример сайта стиля «Гранж»

Учитывая эти и другие факторы, можно предложить такие правила оформления веб-страниц:

- Выберите цветовую гамму для вашего сайта, учитывая ваши предпочтения, тематику сайта и предпочтения потенциальных посетителей. Используйте для оформления не более 2-3 цветов.
- При определении цветовой гаммы страницы выбирайте контрастные цвета для текста и фона, чтобы текст легче читался.
- Разбивайте текст на абзацы, между которыми сделайте увеличенные отступы.
- Выберите размер шрифта, при котором текст будет восприниматься комфортно - не слишком мелкий и в меру крупный. Шрифт в заголовках сделайте крупнее шрифта основного текста. Вид шрифта сделайте одинаковым на всех страницах.
- Выравнивайте заголовки по центру, а основной текст — по ширине. Для структурирования текста используйте таблицы. Размещая фрагменты текста и графические изображения в таблицах, можно создать интересные композиционные решения на веб-страницах.
- Не злоупотребляйте флэш-анимацией, видео-, музыкальными и графическими элементами. Они могут отвлекать внимание от полезных материалов и замедлять загрузку страниц.
- Сделайте гиперссылки для перехода между страницами сайта, но не размещайте на страницах слишком много гиперссылок. Выделяйте гиперссылки цветом, чтобы пользователь видел, что это ссылки и какие из них он уже посетил.
- Сделайте логотип сайта, изображение или текст в заголовке гиперссылками на главную страницу сайта.
- Создайте для посетителей карту сайта для упрощения перехода на страницы с нужными материалами.

 Проверьте себя

- 1°. Назовите этапы разработки веб-сайта.
- 2°. В чем заключается этап постановки задания при разработке веб-сайта?
- 3°. Что такое внутренняя структура веб-сайта?
- 4°. Какие составляющие имеет внешняя структура сайта?
- 5°. Значения свойств каких объектов выбирают на этапе разработки дизайн-макета страниц сайта?
- 6°. Что такое верстка? Что является результатом верстки сайта?
- 7°. Что такое хостинг? Кого называют хостинг-провайдерами?
- 8°. В чем заключается этап популяризации и поддержки сайта?
- 9°. Объясните схему классификаций стилей дизайна (рис. 4.59).
- 10\*. Объясните, почему при разработке дизайна сайта важно учитывать основную аудиторию потенциальных посетителей.
- 11°. Назовите известные вам правила оформления веб-страниц.

 Выполните задания



1. Просмотрите веб-сайты с указанными адресами: <http://pets.uz.ua>, <http://palace.dp.ua>, <http://www.odnoklasnyk.org.ua>, <http://smakuje.livejournal.com>. Дайте ответы на вопросы.
  - а) С какой целью создан сайт?
  - б) Какова тематика сайта?
  - в) Каков тип сайта?
  - г) Что есть оригинального в сайте?
  - д) Кто потенциальные посетители сайта?
  - е) Какие сервисы предложены на сайте?
  - ж) Какие идеи просмотренного сайта вы могли бы использовать при разработке собственного сайта?
- 2°. Просмотрите веб-сайты с указанными адресами: <http://www.kazka.in.ua>, <http://www.npblog.com.ua>, <http://maque.org.ua>, <http://neyrat.net>. Откройте карты сайтов и сравните формы их представления. Определите, на каждом ли сайте есть карта сайта, предоставляет ли она дополнительные удобства посетителям сайта.
3. Просмотрите веб-сайты с указанными адресами: <http://platfor.ma>, <http://starodub.org.ua>, <http://molode.com.ua>, <http://www.4uth.gov.ua>, <http://osobystist.com>. Сравните схему внешней структуры страниц сайтов. Определите, на всех ли сайтах использованы блоки заголовка, информационный, подвал, как размещен блок меню.
- 4\*. Выберите одну из предложенных тем для веб-сайта: *Опасные животные, Космические исследования, Коллекция рекордов, Научные открытия, Художественный салон*.
  - а) Выберите тип для сайта этой тематики.
  - б) Создайте в текстовом редакторе схему внутренней структуры сайта. Сохраните ее в собственной папке в файле с именем задание 4.8.4.docx.
  - в) Разработайте в графическом редакторе схему внешней структуры страниц сайта. Сохраните ее в собственной папке в файле с именем задание 4.8.4.bmp.
- 5°. Просмотрите веб-сайт дизайн-студии ArtPoint (<http://artpoint.com.ua>). Проанализируйте образцы работ, разработки элементов корпоративного стиля. Обратите внимание на соблюдение тематики и цветовой гаммы. Определите, использование каких элементов могло бы подчеркнуть единый стиль страниц вашего сайта.



- 6°. Просмотрите веб-страницу сайта **Файнохост** (<http://www.faynohost.com.ua/design-templates.html>) с образцами шаблонов оформления сайтов. Обратите внимание на схему внешней структуры шаблонов страниц. Определите, какую схему вы могли бы использовать для собственного сайта.
- 7°. Просмотрите веб-сайты с указанными URL-адресами и определите, к какой группе классификации принадлежит стиль дизайна каждого из них.

URL-адрес	Стиль дизайна сайта	
	По цветовой гамме	По тематическому признаку
<a href="http://www.nbg.kiev.ua">http://www.nbg.kiev.ua</a>		
<a href="http://1000plastinok.net">http://1000plastinok.net</a>		
<a href="http://www.vovazilvova.net">http://www.vovazilvova.net</a>		
<a href="http://fcdynamo.kiev.ua">http://fcdynamo.kiev.ua</a>		
<a href="http://www.janda.ru">http://www.janda.ru</a>		



- 8\*. Найдите в Интернете 3-4 сайта, тематика которых совпадает с выбранной вами темой в задании 4. Оцените, соблюдены ли на этих сайтах правила оформления веб-страниц. Запишите, какие идеи просмотренных сайтов вы могли бы использовать при разработке вашего сайта.

## 4.9. Средства автоматизированной разработки веб-сайтов



1. Что такое хостинг? Какие организации называют хостинг-провайдерами?
2. По каким признакам можно классифицировать веб-страницы? Какие типы веб-страниц вам известны?
3. Из каких блоков состоит внешняя структура веб-страницы?
4. Какие вам известны форматы графических, аудио- и видеофайлов? Какие из этих форматов предусматривают сжатие данных?
5. Какие средства используют для изменения формата файлов?

### Средства разработки веб-сайтов

С 9-го класса вам известно, что веб-страницы являются текстовыми файлами, расширение их имен соответствует языку разметки гипертекста, который использован при разработке страницы. Как вы уже знаете, веб-страницы часто создают с использованием языка разметки гипертекста HTML. У файлов, созданных на языке HTML, расширение имен файлов htm или html. В файлах такого формата содержится HTML-код страницы, состоящий из данных двух типов:

текста, который будет отображаться на странице;

команд, которые определяют разметку текста - его структуру, формат фрагментов текста, обеспечивают вставку нетекстовых объектов на страницу и др. Такие команды в языке HTML называют тегами (англ. tag - ярлык, признак).

Просмотреть HTML-код веб-страницы, открытой в окне браузера Internet Explorer, можно, выполнив Вид => Просмотр HTML-кода. В других браузерах эта команда может быть Вид => Начальный код страницы (Mozilla Firefox), Меню => Страница => Инструменты разработчика => Источник (Opera), Инструменты => См. источник (Google Chrome) и др.

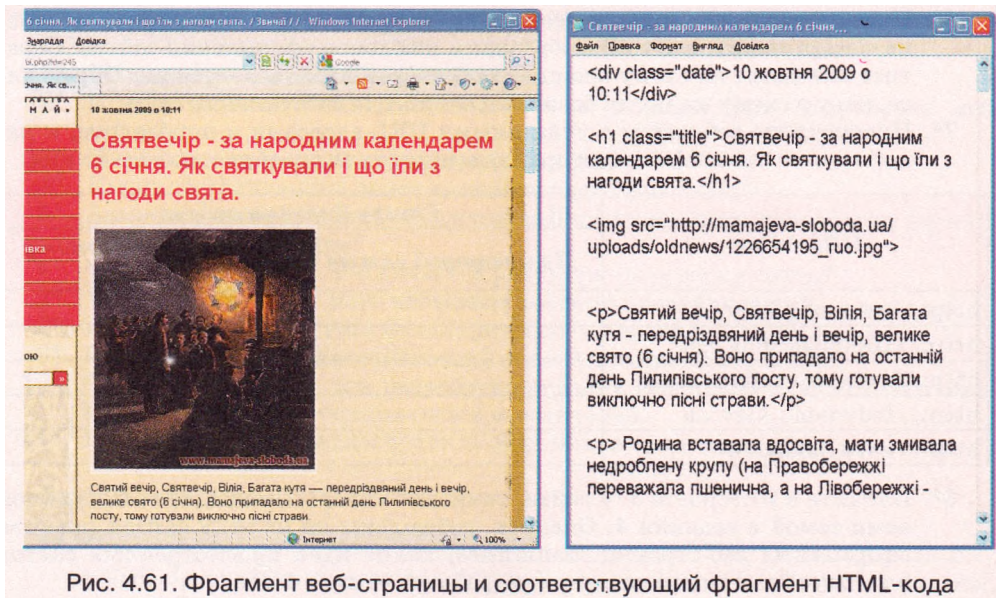


Рис. 4.61. Фрагмент веб-страницы и соответствующий фрагмент HTML-кода

На рисунке 4.61 приведены изображение части веб-страницы сайта и соответствующий фрагмент HTML-кода.

Разработка веб-страниц может выполняться с использованием разных средств:

- **текстовых редакторов**, в среде которых пользователь может вводить текст и теги. Для этого можно использовать текстовые редакторы Блокнот, Edit Plus, Homesite, HTML Pad и др. Готовую страницу нужно сохранить с расширением имени файла `htm` или `html`. Создание веб-страницы такими средствами предусматривает обязательное знание языка разметки гипертекстов, например HTML. Вся работа по отбору тегов разработчик выполняет вручную. Разработка страницы происходит медленно, но благодаря небольшому размеру файла, в котором она сохраняется, такая страница быстро загружается и открывается в окне браузера;
- **прикладных программ общего назначения**, например из пакета Microsoft Office, которые могут сохранять файлы в формате HTML. Как вы уже знаете, документы, созданные в программах Word 2007, PowerPoint 2007, Publisher 2007 и др., можно сохранить, выбрав тип файла *веб-страница* (`*.htm`, `*.html`). При этом создаются веб-страницы, в которых применение тегов для разметки происходит автоматически. Разработчику не нужно знать язык разметки гипертекста. Но файлы, в которых сохраняются такие страницы, имеют намного больший объем, чем файлы, созданные в текстовых редакторах. Причиной является большое количество тегов, которые описывают структуру и форматирование документа и дублируются при применении к каждому отдельному элементу страницы;
- **специализированных веб-редакторов** - программ, предназначенных для разработки веб-сайтов. Популярными веб-редакторами являются Adobe Dreamweaver, Microsoft FrontPage, SharePoint Designer, WYSIWYG Web Builder, KompoZer и др. Эти программы имеют дополнительные средства для создания статических и динамических веб-страниц, при этом не требуют от разработчика знания языка HTML.



Такие программы называют WYSIWYG-редакторами (англ. *What You See Is What You Get* - что вы видите, то вы и получаете), созданная их средствами веб-страница будет выглядеть так, как она сконструирована в редакторе. HTML-код страницы более корректен, чем при использовании, например, программ пакета Microsoft Office, но тоже избыточный; систем управления веб-контентом WCMS (англ. *Web Content Management System* - система управления веб-контентом), которые предоставляют пользователям удобные инструменты для управления текстовым и графическим наполнением веб-сайтов, добавления и удаления статей с информационными материалами, создания системы навигации веб-сайтов и др. Популярными WCMS являются системы Joomla, Wordpress, Drupal, MediaWiki, Mambo, NUKE и др. Системы управления содержимым веб-сайтов предлагают наборы шаблонов оформления веб-страниц и модулей, которые делают сайт динамическим: форумов, чатов, лент новостей, каталогов файлов, контроля статистики и т. п.

Названные средства создания веб-сайтов можно установить на локальном компьютере. Созданные в их среде веб-страницы после завершения процесса разработки нужно опубликовать в Интернете. В то же время существуют WCMS, которые одновременно с услугами по разработке веб-сайтов предоставляют услуги бесплатного хостинга. Такими являются системы uCoz, Google Sites, Prom.ua, Ua7.biz и др. В этих системах создание веб-сайта осуществляется в режиме он-лайн сразу на сервере хостинга.

### Автоматизированное создание веб-сайта

Создание веб-сайта средствами он-лайн системы управления веб-контентом происходит за несколько шагов:

1. Регистрация учетной записи на сервере.
2. Выбор имени сайта и шаблона для его оформления.
3. Создание страниц сайта, системы навигации.
4. Заполнение страниц контентом.

Рассмотрим, как происходит процесс разработки веб-сайта средствами, бесплатно предоставляемыми пользователям веб-сервером Google. Соответствующий сервис имеет название Сайты Google.

Прежде чем разрабатывать сайт, нужно создать учетную запись - аккаунт Google. С этой целью:

1. Откройте в окне браузера главную страницу сайта Google (<http://www.google.ru>).
2. Выберите гиперссылку *Войти => Создайте аккаунт прямо сейчас*.
3. Заполните поля формы Регистрация аккаунта на странице Аккаунты Google. Выберите кнопку Я принимаю условия. Создайте мой аккаунт.
4. Активируйте ваш аккаунт, используя гиперссылку в тексте письма, автоматически отправленного на электронный почтовый ящик, на который вы зарегистрировали ваш аккаунт.
5. Выберите гиперссылку Войти на веб-странице Аккаунты Google в окне браузера, открывающемся после выбора гиперссылки из электронного письма.
6. Заполните поля Электронная почта и Пароль данными, которые вы вводили во время создания учетной записи.
7. Выберите кнопку Войти.
8. Выберите гиперссылку Домашняя страница Google в нижней части веб-страницы.

В итоге откроется главная страница сервера Google, на которой вы будете иметь права пользователя сервисов Google. Адрес, на который зарегистрирована ваша учетная запись, будет отображаться в верхней части веб-страницы.

Для создания сайта нужно:

1. Открыть в окне браузера главную страницу сайта Google (<http://www.google.ru>).
2. Выбрать в верхней части страницы гиперссылку *Еще => Сайты*.
3. Заполнить на странице Сайты Google — бесплатные веб-сайты и вики поля Электронная почта и Пароль данными вашей учетной записи, если они не заполнены. Выбрать кнопку Войти.
4. Выбрать на странице Сайты Google кнопку Создать сайт.
5. Выбрать один из предложенных шаблонов для создания сайта. По умолчанию предлагается шаблон *Пустой шаблон*.
6. Заполнить поле Дайте название своему сайту. Например, если вы создаете сайт вашего класса, то название может быть *Классная жизнь*. Одновременно автоматически будет предложен URL-адрес главной страницы сайта. Он будет иметь вид: [https://sites.google.com/site/<название\\_сайта>](https://sites.google.com/site/<название_сайта>). Часть *название сайта* не должна содержать букв кириллицы, а поэтому система предлагает запись русских слов буквами английского алфавита без пробелов. Для указанного названия сайта системой будет предложен адрес <https://sites.google.com/site/klassnaazizn>, но при желании его можно изменить. URL-адрес главной страницы сайта должен быть уникальным.
7. Открыть список Выбрать тему и выбрать тему оформления сайта. По умолчанию предлагается тема *Запуск по умолчанию*. Например, выберем тему *Ланта*.
8. Ввести символы в поле Введите показанный код.
9. Выбрать кнопку Создать сайт.

В окне браузера откроется домашняя страница вашего сайта. В верхней части страницы будет отображаться указанное вами название сайта, слева - панель навигации, в центральной части - область для информационного блока страницы с заголовком Главная страница (рис. 4.62).

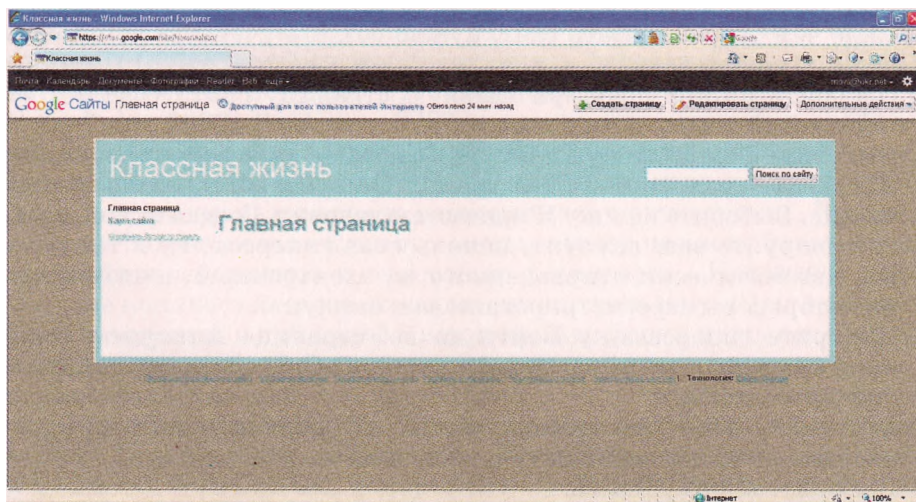
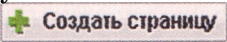


Рис. 4.62. Вид главной страницы только что созданного сайта

Панель навигации содержит две гиперссылки: Главная страница и Карта сайта. Карта сайта на этот момент содержит ссылку только на домашнюю страницу.

## Создание и настройка веб-страниц

Сразу после создания сайт содержит только одну домашнюю страницу. Для создания новой страницы на сайте нужно:

1. Выбрать кнопку Создать страницу  в верхней части окна браузера.
2. Выбрать шаблон содержимого будущей страницы (рис. 4.63):
  - Веб-страница - страница для размещения текста, изображений и т. п. Такие страницы являются информационными страницами тематических разделов.
  - Объявления - страница для размещения текстовых сообщений, упорядоченных в хронологическом порядке, начиная с последних введенных. Каждое сообщение может отображаться как отдельная информационная страница.
  - Файловый менеджер - страница для хранения гиперссылок на загруженные файлы. Гиперссылки сгруппированы в соответствии с именами папок, в которых размещены файлы. Такие страницы являются страницами-контейнерами, содержащими каталоги файлов.
  - Список - страница, на которой представлены структурированные данные как списки из нескольких полей. Списки можно сортировать по данным в каждом поле.

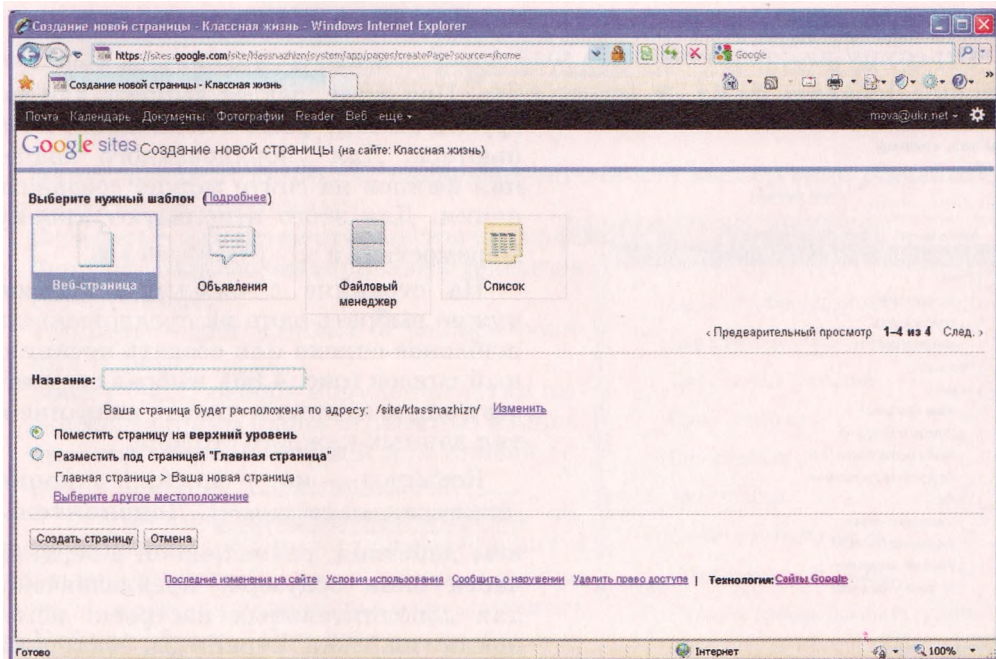


Рис. 4.63. Страница **Создание новой страницы**

Например, для создания страницы *История класса* сайта *Классная жизнь* (рис. 4.56) можно выбрать шаблон *Веб-страница*, для страницы *Учителя и предметы* - шаблон *Список*, *Поэтическое творчество* - шаблон *Объявления*, *Учебные материалы*, - шаблон *Файловый менеджер*.

3. Ввести название страницы в поле *Название*.

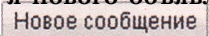
4. Выбрать размещение страницы в структуре сайта. Можно выбрать варианты *Поместить страницу на верхний уровень*, *Разместить под страницей <имя страницы>* или *Выберите другое местоположение*. В первом случае гиперссылка на страницу будет размещена в главном меню сайта, во втором - она появится на выбранной странице. При выборе гиперссылки *Выберите другое местоположение* открывается панель *Выбор страницы* (рис. 4.64) для выбора веб-страницы, с которой будет связана новая страница.

Например, страницы *О нас*, *Учеба* и *Отдых* можно разместить на верхнем уровне, страницы *История класса*, *Список класса* и *Фотогалерея* связать со страницей *О нас*; *Учителя и предметы*, *Расписание уроков*, *Результаты обучения*, *Учебные материалы* - со страницей *Учеба* и т. п.

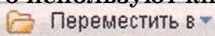
5. Выбрать кнопку *Создать страницу*.

После создания страницы она открывается в режиме редактирования, а панель навигации и карта сайта автоматически дополняются ссылками на новую страницу.

При создании страницы на основе шаблона *Веб-страница* ее нужно заполнить материалами и сохранить. Это статическая страница, на которой не предусматривается частое изменение наполнения.



Страница на основе шаблона *Объявления* будет обновляться регулярно с появлением новостей. Для добавления нового объявления на странице следует выбрать кнопку *Новое сообщение* , ввести текст сообщения и сохранить страницу.

Страницу на основе шаблона *Файловый менеджер* заполняют, выбирая кнопку *Добавить файл* .

При этом следует выбрать для загрузки на сайт файл с локального компьютера. Для упорядоченного хранения файлов на сайте можно создавать папки. Для этого используют кнопку *Переместить в* .

На странице с шаблоном *Список* нужно выбрать один из предложенных шаблонов списка или создать нетипичный список (рис. 4.65), выбрав соответствующую кнопку и указав название и тип данных каждого столбца.

Команды из списка кнопки

 *Дополнительные действия*  *Дополнительные действия*, размещенной в верхней части окна браузера, предназначены для дополнительных настроек: изменения шаблона страницы, удаления страниц, управления доступом к материалам сайта и др.

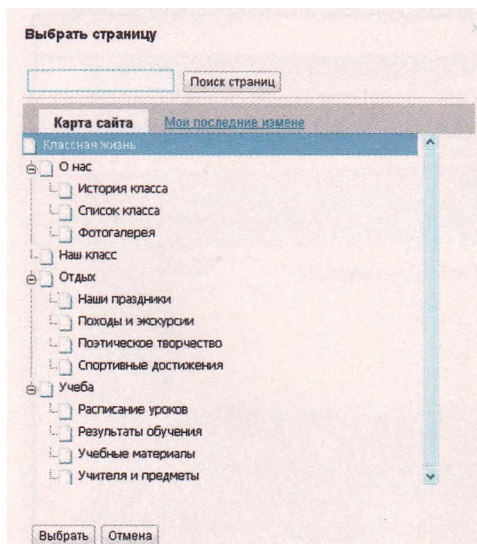


Рис. 4.64. Панель **Выбор страницы**



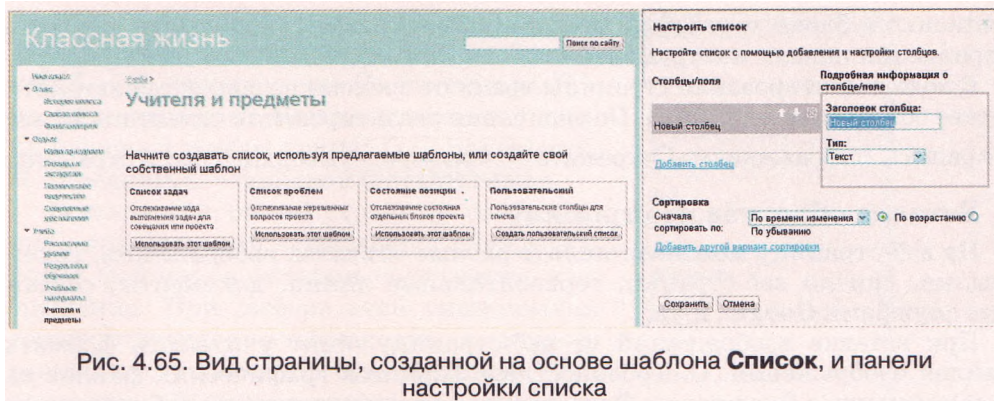


Рис. 4.65. Вид страницы, созданной на основе шаблона **Список**, и панели настройки списка

Панель навигации и карта веб-сайта формируются автоматически во время создания новых страниц. Названия страниц на панели навигации размещаются в алфавитном порядке. Размещение страниц можно изменять, выполнив *Карта сайта => Управление страницами* и перетянув в схеме сайта название одной страницы на название другой, с которой нужно связать страницу.

### Редактирование веб-страниц

Созданную страницу веб-сайта можно редактировать, наполнять ее информационными материалами, изменять модульную сетку и т.п. Для перехода в режим редактирования страниц нужно выбрать кнопку **Редактировать страницу** в верхней части окна браузера. После этого в окне браузера появляется меню и панель инструментов (рис. 4.66).

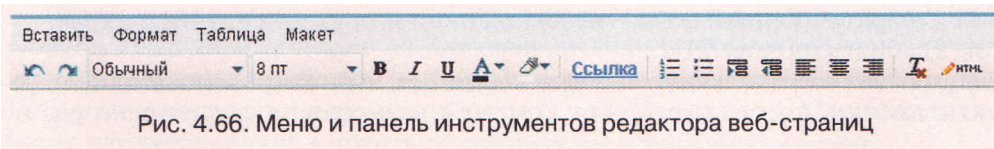


Рис. 4.66. Меню и панель инструментов редактора веб-страниц

Меню содержит команды, предназначенные для выполнения операций с элементами веб-страницы:

- **Вставить** - для вставки объектов на страницу;
- **Формат** - для форматирования текста на веб-странице;
- **Таблица** - для вставки и редактирования таблиц;
- **Макет** - для выбора модульной сетки информационного блока страницы: в один, два, три столбца, с левой или правой боковой панелью и др. (рис. 4.67).

В режиме редактирования рабочая область страницы содержит поля заголовка и информационного блока, в которые можно вводить текст с клавиатуры или вставлять из Буфера обмена. При изменении заголовка страницы его текст одновременно отображается на панели навигации. Текст в инфор-

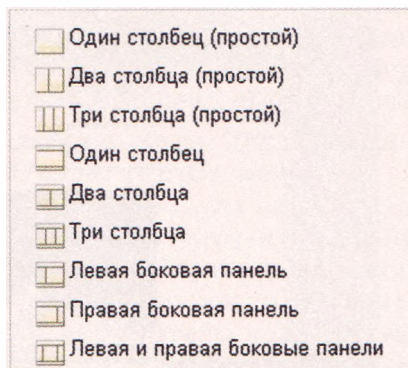
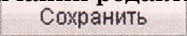


Рис. 4.67. Варианты компоновки веб-страниц

мационном блоке страницы можно форматировать, используя элементы управления панели инструментов.

В ходе редактирования страницы время от времени выполняется автоматическое сохранение черновика. По окончании редактирования изменения нужно сохранить, выбрав кнопку Сохранить  в верхней части страницы.

## Вставка объектов на страницу

На веб-страницу можно вставлять разные объекты: изображения, гиперссылки, списки веб-страниц, горизонтальные линии, документы, созданные службами Google, и др.

При вставке изображений на веб-страницу стоит учитывать форматы файлов изображений. С особенностями форматов графических файлов вы ознакомились в 9-м классе. Фотографии для размещения на веб-страницах чаще всего сохраняют в файлах формата JPG. Анимированные изображения, как правило, содержатся в файлах формата GIF. Их часто размещают на веб-сайтах для придания эмоциональной окраски странице. Специально для размещения растровых изображений в сети был разработан формат PNG. Если изображение, нужное вам для размещения на веб-странице, хранится в файле другого формата, его стоит конвертировать в один из названных, используя средства графического редактора.

Для вставки изображения на веб-страницу нужно:

1. Выполнить *Вставить => Изображение*.
2. В окне Добавление изображения выбрать источник изображения:
  - Загруженные изображения - вставить изображение из файла, уже размещенного на сайте или хранящегося на локальном компьютере;
  - Интернет-адрес (URL) - вставить изображение, хранящееся по указанному URL-адресу в Интернете.
3. Выбрать нужный файл с изображением или ввести его URL-адрес.
4. Выбрать кнопку ОК.

При выборе вставленного изображения под ним открывается панель редактирования, которая содержит команды размещения и установки разме-

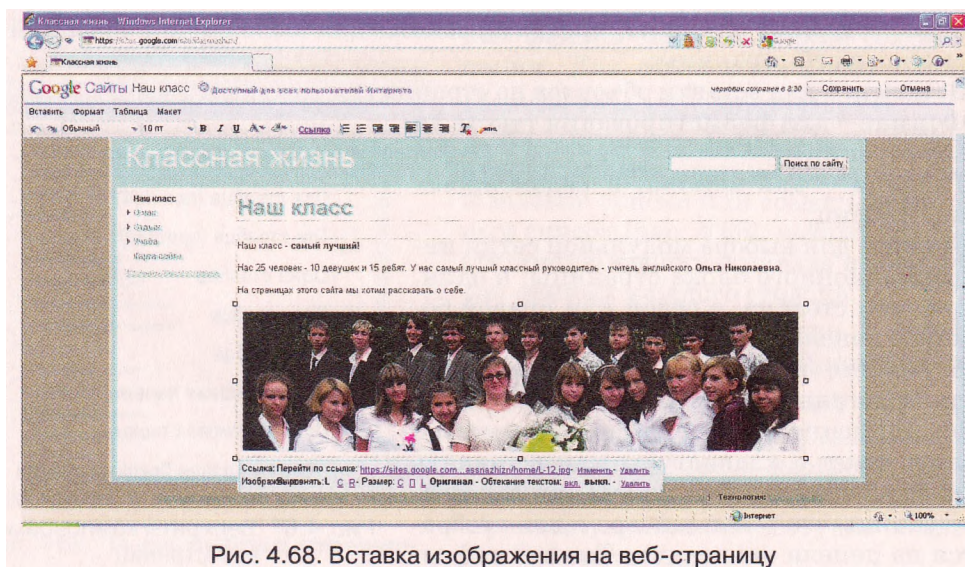


Рис. 4.68. Вставка изображения на веб-страницу

ра (рис. 4.68). Используя гиперссылки на этой панели, можно разместить изображение: **L** - по левому краю, **C** - по центру, **R** — по правому краю. Можно установить один из размеров изображения: **C** - маленький, **M** - средний, **L** - большой или **Оригинал**.

Вставленное на веб-страницу изображение автоматически связывается гиперссылкой с файлом, в котором содержится изображение. При выборе этой гиперссылки соответствующее изображение в полном размере открывается в окне браузера. Для изменения объекта для перехода следует использовать гиперссылку **Изменить** на панели редактирования в строке **Перейти по ссылке**.

Используя меню **Вставить** (рис. 4.69), на веб-страницу можно вставить объекты, созданные с использованием сервисов **Google** (карты, календари, документы, презентации, электронные таблицы, формы для опроса) и фотосервиса **Picasa** (фотографии и слайд-шоу). Все эти объекты вставляются на страницу с использованием **гаджетов** (англ. *gadget* - средство, приспособление) - небольших программ, которые размещаются на веб-страницах и предназначены для воспроизведения некоторых специфических данных.

Видео можно вставлять на веб-страницу с одного из веб-ресурсов: **Видео Google**, **YouTube** или **Видео из Документов Google**. Для этого нужно выполнить **Вставить => Видео**, выбрать источник видеоматериалов, вставить URL-адрес видеофрагмента и выбрать кнопку **Сохранить**. На веб-страницу будет встроена панель гаджета со средствами отображения видео.

Гиперссылку на веб-страницу с этого сайта или на другой ресурс можно вставить, выполнив **Вставить => Ссылка** и выбрав в открывшемся окне **Создать ссылку** объект, на который будет осуществляться переход при выборе гиперссылки. При этом созданная ссылка будет связана с названием страницы или URL-адресом ресурса. Гиперссылки также можно связывать с любыми текстовыми фрагментами или изображениями, которые содержатся на веб-странице. Для этого нужно выделить фрагмент, выбрать кнопку **Ссылка** **Ссылка** на панели инструментов и выбрать объект для перехода - существующую страницу этого сайта или URL-адрес другого ресурса.

Файлы разных форматов (флэш-анимация, звук и др.), для которых может быть недоступной вставка на веб-страницы, можно загрузить на сайт. Загруженные файлы сохраняются на сервере, а на веб-странице в разделе Приложения размещаются гиперссылки, выбрав которые можно сохранить файл на локальном компьютере, удалить или просмотреть, если формат файла совпадает с форматом документов **Google**. Для загрузки файла на сервер нужно выбрать в нижней части страницы гиперссылку **Приложение**, кнопку **Обзор** и выбрать в окне **Открытие файла** нужный файл на ло-

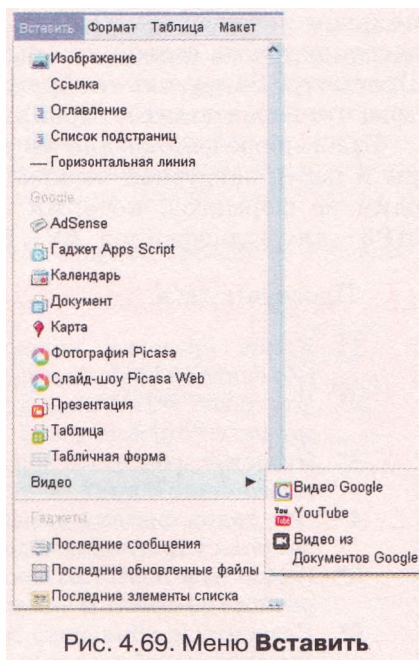


Рис. 4.69. Меню **Вставить**



кальном компьютере. После выбора кнопки **Открыть** файл автоматически загружается на сервер, его имя и соответствующие гиперссылки **Удалить**, **Просмотр**, **Загрузить** отображаются в нижней части веб-страницы. Заметим, что на домашнюю страницу загрузка файлов не разрешена.

При выборе файлов для загрузки нужно обращать внимание на их размеры и перед загрузкой осуществлять конвертацию аудио- и видеофайлов в один из форматов, который предусматривает сжатие данных, например **MP3** - для аудиофайлов, **AVI**, **MP4** - для видеофайлов.

## Проверьте себя

1. Какие средства можно использовать для разработки веб-страниц? Объясните их преимущества и недостатки.
2. Что такое HTML-код страницы? Данные каких видов он содержит? Как можно его просмотреть?
3. Назовите известные вам веб-редакторы. В чем заключаются особенности их использования?
4. Что такое система управления веб-контентом? Назовите известные вам системы управления содержимым веб-сайта.
5. Какие вам известны системы управления веб-контентом, работающие в режиме он-лайн? В чем их особенности?
6. Назовите и объясните этапы автоматизированного создания веб-сайтов средствами веб-серверов.
7. Какие шаблоны веб-страниц используются при создании веб-сайтов средствами сервиса Сайты Google? Опишите их особенности.
8. Как вставить изображение на веб-страницу? Что может быть источником изображения при вставке на веб-страницу?
9. Как связать гиперссылку с текстовым фрагментом?
10. Как вставить видеообъект на веб-страницу? Что может быть источником видео?
11. Что такое гаджет?
12. Какие форматы графических, видео- и аудиофайлов рекомендуется использовать при размещении объектов на веб-страницах? Чем это объясняется?
- 13\*. Чем отличаются результаты операций вставки графических изображений и их загрузки?

## Выполните задания

1. Просмотрите указанные учителем HTML-страницы, например из папки **Тема 4\Задания 4.9\Упражнение 1**, созданные разными средствами. Сравните значки файлов веб-страниц, их содержание в окне браузера, HTML-код и размеры файлов. Заполните таблицу.

<i>Средство создания</i>	<i>Имя файла</i>	<i>Размер файла, в байтах</i>
<b>Блокнот</b>	<b>упражнение 4.9.1.1.html</b>	
<b>Microsoft Word 2007</b>	<b>упражнение 4.9.1.2.html</b>	
<b>KompoZer</b>	<b>упражнение 4.9.1.3.html</b>	
<b>Сайты Google</b>	<b>упражнение 4.9.1.4.html</b>	

Сделайте вывод, какое средство создания веб-страниц самое экономное с точки зрения трафика, затрачиваемого во время загрузки страницы.





- 2\*. Откройте документы, созданные с использованием разных программ пакета **Microsoft Office**, например из папки **Тема 4\Задания 4.9\Упражнение 2**. Сохраните эти документы в папке **Мои документы** с теми же именами, выбрав тип файла *веб-страница (\*.htm, \*.html)*. Обратите внимание на значки сохраненных файлов и создание дополнительных папок. Откройте файлы веб-страниц, сравните их содержимое и оформление с содержимым соответствующих документов. Оцените размеры файлов веб-страниц и дополнительных папок.
3. Заполните таблицу и сравните особенности создания веб-сайтов с использованием разных средств.

<i>Средство создания</i>	<i>Потребность в знании языка HTML</i>	<i>Наличие средств для оформления страниц и наполнения их содержимым</i>	<i>Удобство редактирования</i>	<i>Потребность в размещении на сервере хостинга</i>
Текстовый редактор				
Программа пакета Microsoft Office				
Веб-редактор				
Система управления содержимым веб-сайта				

4. Зарегистрируйте свой аккаунт на сервере **Google** в соответствии с алгоритмом, описанным в пункте.
- 5°. Создайте на сервере **Google** сайт в соответствии со структурой, приведенной на рисунке 4.56. Выберите тему оформления *Земля: вода*. Выберите такие шаблоны страниц: *История класса* - шаблон **Веб-страница**, *Учителя и предметы* - шаблон **Список**, *Поэтическое творчество* - шаблон **Объявления**, *Учебные материалы* - шаблон **Файловый менеджер**. Шаблоны других страниц выберите самостоятельно. Сообщите учителю URL-адрес вашего сайта.
- 6°. Создайте на сервере **Google** сайт с темой *Художественный салон*:
1. Выберите URL-адрес для сайта, соответствующий его названию.
  2. Выберите тему оформления *Мэдисон*.
  3. Просмотрите предложенную схему внутренней структуры сайта, например из файла **Тема 4\Задания 4.9\Упражнение 6\схемы.docx**.
  4. Заполните **Главную страницу** данными о назначении сайта, описанием материалов, которые будут размещены на страницах, и данными о вас как разработчике сайта.
  5. Создайте веб-страницы на основе шаблона **Веб-страница**, разместите их в соответствии с предложенной схемой внутренней структуры.
  6. Разместите на страницах текстовые и графические материалы, например из папки **Тема 4\Задания 4.9\Упражнение 6**, таким образом, чтобы текст и соответствующее изображение на странице были размещены рядом. Используйте для компоновки страниц модульную сетку *Два столбца (простой)*.
  7. Свяжите гиперссылки с текстовыми и графическими объектами так, чтобы, выбрав изображение, можно было открыть одну из информационных страниц сайта, а выбрав текстовый фрагмент - другую страницу.
  8. Загрузите на страницы сайта файлы, которые содержатся в папке **Тема 4\Задания 4.9\Упражнение 6\Файлы**.

# Глава 4

9. Направьте учителю информатики электронное письмо с адресом созданного сайта.
7. Откройте свой сайт, созданный во время выполнения задания 6, и отредактируйте его таким образом:
  1. Добавьте страницу *События* на основе шаблона **Объявления**. Добавьте на страницу запись с объявлением о выставке современных художников, которая состоится в художественном салоне.
  2. Вставьте на главную страницу таблицу, заполните ее расписанием работы художественного салона.
  3. Скомпонуйте содержимое информационных страниц, выбрав модульную сетку Два столбца. В подвале страниц укажите ваши данные как разработчика сайта.
- 8\*. Создайте на сервере **Google** сайт о своей будущей профессии.
- 9\*. Используя средства одного из специализированных веб-редакторов, разработайте свой персональный веб-сайт, разместите его на сервере бесплатно хостинга **TopUa** (<http://www.topua.net>).

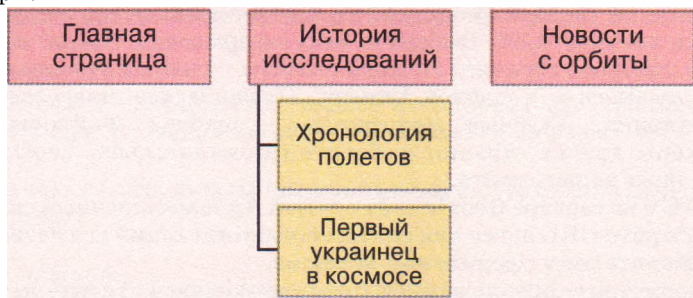


## Практическая работа № 18. Автоматизированное создание веб-сайта

**Внимание!** Во время работы с компьютером соблюдайте правила безопасности и санитарно-гигиенические нормы.

Создайте на сервере **Google** сайт на тему *Украинские исследователи космоса*:

1. Подберите URL-адрес для сайта, соответствующий его названию.
2. Выберите тему оформления *Блеск*.
3. Заполните *Главную страницу* описанием материалов, которые будут размещены на страницах сайта, и данными о вас как разработчике сайта.
4. Создайте и разместите веб-страницы на основе такой схемы внутренней структуры:



Выберите для страниц такие шаблоны: *История исследований*, *Первый украинец в космосе* - шаблон **Веб-страница**, *Новости с орбиты* - шаблон **Объявления**, *Хронология полетов* - шаблон **Список**.

5. Заполните страницу *История исследований* текстовыми материалами, например из файла **Тема 4\Практическая 18\космические исследования.docx**.
6. Создайте на странице *История исследований* гиперссылку для перехода на страницу *Новости с орбиты*, связав его с первым предложением на странице.
7. Разместите на странице *Первый украинец в космосе* текстовые и графические материалы, например из папки **Тема 4\Практическая 18\Попович**, таким образом, чтобы фотография размещалась по центру страницы, а текст - слева и справа от нее. Используйте для компоновки страницы модульную сетку *Три столбца (простой)*.
8. Создайте на странице *Первый украинец в космосе* гиперссылку для перехода на страницу с URL-адресом <http://h.ua/story/285307>, связав ее с графическим изображением.

9. Настройте на странице **Хронология полетов** нетипичный список так, чтобы в нем содержались о каждом полете такие данные: год полета, космонавт, космический корабль, описание. Заполните список данными о 3-4 полетах, например из файла Тема 4\Практическая 18\полеты.c10x.
10. Разместите на странице **Новости с орбиты** запись с последней новостью с сайта <http://space.vn.ua>.
11. Загрузите на страницу сайта **Новости с орбиты** файлы, которые содержатся в папке Тема4\Практическая 18\Файлы.
12. Сообщите учителю информатики URL-адрес вашего сайта.

## 4.10. Технологии и сервисы Веб 2.0. Веб-сообщества. Создание блогов





1. Что такое гаджеты?
2. Как создаются и редактируются статьи веб-энциклопедии **Википедия**?
3. Как можно найти нужную статью в веб-энциклопедии **Википедия**?
4. Что такое форум и чат? Какую структуру имеют форумы?

### Обзор технологий и сервисов Веб 2.0

В последнее время быстро развиваются сервисы Интернета, которые получили обобщенное название сервисы **Веб 2.0**.

Термином **Веб 2.0** называют несколько технологий, на основе которых разрабатываются современные веб-ресурсы. Такими технологиями являются:

- **AJAX** (англ. *Asynchronous JavaScript and XML* - асинхронный JavaScript и XML, языки программирования и разметки веб-страниц) - технология построения веб-страниц, при использовании которой возможна динамическая перезагрузка части содержимого веб-страницы в ходе просмотра ее в окне браузера;
- **RSS** (англ. *Really Simple Syndication* - действительно простое объединение) - технология экспорта гипертекста с одних ресурсов на другие, используется для получения обобщенных коротких сообщений с сайтов, которые интересуют пользователя, - **лент новостей**; 
- **FOAF** (англ. *Friend of a Friend* - друг друга) - технология, обеспечивающая пользователям возможность оформлять подписку на получение новых материалов от тех пользователей, данные которых занесены в так называемые **списки друзей**; 
- **трекбек** (англ. *track back* - путь назад) - технология отправки сообщения на сервер об установке гиперссылки на некоторый ресурс этого сервера и автоматического формирования обратной гиперссылки

и др.

Классические веб-ресурсы, которые теперь называют Веб 1.0 (статические веб-страницы, форумы, чаты и др.), функционируют на основе того, что разрабатывает ресурс один человек - разработчик, а пользуется другой - пользователь, наполняет ресурс информационными материалами автор, а пользователь является только читателем этих материалов. На информационные материалы распространяется авторское право. Единственный способ получения этих материалов - открыть страницы веб-сайта, на котором эти

материалы размещены. Ресурсы Интернета используются в основном как источник информационных материалов.

В отличие от ресурсов **Веб 1.0**, сервисам **Веб 2.0** присущи такие особенности:

- **веб как платформа** - ресурсы **Веб 2.0** используются для создания документов (презентаций, электронных таблиц, флэш-проектов и т.п.), которые можно сохранять как на локальном компьютере, так и он-лайн. Многие ресурсы **Веб 2.0** разрабатываются как программы, которые работают на веб-страницах. Они получили название веб-приложения. Их размещение в сети освобождает пользователя от необходимости установки соответствующих программ на локальном компьютере. Примером является использование он-лайн словарей и переводчиков, доступ к которым можно получить не только с компьютера, но и с мобильных устройств (коммуникаторов, смартфонов и др.), имеющих доступ к сети;
- **mash-up** (англ. *mash-up* - смешивание) — функциональность ресурсов Интернета расширяется за счет добавления новых сервисов к уже существующим, предыдущие разработки находят применение в новых ресурсах. Так, на многих веб-страницах можно увидеть гаджеты для воспроизведения видео, при этом сами видеофрагменты и механизмы управления ими содержатся на других веб-ресурсах;
- **«коллективный разум»** - пользователи активно участвуют в создании и оценивании контента ресурса, реализуют совместные проекты, а не остаются пассивными потребителями чужого контента. Примером является веб-энциклопедия **Википедия**, содержимое которой создается и улучшается совместными усилиями всех заинтересованных пользователей;
- **фолксномия** (англ. *folk* - народный, *taxonomy* - таксономия, принцип упорядочения) - народная классификация, практика коллективного распределения данных по категориям с использованием ключевых слов - тегов, которые пользователи ресурса подбирают самостоятельно для пометки содержания контента;
- **веб-синдикация** - ресурсы **Веб 2.0** динамические, отдельные элементы содержимого имеют собственные URL-адреса, и поэтому на них можно установить гиперссылки и написать комментарий. За счет использования технологий **RSS** и **FOAF** для просмотра обновленного содержимого сайтов, на которые оформлена подписка, или ресурсов пользователей из списка друзей можно не посещать сами сайты, а следить за их изменениями с использованием лент новостей;
- **социализация** - использование сервисов **Веб 2.0** способствует созданию сообществ в сети с целью обсуждения вопросов и совместной работы над проектами, при этом возможность выполнения индивидуальных настроек, создание собственного профиля пользователя выделяет каждую личность среди прочих

И др.

Как следствие социализации сервисы **Веб 2.0** получили название **социальных**. Назначением этих сервисов является создание условий для организации совместной деятельности и для общения пользователей Интернета.

Социальные сервисы можно классифицировать по видам деятельности пользователей в сети. Назовем отдельные виды деятельности и соответствующие сервисы **Веб 2.0** (табл. 4.7).



Таблица 4.7. Примеры сервисов Веб 2.0 по видам деятельности пользователей

<i>Вид деятельности</i>	<i>Сервисы Веб 2.0, примеры ресурсов</i>	<i>Описание сервиса</i>
Поиск данных	Социальные поисковые системы: Флексум ( <a href="http://flexum.ru">http://flexum.ru</a> ), Open Directory Project ( <a href="http://www.dmoz.org">http://www.dmoz.org</a> ), Квинтура ( <a href="http://quintura.ru">http://quintura.ru</a> )	Пользователи заполняют поисковые базы данных адресами и описанием интересных ресурсов Интернета, помечают их тегами, создавая перечень ресурсов, в пределах которого можно осуществлять поиск. Сформированные списки ресурсов для поиска могут размещаться на веб-страницах для осуществления поиска нужных пользователю данных только из ресурсов предложенного списка
Создание коллективных каталогов ссылок	Народные классификаторы (службы социальных закладок): БобрДобр ( <a href="http://bobrdobr.ru">http://bobrdobr.ru</a> ), Делишес ( <a href="http://del.icio.us">http://del.icio.us</a> ), Memori.ru ( <a href="http://memori.ru">http://memori.ru</a> )	Пользователи создают закладки на разные ресурсы Интернета, помечая их тегами. При необходимости найти ресурс, содержащий материал по определенной теме, достаточно на сайте сервиса выбрать соответствующие теги из полного перечня, и из помеченных ресурсов будет построен список ссылок
Совместная разработка новых информационных материалов	Вики-проекты: ЗапоВики ( <a href="http://wiki.ciit.zp.ua">http://wiki.ciit.zp.ua</a> ), Веб-энциклопедия Киева ( <a href="http://wek.kiev.ua">http://wek.kiev.ua</a> ), Викисловарь ( <a href="http://uk.wiktionary.org">http://uk.wiktionary.org</a> )	Пользователи сайта после регистрации имеют возможности создавать новые веб-страницы, размещая на них информационные статьи с текстовым и мультимедийным содержанием, редактировать собственные материалы и статьи, созданные другими пользователями, просматривать историю редактирования, связывать гиперссылками страницы этого сайта и устанавливать ссылки на внешние ресурсы, обсуждать размещенный материал
	Веб-хостинги со встроенными системами управления контентом: uCoz ( <a href="http://www.ucoz.ua">http://www.ucoz.ua</a> ), Сайты Google ( <a href="http://sites.google.com">http://sites.google.com</a> ), Блоги I.ua ( <a href="http://blog.i.ua">http://blog.i.ua</a> )	Пользователи могут создавать веб-сайты и наполнять их контентом непосредственно на серверах веб-хостинга. Веб-страницы создаются на основе готовых шаблонов. Посетители сайтов, как правило, имеют возможность оставлять комментарии на веб-страницах для высказывания собственных мнений относительно содержания сайта

<b>Вид деятельности</b>	<b>Сервисы Веб 2.0, примеры ресурсов</b>	<b>Описание сервиса</b>
	Он-лайн офисы: Документы Google ( <a href="https://docs.google.com">https://docs.google.com</a> ), Toffix.Ru ( <a href="http://toffix.ru">http://toffix.ru</a> ), Feng Office ( <a href="http://www.fengoffice.com">http://www.fengoffice.com</a> )	Пользователи могут создавать и хранить электронные документы в Интернете. Документ одновременно могут просматривать и редактировать несколько пользователей, для которых установлены соответствующие разрешения автором документа
Формирование карт знаний	Карты знаний: Gliffy ( <a href="http://www.gliffy.com">http://www.gliffy.com</a> ), Bubbl.us ( <a href="http://bubbl.us">http://bubbl.us</a> ), Mapul ( <a href="http://www.mapmyself.com">http://www.mapmyself.com</a> )	Пользователи в режиме он-лайн могут создавать и редактировать карты знаний (когнитивные карты) - анимированные графические изображения, которые визуализируют связи между объектами определенной предметной области
Хранение документов разных видов для общего использования	Социальные медиа-хранилища: он-лайн фотоальбом Фламбер ( <a href="http://flamber.ru">http://flamber.ru</a> ), сборник видео Рутьюб ( <a href="http://www.rutube.com">http://www.rutube.com</a> ), хранилище документов разных форматов Scribd ( <a href="http://www.scribd.com">http://www.scribd.com</a> ), хранилище аудиоматериалов Bee.fm ( <a href="http://www.bee.fm">http://www.bee.fm</a> )	Пользователи имеют возможность хранить на сервере собственные документы, помечать их тегами, воспроизводить в режиме он-лайн документы, графические изображения, презентации, видеоролики других пользователей, размещенные на сервере медиа-хранилища, оставлять собственные комментарии. Обеспечивается возможность доступа к ним с разных мест и разными пользователями, удобного поиска, общего использования документов.
	Файлообменные системы (файловый хостинг): Shareua.Com ( <a href="http://www.shareua.com">http://www.shareua.com</a> ), Upload.Com.Ua ( <a href="http://upload.com.ua">http://upload.com.ua</a> ), Fileshare ( <a href="http://fileshare.in.ua">http://fileshare.in.ua</a> )	Пользователи имеют возможность загружать на сервер для хранения файлы разных типов. Каждый файл получает собственный URL-адрес. При наличии адреса другие пользователи могут загрузить соответствующий файл с сервера на свой локальный компьютер. Благодаря этому осуществляется обмен файлами
Организация и планирование индивидуальной и совместной проектной деятельности	Он-лайн календари: Календарь Google ( <a href="http://www.google.com/calendar">http://www.google.com/calendar</a> ), Яндекс.календарь ( <a href="http://calendar.yandex.ru">http://calendar.yandex.ru</a> )	Пользователи имеют возможность фиксировать в он-лайн календаре запланированные события для организации индивидуальной и совместной деятельности, получать сообщения о приближении времени события, оставлять комментарии относительно каждого события

<b>Вид деятельности</b>	<b>Сервисы Веб 2.0, примеры ресурсов</b>	<b>Описание сервиса</b>
	<p><b>Органайзеры:</b>  <b>Organizeit.ru</b>            (<a href="http://organizeit.ru">http://organizeit.ru</a>),  <b>Business it online</b> (<a href="http://www.businessitonline.com">http://www.businessitonline.com</a>)</p>	<p>Он-лайн органайзеры объединяют несколько функций для организации деятельности: календарь, адресную книгу, записную книжку, программу для планирования заданий, систему напоминания о событиях с использованием электронной почты, IM-мессенджера или SMS. Пользователям предоставляется возможность совместно редактировать записи и использовать названные средства</p>
	<p><b>Он-лайн менеджеры проектов:</b>  <b>ZohoProject</b>            (<a href="http://projects.zoho.com">http://projects.zoho.com</a>),  <b>EasyProjects</b> (<a href="http://www.easyprojects.net">http://www.easyprojects.net</a>)</p>	<p>Веб-приложения предназначены для предоставления группам пользователей услуг по управлению проектами, их планированию, общению в процессе выполнения работ, наблюдению за ходом проекта, подготовке отчетов и т. п.</p>
<p>Определение географического положения и прокладывание маршрутов</p>	<p><b>Геосервисы:</b>  <b>Карты Google</b> (<a href="http://maps.google.com">http://maps.google.com</a>),  <b>Панорамно</b> (<a href="http://www.panoramio.com">http://www.panoramio.com</a>),  <b>Google Планета Земля</b>            (<a href="http://earth.google.com">http://earth.google.com</a>)</p>	<p>Пользователи могут находить объекты на карте земной поверхности, определять их географические координаты, связывать фотографии с объектами или координатами, комментировать их, прокладывать маршруты через определенные пункты, просматривать спутниковые фотографии поверхности Земли и т. п.</p>
<p>Общение</p>	<p><b>Блоги</b> (англ. <i>web log</i> - сетевой журнал) - <b>сетевые дневники:</b>  <b>Украинская блогосфера</b>            (<a href="http://blogosphere.com.ua">http://blogosphere.com.ua</a>),  <b>Blogreader</b> (<a href="http://biogoreader.org.ua">http://biogoreader.org.ua</a>),  <b>Блог газеты «Украинская правда»</b> (<a href="http://blogs.pravda.com.ua">http://blogs.pravda.com.ua</a>)</p>	<p><b>Блоги</b> - это веб-сайты, которые создаются для размещения автором сообщений с изложением своих мыслей, размышлений о тех или других событиях, презентации собственного творчества и т.п. Сообщения в блогах имеют собственные URL-адреса, поэтому их можно найти средствами поисковых систем. Каждый пользователь Интернета может оставить свой комментарий на сообщение в блоге, если это разрешено <b>блоггером</b> - автором блога</p>

Вид деятельности	Сервисы Веб 2.0, примеры ресурсов	Описание сервиса
	<b>Ресурсы социальных сетей:</b> Connect.ua ( <a href="http://connect.ua">http://connect.ua</a> ), PROFEO ( <a href="http://www.profeo.ua">http://www.profeo.ua</a> ), <b>ВКонтакте</b> ( <a href="http://vkontakte.ru">http://vkontakte.ru</a> ), <b>Мой Мнр</b> @mail.ru ( <a href="http://my.mail.ru">http://my.mail.ru</a> )	Сайты, предназначенные создавать условия для общения группе пользователей, объединенных общими интересами, вкусами, местом учебы, работы, отдыха и т. п. В ходе регистрации и участия в обсуждениях пользователи наполняют сайт содержанием. Сайты социальных сетей имеют средства поиска учетных записей участников сетевого сообщества и обмена сообщениями между ними

При значительной популярности сервисов Веб 2.0 нужно помнить о проблемах, связанных с их распространением. В результате доступности этих сервисов ресурсы Интернета заполняются некачественными информационными материалами, снижающими доверие к электронным источникам. Массово нарушаются авторские права на объекты интеллектуального труда. Возможность высказываться анонимно снимает моральную ответственность за содержание сообщений. Кроме того, регистрация на ресурсах социальных сетей может привести к незаконному использованию личных данных пользователей.

### Вики-технологии

В числе сервисов **Веб 2.0** особое место занимают вики-проекты. Основное их назначение - накопление и структурированное хранение информационных материалов с возможностью редактирования и обсуждения этих материалов группами пользователей. Самый известный из вики-проектов - свободная мультязычная веб-энциклопедия **Википедия**, с особенностями использования которой в учебном процессе вы ознакомились в 10-м классе.

Вики-проекты разработаны на основе технологии построения веб-сайтов WikiWiki — системы хранения и структурирования базы знаний, вывода их в гипертекстовом формате, обеспечения навигации и поиска.

Вики-технологии предоставляют пользователю такие возможности:

- создать новую страницу с данными (статью) может каждый посетитель вики-проекта, зарегистрированный в системе, при этом разрешение на просмотр и редактирование страницы может быть предоставлено лишь определенной группе пользователей;
- каждая статья имеет собственное имя, уникальное в пределах вики-проекта, которое используется для установки гиперссылки на нее;
- возможна одновременная обработка статьи в вики-проекте несколькими пользователями;
- возможно многократное редактирование статьи вики-проекта с сохранением истории редактирования;
- отображение изменения в статьях происходит сразу после сохранения отредактированной страницы;



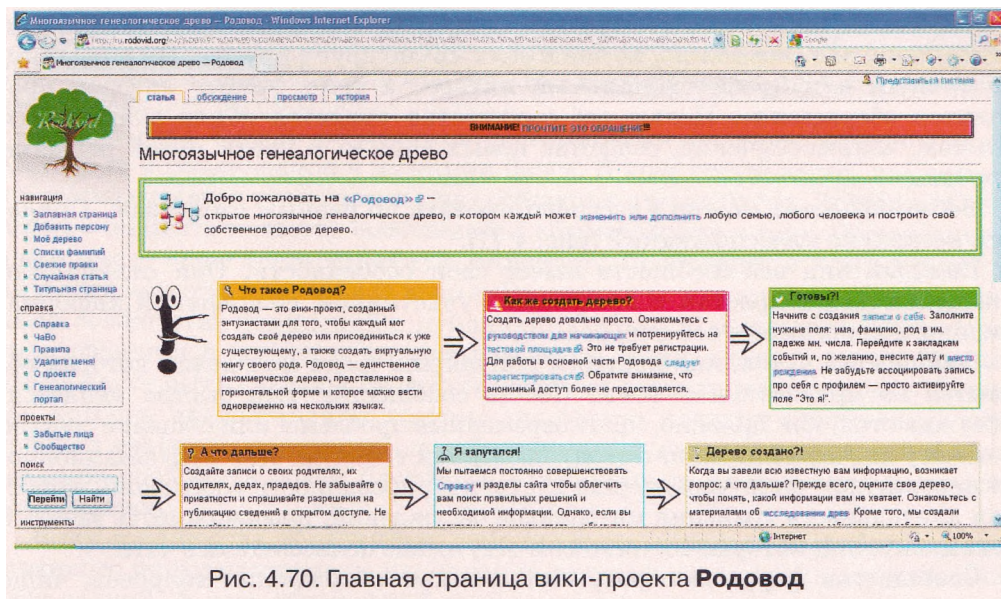


Рис. 4.70. Главная страница вики-проекта Родовод

- возможно возвращаеми к предыдущим версиям статьи, если изменения, по мнению автора статьи или администратора проекта, ухудшили ее качество;
  - пользователю предоставляются удобные средства для форматирования символов, вставки таблиц, изображений и других мультимедийных объектов, создания гиперссылок;
  - широко используется механизм меток-тегов для объединения материалов веб-страниц проекта по категориям и легкого поиска данных в базе знаний;
  - создание новой статьи осуществляется путем создания гиперссылки на нее в любой существующей статье
- и др.

С использованием вики-технологий разработаны энциклопедические ресурсы, например OpenWiki (<http://www.openwiki.org.ua>) - проект свободной украинской энциклопедии, Майдан Вики (<http://www.maidan.org.ua/wiki>) - политическая энциклопедия, Веб-энциклопедия Киева (<http://wek.kiev.ua/uk>) и т. п., и ресурсы некоторых общественных, образовательных, политических организаций, например ЗапоВики (<http://wiki.ciit.zp.ua>) - сетевое объединение участников учебно-воспитательного процесса Запорожской области, Родовод (<http://ru.rodovid.org>) - многоязычное генеалогическое древо (рис. 4.70), Вики Кировоградского государственного педагогического университета им. В. Винниченко (<http://wiki.kspu.kr.ua>) и др.

## Веб-сообщества

Социализация сервисов Веб 2.0 находит свое отображение в том, как активно пользователи сервисов принимают участие в обсуждении многочисленных материалов, оставляя свои комментарии, общаются друг с другом, отслеживают новости, которые появляются на любимых ресурсах. Как следствие, из групп пользователей Интернет-ресурсов образуются веб-

сообщества. Чаще всего это группы людей, не связанных официальными отношениями и, возможно, не знакомых лично друг с другом. Их объединяет личный или профессиональный интерес к теме, вокруг которой проводится обсуждение. Целью веб-сообщества является обмен знаниями и опытом, взаимообучение, создание новых знаний и поиск путей решения проблем.

Веб-сообщества можно классифицировать по разным свойствам. Приведем несколько классификаций (рис. 4.71).

Каждый вид веб-сообществ имеет свои особенности. Они отличаются численностью, возрастом и профессией участников, спецификой информационного наполнения веб-ресурса.

Сообщества чатов, как правило, популярны среди молодежи, которая общается на произвольные темы. Состав сообщества чата быстро меняется. Обсуждаются, как правило, кратковременные проблемы или общение сведено к обмену впечатлениями по поводу последних событий. Для информационного наполнения чатов характерны неструктурированность сообщений, короткая длительность хранения материалов обсуждений, невозможность узнать о ходе дискуссий, которые состоялись в отсутствие пользователя в чате.

Сообщества форумов образуются с целью обсуждения вопросов, чаще всего в пределах определенной темы, часто профессионального направления. В ходе обсуждений, как правило, высказываются мнения относительно сообщений автора темы и комментариев других членов сообщества. Информационное наполнение форумов структурировано, сохраняется длительное время, подается в удобной форме. В сообществе реализована система оценивания активности участников обсуждений и возможность получения автором темы сообщений о новых публикациях в форуме. Разработаны правила общения на форумах, за соблюдением которых следят модераторы.

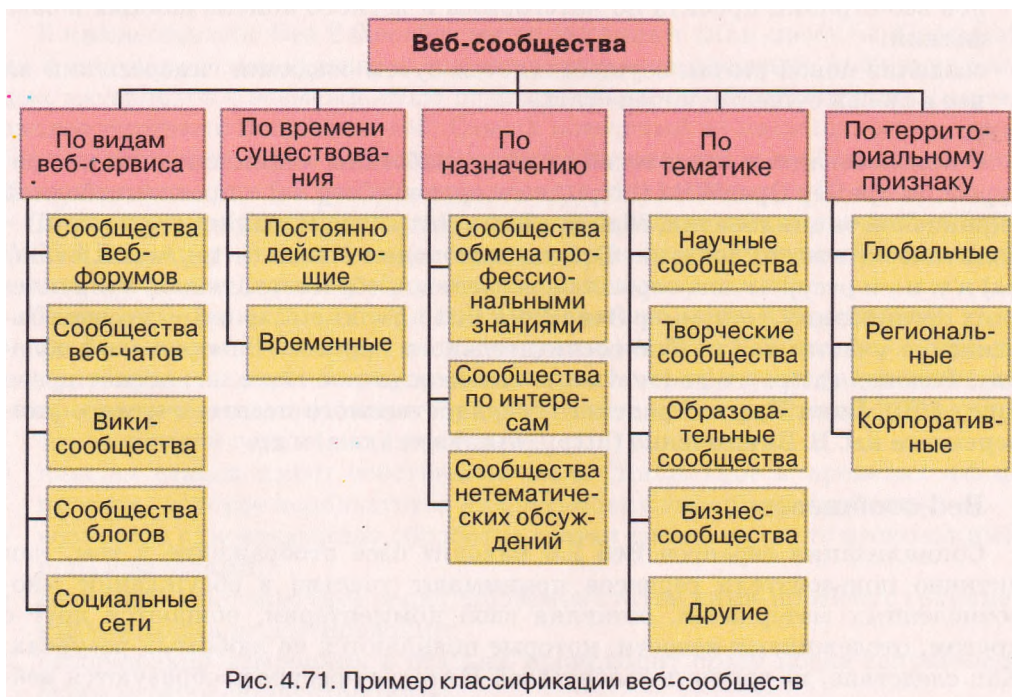


Рис. 4.71. Пример классификации веб-сообществ

Существует механизм ограничения доступа к обсуждениям. Для сообществ форумов характерно объединение при обсуждении одних проблем и изменение состава при рассмотрении других.

Сообщества блогов образуются для общения по теме, предложенной блоггером и объявленной в названии блога. В ходе обсуждений, как правило, высказывается точка зрения на сообщения автора блога. Читатели блога могут оценивать качество сообщений и комментариев к ним. Обсуждение в блоге имеет чаще всего линейную структуру. Сообщения и комментарии к ним сохраняются длительное время. Реализованы средства получения членами сообщества извещений о публикации новых сообщений в блоге. Правила общения и доступ к материалам блога устанавливает автор. Разные блоги могут быть связаны гиперссылками по технологии «трекбек». Службы, которые предоставляют хостинг для блогов, создают собственные внутренние механизмы, которые способствуют распространению связей и образованию сообществ среди блоггеров. Совокупность блогов сети называют блогосферой.

Вики-сообщества образуются, как правило, с целью разработки качественного информационного наполнения ресурса, создания так называемой сети знаний. В ходе обсуждений уточняются формулировки понятий, формируется стиль энциклопедических и справочных статей, дополняются ссылки на источники данных и на внешние ресурсы и т. п. Вики-сайты часто используют для организации учебных проектов, дистанционного обучения, что способствует формированию образовательных сообществ. Информационные материалы сообщества хорошо структурированы, связаны многочисленными связями. Правила общения и доступ к материалам вики-сообщества устанавливают члены сообщества в ходе обсуждений.

Социальные сети, образованные на веб-ресурсах, формируются, как правило, на основе личного знакомства членов сообщества и последующего расширения сети знакомств. Объединение в сообщества осуществляется на основе личных и профессиональных интересов. Информационное наполнение слабо структурировано. Статус членов сообщества и состав администрации формируются на основе активности в обсуждениях.

## Блоги. Разновидности блогов

Сервис сетевых дневников - блогов - стал альтернативой персональным веб-сайтам. Как и персональный сайт, блог используется для выражения личной позиции владельца, но в блогах важной составляющей являются комментарии посетителей, их диалог с автором блога.

Блог построен как лента сообщений, размещенных на веб-странице в хронологическом порядке, к которым посетители блога могут оставлять свои комментарии. Последние сообщения автора располагаются, как правило, в верхней части веб-страницы, комментарии к сообщению - под ним (рис. 4.72). В тексте сообщения можно размещать мультимедийные фрагменты. У каждого сообщения блога есть заголовок и собственный URL-адрес.

Для блогов характерны:

- краткие сообщения;
- регулярное обновление информационного наполнения;
- своеобразный диалог между автором и читателями;



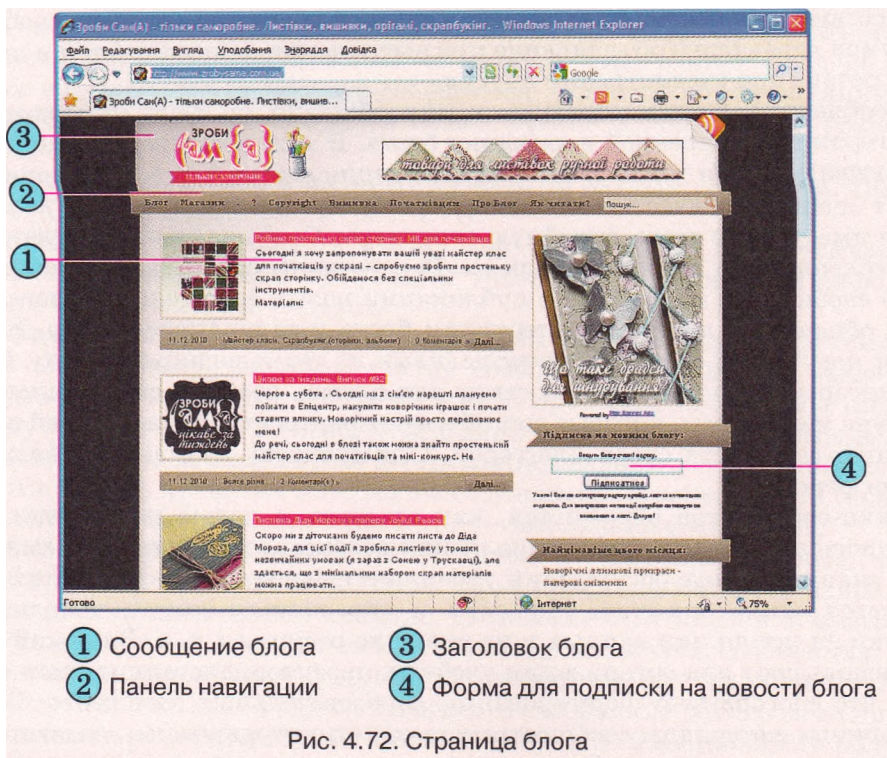


Рис. 4.72. Страница блога

- возможность получения извещений о публикации новых сообщений;
- наличие единой для всех сообщений идеи, объявленной в названии блога.

Блоги можно классифицировать по разным признакам.

В зависимости от цели создания различают блоги:

личный - аналог личного дневника, содержит описание текущих событий и связанные с ними размышления автора;

- тематический - содержит сообщения по одной тематике;
- бизнес-блог - создается для продвижения определенного товара или услуги, содержит рекламу, обзор продукции, комментарии и т. п.;
- общественный - создается с целью общения с коллегами или клиентами, обществом;
- образовательный - создается с целью общения участников учебного процесса

и др.

По типу сообщений блоги делятся на:

текстовые - блоги, основным видом сообщений в которых являются текстовые сообщения;

фотоблоги - блоги, содержащие подборки фотографий;

- музыкальные - блоги, основным содержанием которых являются музыкальные записи;

блогкасты, или подкасты (*iPod* - портативный проигрыватель, англ.

*broadcast* - передача, вещание) - блоги, содержанием которых являются аудиофайлы, надиктованные и опубликованные в сети автором;

видеоблоги - основное содержание блогов составляют видеоматериалы.



Блоги можно создать на сайтах **Blog.net.ua** (<http://blog.net.ua>), Блоги@**Mail.Ru** (<http://blogs.mail.ru>), **Blogger** (<http://blogger.com>) и др.

## Создание блога. Публикация сообщений

Рассмотрим процесс создания блога на примере сервиса **Blogger** (<http://blogger.com>). Для пользования им нужно иметь аккаунт **Google**.

Для создания блога нужно:

1. Открыть в окне браузера главную страницу сервиса **Blogger** (<http://blogger.com>).
2. Ввести данные вашей учетной записи **Google**.
3. Выбрать на странице **Blogger**: Панель инструментов гиперссылку **Создайте блог**.
4. Ввести в поле **Название блога** заголовок вашего блога (рис. 4.73).
5. Ввести в поле **Адрес блога (URL)** часть URL-адреса блога. Полный URL-адрес будет иметь вид <http://адрес.blogspot.com>, где *адрес* - введенный пользователем текст.
6. Выбрать гиперссылку **Проверить доступность** для проверки адреса на уникальность. При необходимости изменить адрес.
7. Выбрать гиперссылку **Продолжить**.
8. Выбрать начальный шаблон оформления блога из предложенных. Выбрать гиперссылку **Продолжить**.
9. Выбрать на странице **Blogger**: Процесс успешно завершен соответствующую гиперссылку для последующих действий:
  - добавить в него сообщения - для создания и публикации первого сообщения в блоге. Такое же действие будет выполнено при выборе гиперссылки **Перейти к блогу**;
  - настроить его внешний вид - для выбора шаблона оформления, фона, разметки страницы, цветовой гаммы, шрифтов и др.;

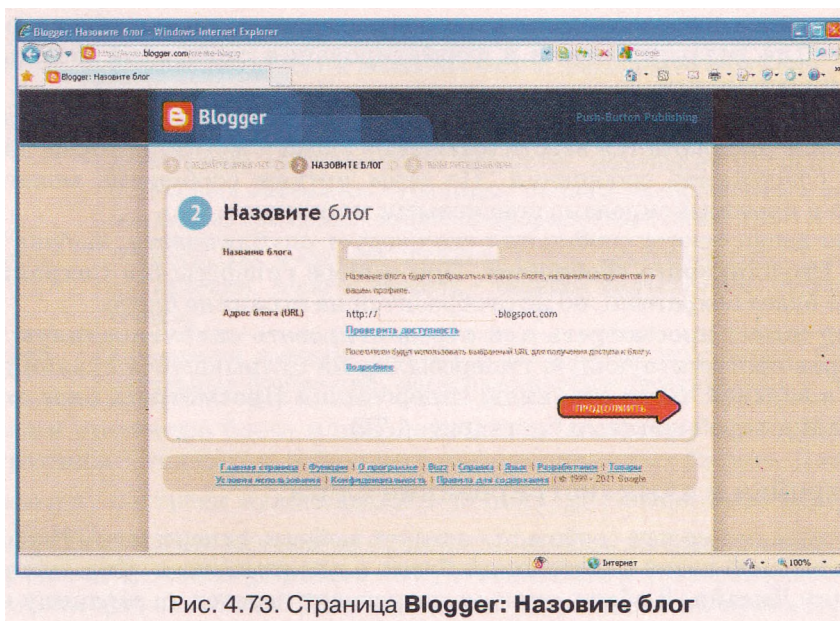


Рис. 4.73. Страница **Blogger**: Назовите блог

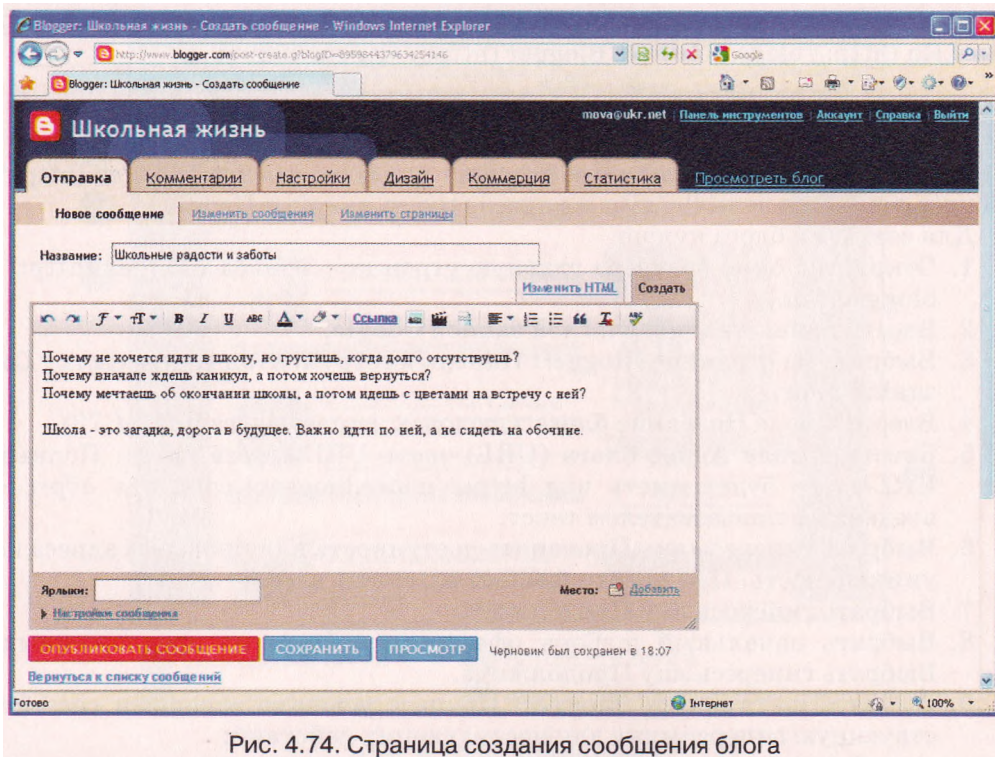


Рис. 4.74. Страница создания сообщения блога

- Дополнительные параметры настройки - для изменения домена и импортирования существующего блога в текущий.

При выборе первой из названных гиперссылок откроется страница Blogger: *Название блога* - Создать сообщение, где *Название блога* - название, которое было введено пользователем (рис. 4.74). Для создания сообщения следует ввести его заголовок в соответствующем поле и текст в рабочей области. Для редактирования и форматирования сообщений используется панель инструментов.

Ниже рабочей области размещено поле Ярлыки для этой публикации. В этом поле предлагается ввести ключевые слова - метки, которые характеризуют содержание сообщения. По этим меткам сообщение может быть найдено и проиндексировано поисковыми машинами.

После ввода текста сообщения его следует опубликовать, выбрав гиперссылку Опубликовать сообщение. При выборе гиперссылки Сохранить сообщение будет сохранено, но не отобразится на странице блога.

Автор может просмотреть или отредактировать опубликованную заметку, выбрав соответствующую гиперссылку на открывшейся странице. При выборе в верхней части страницы гиперссылки Просмотреть блог откроется страница с сообщениями блога (рис. 4.75).

## Настройка и администрирование блога

Для создания нового сообщения следует выбрать гиперссылку Новое сообщение в верхней части веб-страницы блога, а для изменения оформления - гиперссылку Дизайн. В обоих случаях происходит переход на страницу настро-

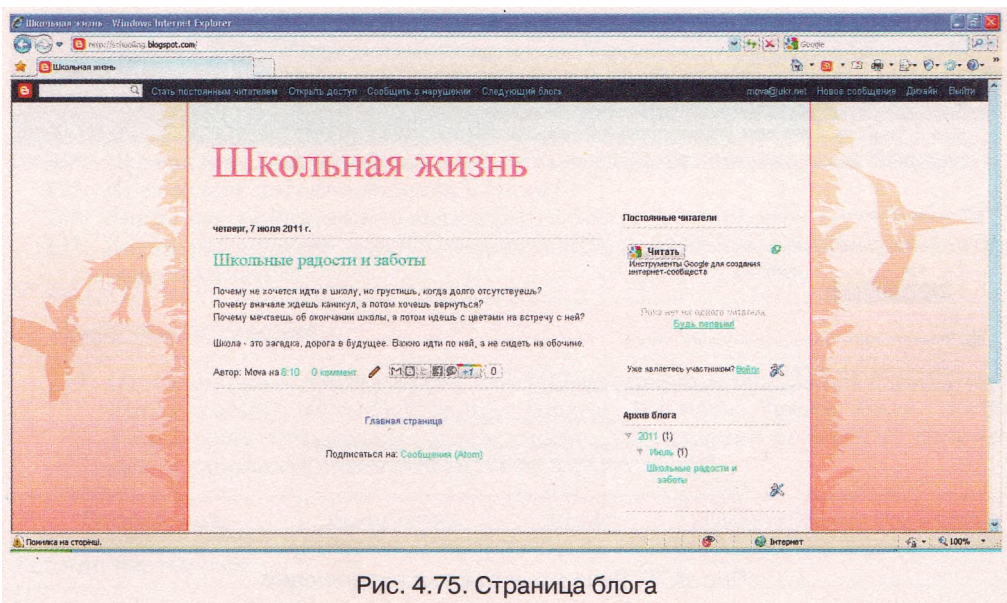




Рис. 4.75. Страница блога

ек блога (рис. 4.74), которая имеет несколько вкладок. Каждая из вкладок содержит элементы управления для выполнения отдельных настроек (табл. 4.8).

Таблица 4.8. Назначение вкладок настройки блога

Вкладка	Назначение
Отправка	Создание, редактирование, сохранение, предварительный просмотр и публикация сообщений автором блога
Комментарии	Обработка комментариев. Каждый комментарий к сообщениям автор блога может пометить как спам, удалить полностью или удалить только содержание комментария
Настройки	Изменение автором названия блога, добавление описания, включение в списки блогов, которые просматриваются поисковыми системами, установка разрешений, форматирование блога и др.
Дизайн	Добавление автором блога гаджетов, изменение расположения отдельных элементов страницы, вставка изображения в заголовки блога, изменение шаблона оформления блога и др.
Коммерция	Регистрация в системе AdSense (англ. <i>advertisement</i> - реклама, объявление, <i>sense</i> - смысл) - сервисе контекстной рекламы от Google
Статистика	Получение автором данных о количестве просмотров блога и отдельных сообщений, местонахождении посетителей и т. п.

Во время просмотра блога можно выполнить настройку отдельных объектов страницы - гаджетов, отдельных информационных блоков. Для перехода к режиму настроек нужно выбрать кнопку . Выполнить переход на страницу редактирования текстов сообщений можно, выбрав кнопку .

Автор может администрировать блог, устанавливая пользователям права доступа к материалам блога. По умолчанию разрешение на просмотр сооб-



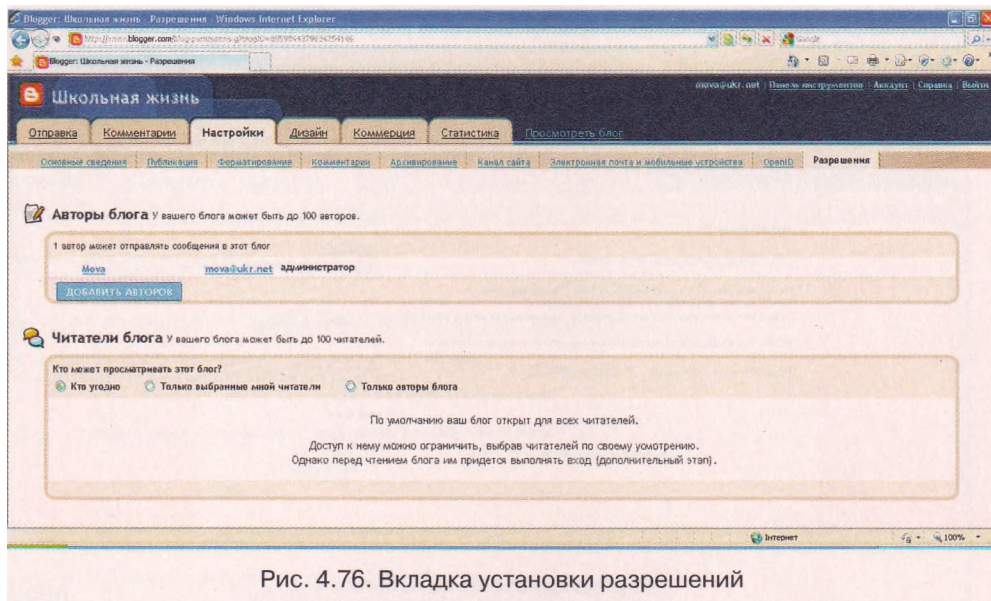


Рис. 4.76. Вкладка установки разрешений

щений блога предоставлено всем пользователям Интернета. Право публиковать новые сообщения имеет только автор, оставлять комментарии к сообщениям - только зарегистрированные пользователи. Эти настройки можно изменить. Для этого нужно:

1. Открыть страницу настроек блога.
2. Выбрать гиперссылку **Комментарии** на вкладке **Настройки**.
3. Указать, какие группы пользователей имеют право оставлять комментарии, выбором соответствующего переключателя в группе **Кто может отправлять комментарии?**
4. Выбрать гиперссылку **Разрешения** на вкладке **Настройки** (рис. 4.76).
5. Выбрать кнопку **Добавить авторов**, чтобы предоставить право другим пользователям создавать сообщения в блоге, указать адреса их электронных почтовых ящиков.
6. Установить, каким группам пользователей будет разрешено читать публикации блога, выбором соответствующего переключателя в области **Читатели блога**.

## Проверьте себя

1. Какие технологии положены в основу разработки сервисов Веб 2.0? В чем заключаются эти технологии?
2. Каковы особенности сервисов Веб 2.0? Поясните их.
3. Какие существуют типы сервисов Веб 2.0 по видам деятельности пользователей?
- 4\*. Чем, на ваш взгляд, объясняется популярность сервисов Веб 2.0?
5. Какие проблемы связаны с распространением сервисов Веб 2.0?
- 6°. Каковы возможности вики-технологий?
7. По значениям каких свойств можно классифицировать веб-сообщества? Поясните особенности каждого вида.
- 8°. Что такое блог?



- 9\*. Чем, на ваш взгляд, объясняется популярность блогов у пользователей?  
Почему блоги считаются альтернативой персональным веб-страницам?
10. По значениям каких свойств можно классифицировать блоги? Объясните особенности каждого вида.
- 11\*. Какие вы знаете сайты, предназначенные для создания блогов?
- 12\*. Приведите алгоритм создания блога с использованием сервиса Blogger.
- 13\*. Как создать новое сообщение в блоге?
14. Какие настройки можно выполнять в блогах?
15. Какие разрешения можно предоставлять пользователям при администрировании блога?



 **Выполните задания**

- 1\*. Ознакомьтесь с материалами сайтов **Википедия** (<http://uk.wikipedia.org>), **Flickr** (<http://www.flickr.com>), **Google Планета Земля** (<http://earth.google.com>). Определите, в чем состоит применение названных технологий, и заполните сравнительную таблицу.


<i>Технология</i>	<i>Википедия</i>	<i>Flickr</i>	<i>Google Планета Земля</i>
Веб как платформа			
Mash-up			
Коллективный разум			
Фолксномия			
Веб-синдикация			
Социализация			

2. Установите соответствие между видами деятельности человека и видами сервисов **Веб 2.0**.

	<i>Вид деятельности</i>		<i>Сервис Веб 2.0</i>
1	Поиск данных	А	Народные классификаторы
2	Совместная разработка информационных материалов	Б	Социальные поисковые системы
3	Хранение документов	В	Менеджеры проектов
4	Создание коллективных каталогов ссылок	Г	Блоги
5	Определение географического положения	Д	Вики-проекты
6	Общение	Е	Геосервисы
7	Организация совместной проектной деятельности	Ж	Социальные медиа-хранилища

-  3\*. Используя данные из таблицы 4.7, создайте иерархическую диаграмму — схему классификации сервисов **Веб 2.0** по видам деятельности пользователя.
-  4. Сравните особенности функционирования двух вики-проектов **Родовод** (<http://ru.rodovid.org>) и **Веб-энциклопедии Киева** (<http://wek.kiev.ua/uk>) и заполните таблицу **4.9**.

<i>Вопрос для сравнения</i>	<i>Сайт 1</i>	<i>Сайт 2</i>
Какие материалы представлены в проекте?		
Какими средствами сообщается тематика сайта?		
Какие средства предусмотрены для перехода между страницами сайта?		
Как получить справку об особенностях работы с материалами сайта?		
Как можно найти нужные материалы на сайте?		
Какие средства предлагаются для редактирования статей?		
Нужна ли регистрация в проекте для участия в редактировании статей?		
Какие средства предлагаются для просмотра истории редактирования?		
Какие средства предлагаются для создания новой статьи?		
Как организовано обсуждение материалов?		
Нужна ли регистрация в проекте для участия в обсуждениях материалов?		
Когда на сайте были внесены последние изменения?		

 5. Сравните особенности функционирования сайтов ЭкоАгроВики (<http://agrowiki.nubip.edu.ua>) и Фармацевтическая энциклопедия (<http://www.pharmencyclopedia.com.ua>) и заполните таблицу 4.9.

6\*. Заполните сравнительную таблицу веб-сообществ.

<i>Свойство</i>	<i>Сообщество</i>				
	<i>Чат</i>	<i>Форум</i>	<i>Вики</i>	<i>Блог</i>	<i>Социальная сеть</i>
Цель образования сообщества					
Длительность хранения материалов					
Структурированность материалов					
Наличие системы оценивания активности членов сообщества					
Кто устанавливает правила общения					
Наличие механизма ограничения доступа к обсуждениям					
Наличие средств получения извещений о новых материалах					
Возможность поиска нужных материалов					
Возможность редактирования информационных материалов каждым членом сообщества					



- 7°. Создайте иерархическую диаграмму - схему классификации блогов.
- 8°. Просмотрите блог Украинская блогосфера (<http://blogosphere.com.ua>). Проанализируйте дизайн блога, его структуру, сообщения и комментарии к ним. Определите, что из данного блога вам было бы интересно использовать в собственном блоге.
- 9°. Создайте собственный блог с использованием сервиса Blogger (<http://blogger.com>). Посвятите ваш блог обсуждению особенностей содержания домашних животных. Выберите шаблон оформления из категории *Водяной знак*. Разместите в блоге сообщение о животном, которое вы хотели бы держать дома. Сообщите учителю информатики адрес вашего блога.
- 10°. Добавьте сообщение в ваш блог о домашнем животном вашего друга.
- 11°. Измените настройки вашего блога: в заголовке добавьте описание блога, разместите фотографию домашнего животного, например из папки Тема 4\Задания 4.10\Животные. Добавьте гаджет *Статистика блога*. Измените макет тела страницы блога, разместив область гаджетов слева от области сообщений.
- 12°. Измените разрешения на работу пользователей с вашим блогом: разрешите оставлять комментарии в блоге всем, включая анонимных пользователей, создавать сообщения в блоге учителю информатики. Отправьте приглашение на адрес электронной почты вашего учителя информатики быть одним из авторов вашего блога.
- 13°. Просмотрите блог вашего одноклассника. Оставьте комментарий к одному из сообщений.

### Практическая работа № 19. Создание и ведение собственного блога

Внимание! *Во время работы с компьютером соблюдайте правила безопасности и санитарно-гигиенические нормы.*

Создайте собственный блог с использованием сервиса Blogger (<http://blogger.com>):

1. Посвятите ваш блог обсуждению вашей будущей профессии. Подберите соответствующее название блога.
2. Выберите шаблон оформления из категории *Путешествие*.
3. Разместите в блоге сообщение о профессии, которая вам нравится.
4. Измените настройки вашего блога:
  - а) в заголовок добавьте описание блога;
  - б) разместите в заголовке изображение атрибутов профессии, например из папки Тема 4\Практическая 19\Профессии;
  - в) добавьте к блогу гаджет Окно поиска, разместив его ниже заголовка блога;
  - г) измените макет тела страницы блога, разместив область гаджетов в две колонки слева и справа от области сообщений.
5. Добавьте сообщение в ваш блог, рассказав о профессии одного из ваших родителей или знакомых и вашем отношении к этой профессии.
6. Измените разрешения на работу пользователей с вашим блогом:
  - а) разрешите оставлять комментарии в блоге только пользователям учетных записей Google;
  - б) читателями блога назначьте только выбранных вами лиц, пригласите трех одноклассников почитать ваш блог;
  - в) отправьте приглашение учителю информатики быть одним из авторов вашего блога.

## 4.11. Совместная работа над документами



1. Назовите известные вам сервисы **Веб 2.0**. Какие из них предназначены для организации общения пользователей?
2. Для чего предназначены он-лайн офисы? Какие их возможности?
3. Какие сервисы Google вы знаете? Охарактеризуйте их назначение.
4. Что такое учетные записи? В каких службах они используются?
5. Какие правила работы на форуме и в чате?

### Системы коллективного взаимодействия

С распространением Интернета кардинально изменилось понятие рабочего коллектива. Люди, которые живут в разных странах и работают в разных организациях, могут принимать участие в общей разработке проектов или в работе над одними и теми же документами. Сегодня уже никого не удивит сотрудниками, которые никогда не заглядывают в офис компании, а выполняют свою работу, не выходя из дома. Другие - постоянно ездят по миру и в то же время принимают участие в различных совещаниях, семинарах, находясь в дороге.

Достаточно часто сотрудники, которые работают над одним проектом, находятся в разных зданиях, кабинетах и др. В таких случаях для организации документооборота между участниками одного проекта необходимо организовать *виртуальный офис*, где будут сохраняться документы для их общего пользования. Главная проблема такой организации деятельности заключается в доступе всех участников к общим документам и в организации коллективного общения удаленных друг от друга участников. Для этого можно использовать известные вам средства - электронную почту, чаты, службы мгновенного обмена сообщениями, различные сервисы Веб 2.0.

Также существует множество специально разработанных программ для организации групповой работы и использования общих документов, например Google Wave, 12Sprints, Sharepoint, Novell Pulse, IPI.Manager, Feng Office, Microsoft Office Groove 2007 и др. Такой вид программного обеспечения пока еще не получил однозначного названия, но чаще всего эти программы называют *системы коллективного взаимодействия*. Они могут использоваться как в локальных, так и в глобальных сетях.

Основное назначение систем коллективного взаимодействия:  
доступ к общим ресурсам некоторой группы участников;  
организация общения между участниками группы;  
координация и документирование хода совместных работ.

### Организация совместной работы в Microsoft Office Groove 2007

Одной из программ для организации коллективной работы и виртуального офиса является программа Microsoft Office Groove 2007, которая входит в пакет прикладных программ Microsoft Office 2007.

Она предоставляет пользователям среду для проведения онлайн-встреч, обсуждения общих проектов, передачи файлов непосредственно с одного компьютера на другой, хранения общих документов в едином для всех хранилище, доступ к которому возможен с разных компьютеров, автоматического обновления файлов в хранилище при их изменении кем-либо из участников группы, хранения разных версий файлов и т. п. Также программа дает возможность пользователям узнать о присутствии участников



рабочей группы в общем пространстве, просматривать сведения о них, направлять им сообщение и др. Groove 2007 включает в себя службы мгновенного обмена сообщениями, чат, а также возможности совместной работы над документами. Взаимодействие участников происходит в так называемых рабочих областях, где все участники группы, программные средства и общие документы концентрируются в одной зоне.

Для работы с программой нужно иметь подключение к сети Интернет. Для запуска программы следует выполнить *Пуск => Все программы => Microsoft Office => Microsoft Office Groove 2007*.

Если программа запускается впервые, то на экране отобразится Мастер настройки учетной записи, в окне которой нужно выполнить такие шаги:

1. Выбрать ответ на вопрос - хотите ли вы создать новую учетную запись для регистрации в этой программе или у вас уже есть такая?
2. Ответить на вопрос - имеете ли вы код конфигурации учетной записи Groove? Если эти данные неизвестны, то следует выбрать переключатель У меня нет кода конфигурации учетной записи.
3. Указать имя пользователя, адрес электронной почты и пароль для доступа к учетной записи.
4. Выбрать вид регистрации созданной учетной записи. Желательно выбрать режим *в общем каталоге Groove* - это упростит другим участникам поиск сведений о вас и приглашение вас в группу.

Если использование программы происходит не первый раз, то после запуска программы нужно в текстовое поле ввести пароль для доступа к учетной записи пользователя.

После того как вход в программу будет осуществлен, на экране отобразится основное окно программы (рис. 4.77), в котором расположены меню программы, две вкладки Рабочие области и Контакты, а также область Команды.

*Рабочая область* - это созданное одним из пользователей место, которое все участники группы используют для размещения документов и обсуждения того или иного вопроса. Ее можно сравнить с отдельной комнатой в чате. В Groove 2007 можно создать рабочие области трех типов, которые отличаются начальным набором доступных инструментов:

- Стандартная - предназначена для создания виртуальной среды общения участников и размещения файлов для общего пользования.
- Совместный доступ к файлу - предназначена для создания папки на локальном компьютере участника группы, содержание которой будет предоставлено для общего доступа другим участникам группы.
- Шаблон - дает возможность создать рабочую область, выбрав необходимые средства, которые будут использоваться участниками группы в совместной работе:
  - о Альбом - для общего рисования;
  - о Блокнот - для создания кратких сообщений;

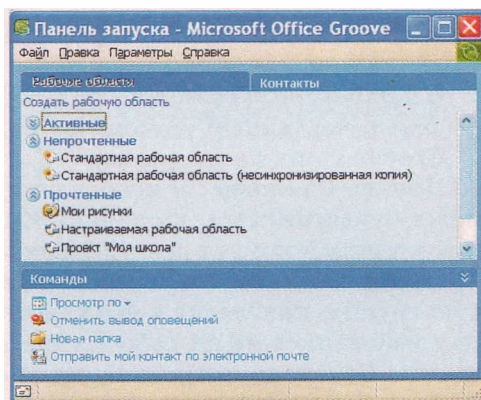


Рис. 4.77. Основное меню программы Microsoft Office Groove 2007

*Календарь* - для создания коллективного графика работы над проектом, собственного распорядка дня, назначения встреч;

*Обсуждение* - для ведения диалогов в стиле «вопрос-ответ»;

*Рисунки* - для добавления и просмотра изображений;

*Собрание* - для назначения повестки дня, поручений, ведения протокола совещания

и др.

Пользователь, который создал рабочую область, считается ее *Руководителем*. Ему предоставлены права на приглашение и исключение участников группы, подключение и отключение инструментов, смену роли определенного участника рабочей группы и выполнение других операций по управлению рабочей областью. Остальные пользователи выступают в роли *Участника* или *Гостя*. Участник может добавлять новые панели инструментов и приглашать новых участников в эту группу. Гостю же не предоставлены никакие разрешения, кроме участия в обсуждениях и работе над документами.

Для создания рабочей области нужно:

1. Выбрать в окне Панель запуска ссылку Создать рабочую область.
2. Выбрать тип создаваемой рабочей области (*Стандартная, Совместный доступ к файлу, Шаблон*).
3. Ввести название рабочей области (например, Проект «Моя школа»).
4. Выбрать кнопку ОК.

Созданная рабочая область появится в списке. Чтобы открыть ее, нужно дважды щелкнуть на ее имени, после чего откроется окно этой области. Например, на рисунке 4.78 представлено окно стандартной рабочей области, в которой для пользователей предоставляется два средства - *Файлы* и *Обсуждения*, ярлыки вкладок которых расположены в нижней части окна (9, 10).

В верхней части окна при открытой вкладке *Файлы* размещено меню (2) и панель инструментов (3), в левой части окна - дерево папок этой области (4), в центральной - их содержание (5). Используя инструменты и команды рабочей области, можно добавить в нее файлы или папки для общего использования.

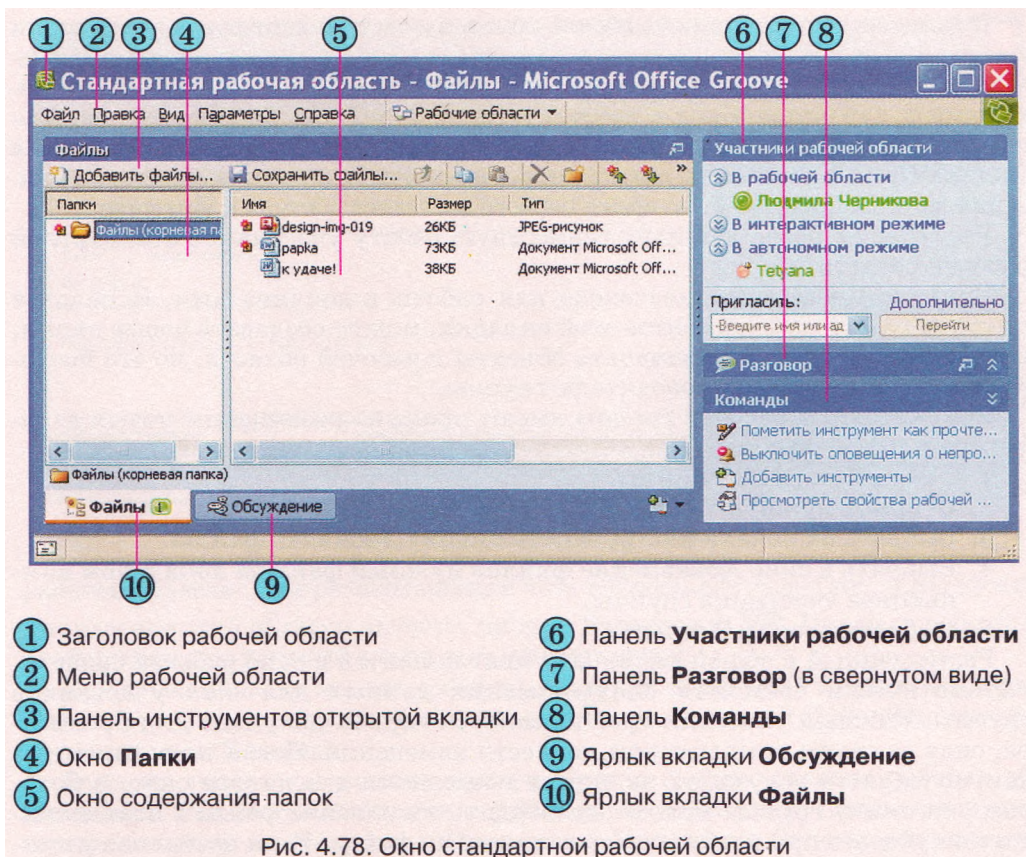
В правой части экрана отображается список участников группы (6). Все участники распределяются на три категории:


- *В рабочей области* - те участники, которые сейчас присутствуют в рабочей области.
- *В интерактивном режиме* - пользователи, которые сейчас имеют подключение к Интернету и установленную программу Groove, но не присоединились к рабочей группе.
- *В автономном режиме* - пользователи, которые присоединились к группе, но на данный момент не подключены к Интернету или программа Groove у них не запущена.

Панель *Разговор* (7) используется для организации текстового общения между участниками, на панели *Команды* (8) перечислены операции, которые в данный момент доступны участнику группы.

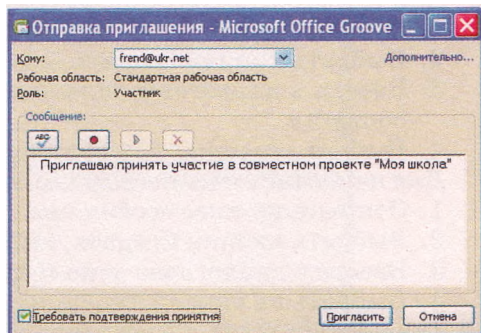
После создания совместной рабочей области нужно организовать рабочую группу, пригласив в нее участников. *Обращаем внимание*, что для этого у участников на их рабочих компьютерах должна быть установлена программа Groove и настроена учетная запись, также компьютеры участников должны быть подключены к сети Интернет.

Для приглашения новых членов группы необходимо в нижней части панели *Участники* рабочей группы в текстовом поле ввести электронный



адрес человека, которого планируется пригласить в группу, и выбрать кнопку **Перейти**. После чего в окне **Отправка приглашения** (рис. 4.79) выбрать роль участника, ввести текст сообщения и выбрать кнопку **Пригласить**. Если у вас подключен микрофон, то приглашение можно записать как голосовое сообщение, воспользовавшись кнопкой , которая размещена над окном сообщения.

Как только отправленное приглашение будет доставлено пользователю (как только данный пользователь запустит программу Groove и подключится к Интернету), его программа Groove известит об этом отображением текста сообщения в правом нижнем углу экрана. Когда приглашение будет принято, автору сообщения будет отправлено подтверждение аналогичным образом. А на мониторе у нового участника группы откроется окно программы Groove с содержанием общей рабочей области. Имена всех пользователей, которые приняли приглашение, появятся в списке панели **Участники рабочей группы**.





В дальнейшем участники рабочей области могут обмениваться текстовыми сообщениями, создавать и редактировать общие файлы, оставлять свои замечания, изменять интерфейс и содержание рабочей области. Все изменения автоматически будут сохраняться в рабочей области и будут автоматически пересылаться всем участникам группы - данный процесс называется *синхронизация данных*. При отсутствии доступа к сети кого-либо из членов группы синхронизация данных произойдет во время первого же сеанса доступа к Интернету.


Рассмотрим более детально совместную работу участников стандартной рабочей области.

Вкладка **Файлы** предназначена для работы с документами. Используя средства панели инструментов этой вкладки, можно создавать новые папки, копировать, удалять, перемещать объекты в рабочей области, но это позволяет делать только руководителю группы.

Все участники рабочей группы имеют право на размещение новых документов в общей области. Для этого следует выполнить такие действия:

1. Открыть вкладку **Файлы**.
2. Открыть нужную папку.
3. Выбрать на панели инструментов кнопку **Добавить файлы**.
4. Выбрать в окне **Добавление файлов** нужный файл на локальном компьютере участника группы.
5. Выбрать кнопку **Открыть**.

Размещенный в общей папке документ появится в окне рабочей области. Автоматически состоится синхронизация данных для всех участников группы. Каждый участник группы может открыть документ для просмотра, сидя за своим компьютером, и внести изменения. После закрытия отредактированного документа на экране появляется диалоговое окно, в котором участнику группы нужно подтвердить сохранение файла с изменениями или указать при необходимости новое имя файла. Если несколько участников будут одновременно записывать один и тот же документ, то программа сообщит об этом и создаст копию документа с другим номером версии.

В списке возле имен измененных объектов будет расположен значок . Это предупреждение участникам группы, что они еще не просмотрели новую версию документа. После открытия такого документа значок исчезает.

Другое средство для совместной работы участников стандартной рабочей области - это вкладка **Обсуждение** (рис. 4.80). Используя кнопки панели инструментов данной вкладки, можно создавать, удалять, сортировать сообщения и др. Правила использования этого средства, способ размещения сообщений на этой вкладке аналогичны работе на форумах.

Для создания новой темы обсуждения нужно выполнить такие действия:

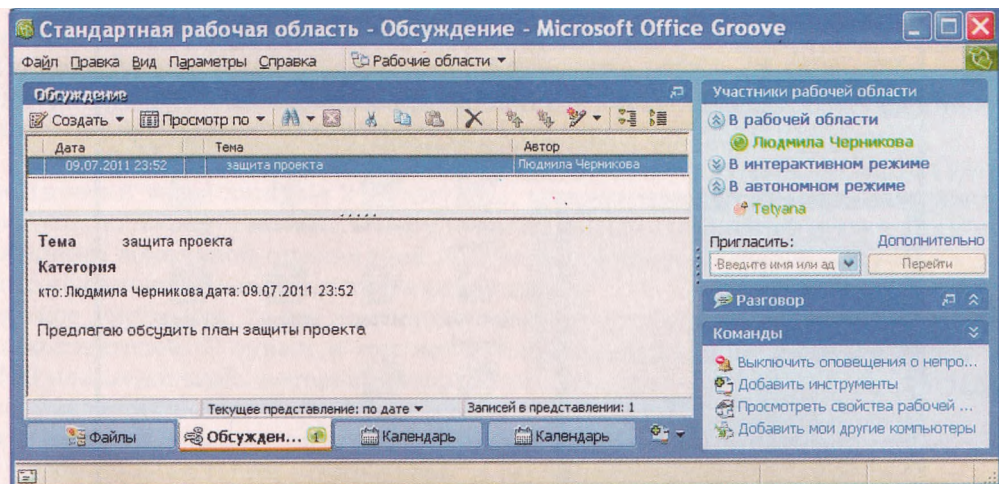
1. Открыть вкладку **Обсуждение**.
2. Выбрать кнопку **Создать**, выбрать команду **Тема**.
3. Ввести в диалоговое окно название темы для обсуждения и текст сообщения.
4. Выбрать на панели инструментов кнопку **Сохранить**.

Для написания ответа на какое-либо сообщение алгоритм действий такой:

1. Открыть нужное сообщение.
2. Выбрать кнопку **Создать**, выбрать команду **Ответ**.
3. Ввести в диалоговое окно текст ответа на сообщение.
4. Выбрать на панели инструментов кнопку **Сохранить**.

Также участники группы могут вести разговор в формате чата. Для этого предназначена панель **Разговор**, которая расположена в правой части окна

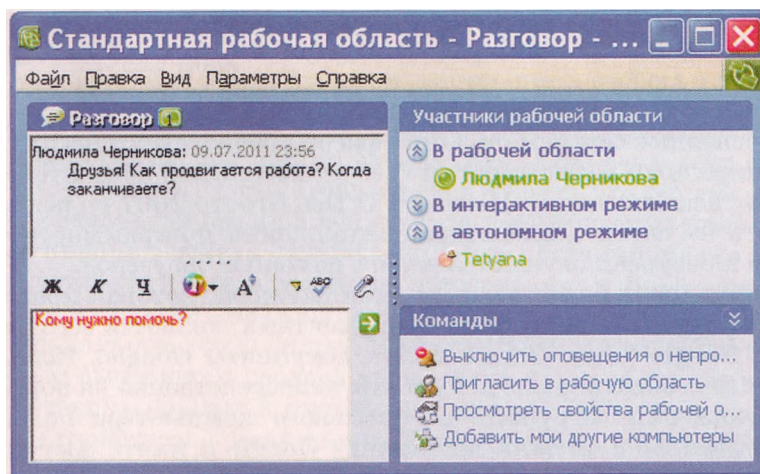


Рис. 4.80. Вкладка **Обсуждение**

рабочей области. Для развертывания чата в отдельное окно (рис. 4.81) нужно в заголовке этой области выбрать кнопку **Развернуть**. Правила работы и отображения сообщений в чате вам уже известны.

К рабочей области можно добавить и другие средства (Рисунки, Альбом, Календарь и др.), воспользовавшись для этого командой **Добавить инструменты** панели **Команды**.

Если на локальном компьютере участника группы создана рабочая область типа *Совместный доступ к файлам*, то управление такими папками осуществляется из программы **Проводник**. Программа Groove добавляет на панели инструментов окна программы **Проводник** новую кнопку **Синхронизация папки**, выбор которой открывает в окне соответствующую панель **Задачи синхронизации** (рис. 4.82). Использование этой панели дает возможность наблюдать за тем, кто из пользователей находится в рабочей области, приглашать новых пользователей, добавлять компьютеры, с которыми

Рис. 4.81. Развернутое окно **Разговор**

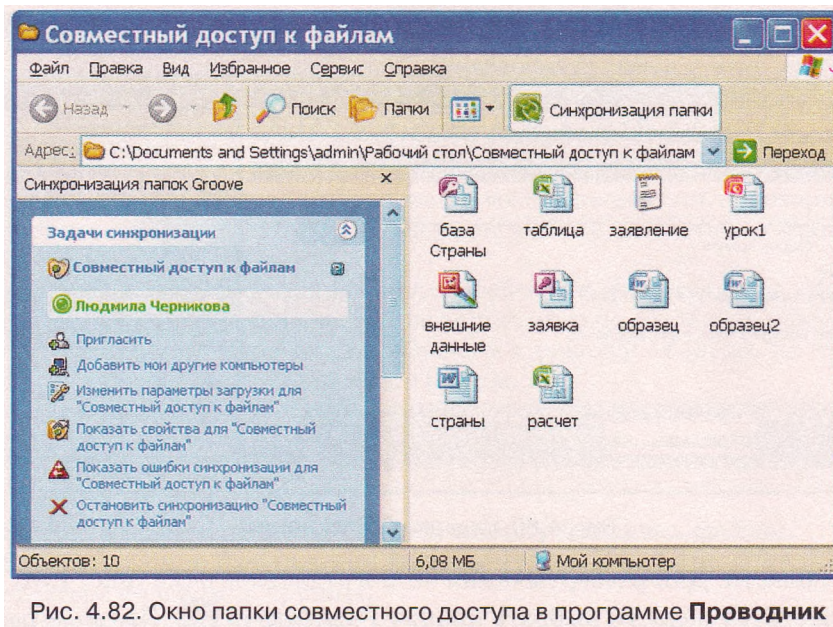


Рис. 4.82. Окно папки совместного доступа в программе **Проводник**

нужно выполнять синхронизацию, синхронизировать файлы рабочей области с файлами на компьютере и др.

## Службы онлайн-офиса

Организовать совместную работу над документами участников некоторой группы можно и с помощью он-лайн сервисов, которые получили название «он-лайн офисы». Их использование дает возможность людям со всего мира в удобное для них время работать вместе над общими документами, хранить их не на локальном компьютере, а на удаленном компьютере в сети и иметь доступ к своим материалам, где бы эти лица не находились.

Набор услуг, которые предоставляются он-лайн сервисами, обычно включает все основные средства традиционных офисных пакетов, такие как текстовый редактор, табличный процессор, программы для создания презентаций, СУБД, органайзеры, календари и т. п. Он-лайн офис может быть доступен с любого компьютера, который имеет подключение к Интернету, независимо от того, какую операционную систему он использует.

Обычно основные сервисы он-лайн офисов предлагаются бесплатно, а доступ к расширенным возможностям требует небольшой абонентской платы. В отличие от использования Microsoft Office Groove 2007 пользователю не нужно иметь на своем компьютере специального программного обеспечения. Доступ к ресурсам осуществляется с помощью браузеров.

Самой популярной бесплатной он-лайн службой, которая предоставляет сервисы для коллективной работы с документами, является сегодня служба Документы Google (англ. *Google docs* - документы Google). Используя ее, можно создавать электронные документы непосредственно на портале [www.docs.google.com](http://www.docs.google.com) или загружать с локального компьютера пользователя, хранить документы в он-лайн хранилище Google и иметь доступ к ним с любого места по сети Интернет, предоставлять при необходимости доступ к этим ресурсам другим пользователям.

Для работы с электронными документами в Документы Google используются простые программные средства: текстовый он-лайн редактор Google Documents, табличный он-лайн процессор Google Spreadsheets, программа для работы с он-лайн презентациями Google Presentations, графический он-лайн редактор Google Drawings и средство создания форм для опроса.

Он-лайн документы во время их создания и обработки сохраняются автоматически через каждые 2-3 секунды, каждая правка фиксируется, и есть возможность воспользоваться функцией отмены изменений так же, как и в обычной локальной прикладной программе. Работа пользователей происходит в реальном времени, то есть если один из пользователей изменяет содержимое документа, то все изменения сразу же отображаются и в окне других пользователей. С одним и тем же файлом одновременно могут работать до 200 пользователей, которых автор документа пригласил к совместной работе, указав их роль:

- *Соавтор* - может читать и редактировать документ, сохранять его копию на локальном компьютере, а также удалять и добавлять новых соавторов;
- *Читатель* - может просматривать содержимое последней версии документа, а также сохранять ее на своем компьютере.

На работу с электронными документами в Документы Google существует ряд ограничений, которые касаются размеров и форматов файлов документов, которые загружаются, их количества и др. Ознакомиться с этими ограничениями можно на сайте [www.docs.google.com](http://www.docs.google.com), выбрав *Загрузить => Файлы* и выбрав ссылку *Типы файлов и ограничения размера*.

## Совместная работа с он-лайн документами в Google

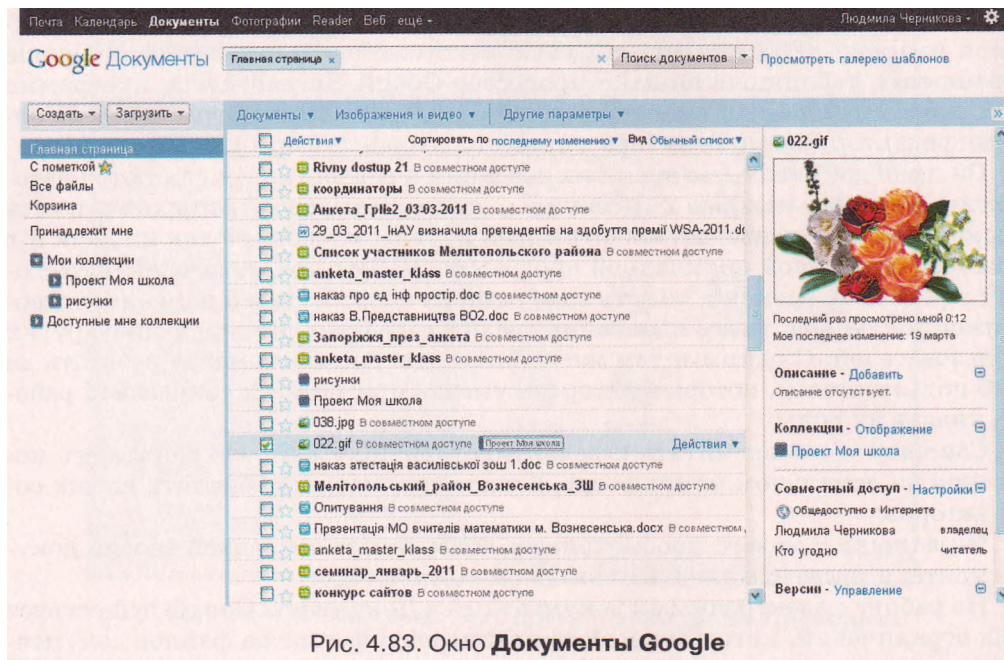
Для того чтобы использовать сервис Документы Google, нужно иметь свой Google-аккаунт. Как его создать, объяснялось в предыдущих пунктах учебника.

Чтобы получить доступ к он-лайн офису, нужно:

1. Открыть главную страницу портала Google по адресу [www.google.com](http://www.google.com).
2. Выбрать ссылку *Войти* в правом верхнем углу окна.
3. Открыть свой аккаунт, введя логин и пароль.
4. Выбрать ссылку *Еще*.
5. Выбрать в списке команду *Документы*, что приведет к открытию окна *Документы Google*.

Окно он-лайн офиса Документы Google (рис. 4.83) разделено вертикально на три части. В левой части отображается область навигации со списком доступных разделов хранилища документов. Над областью навигации размещены две кнопки *Создать* (используется для создания нового документа средствами встроенного программного обеспечения) и *Загрузить* (для импортирования в он-лайн хранилище ранее созданных файлов, которые сохранены на локальном компьютере). В центральной части окна отображается содержимое выбранного раздела хранилища. Над ним размещены кнопки для фильтрации файлов и сортировки документов по типам, приоритетам и другим свойствам - *Документы*, *Изображения* и *видео*, *Другие параметры* и др. В правой части окна отображаются значения некоторых свойств выделенных файлов и команды для их настройки.



Рис. 4.83. Окно **Документы Google**

Открыв хранилище, пользователь попадает в раздел Главная страница, где может увидеть упорядоченный список всех объектов, которые находятся в хранилище и к которым у него есть доступ. При наведении указателя мыши на любой документ возле его имени появляется кнопка Действия, выбор которой открывает список возможных операций с файлом - открыть, переименовать, настроить совместный доступ и др.

Чтобы создать новый файл средствами службы Документы Google, нужно:

1. Выбрать в области навигации левой части окна кнопку Создать.
2. Выбрать в списке тип нужного объекта — документ, презентация, таблица и т. д.
3. Создать содержание нового файла средствами он-лайн программы. Сохранять созданный файл не нужно, это будет сделано автоматически, о чем будет свидетельствовать надпись на кнопке Сохранено.

Для загрузки в он-лайн хранилище Google файла с локального компьютера пользователя нужно:

1. Выполнить Загрузить => Стандартная.
2. Выбрать на странице ссылку Выбрать файлы для загрузки.
3. Выбрать нужный файл в окне Загрузка файла.
4. Выбрать ссылку Выбрать другие файлы, если нужно загрузить несколько файлов.
5. Установить метку флажка для выбора способа конвертации файлов в формат документов Google.
6. Выбрать кнопку Целевая коллекция для выбора папки, где будет размещен файл.
7. Выбрать кнопку Частный для выбора доступа пользователей к файлу.
8. Выбрать кнопку Начать загрузку.
9. Дождаться загрузки файла и выбрать ссылку Вернуться в Документы Google.



Через некоторое время после завершения загрузки файл отобразится в списке документов.

Для упорядочения документов в хранилище пользователь может создавать новые папки (коллекции):

1. Выбрать кнопку Создать в области навигации левой части окна.
2. Выбрать в списке тип нужного объекта - Коллекция.
3. Ввести в текстовое поле имя новой папки.
4. Выбрать кнопку ОК.

Чтобы переместить объекты в некоторую папку, следует перетянуть их из центральной области окна на название нужного раздела в области навигации. Для удаления файла нужно в списке кнопки Действия выбрать команду Переместить в корзину (если вы владелец этого файла) или Удалить из моего Списка документов (если вы не являетесь владельцем этого документа).

Каждый документ, размещенный в хранилище Документы Google, можно открыть для общего чтения и редактирования несколькими пользователями. Для этого нужно:

1. Выделить файл и выбрать кнопку Действия возле документа.
2. Выбрать команду Совместный доступ => Настройка совместного доступа, что откроет соответствующее окно (рис. 4.84).
3. В области Разрешения ознакомиться с установленными по умолчанию правами доступа пользователей к файлу.
4. Выбрать при необходимости ссылку Изменить:
  - 1) Установить другие права доступа: *Общедоступно в Интернете; Всем, кому дано разрешение; Личный.*
  - 2) Установить при необходимости разрешение на редактирование файла.
  - 3) Выбрать кнопку Сохранить.
5. Ввести при необходимости в текстовое поле Добавить пользователей электронные адреса тех, кому предоставляется доступ к файлу.
6. Выбрать кнопку Закрыть.

Доступ к файлу также может быть предоставлен другим пользователям, если вы сообщите им адрес ссылки на этот файл в Интернете (отображена в верхней части окна).

Открыв общий документ, соавторы средствами предоставленного программного обеспечения могут работать с документом в режиме реального времени. Каждый может внести в документ свои комментарии, выделяя их цветом и добавляя свое имя (*Вставка => Комментарии*). При необходимости можно отменить правки, которые внесены соавторами. В процессе работы над документом можно видеть, кто из соавторов одновременно с вами редактирует данный документ. Сообщение об этом отображается над строкой меню.

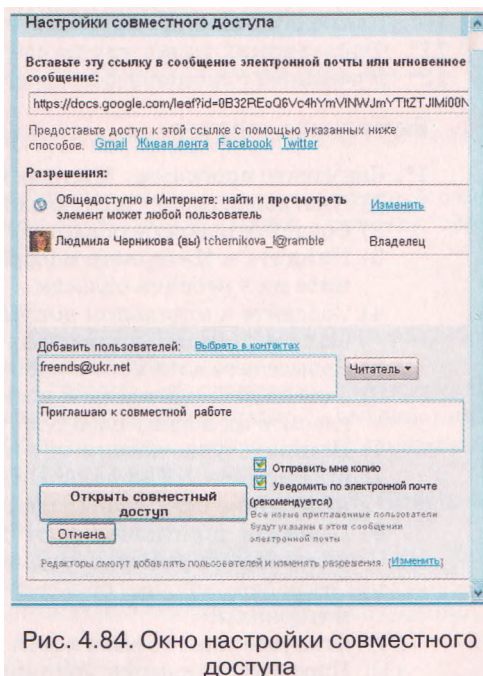


Рис. 4.84. Окно настройки совместного доступа



Для организации совместной работы над документами, виртуального офиса можно использовать и сервисы Windows Live.

Windows Live - это набор бесплатных он-лайн сервисов и локальных программ для обмена информацией, организации общения между пользователями, использования разнообразных мультимедийных данных, поиска и размещения информации в Интернет, совместной работы с документами.

Эта система насчитывает до 20 различных служб. Основной принцип, который провозглашается творцами Windows Live (корпорация Microsoft): «Все возможности Интернета собраны здесь». Одни продукты и службы Windows Live предназначены для использования в Интернете, другие - для работы на локальном компьютере, куда эти программы необходимо загрузить из Интернета.

Адрес сайта сервисов Windows Live - [www.live.com](http://www.live.com).



### Проверьте себя

- 1°. Объясните, в чем суть совместного использования документов.
- 2°. Охарактеризуйте назначения и функции программ для совместной работы над документами.
- 3°. Назовите программы, которые можно использовать для коллективной работы над проектом. Охарактеризуйте их возможности для коммуникации участников.
- 4°. Что такое виртуальный офис? Каково его назначение?
- 5°. Опишите возможности программы Microsoft Office Groove 2007 для организации совместной работы с документами.
- 6°. Опишите возможности службы Документы Google для организации совместной работы с документами.
- 7°. Объясните, чем отличается использование программы Microsoft Office Groove 2007 от среды Документы Google.
- 8°. Что такое рабочая область Microsoft Office Groove 2007? Какие типы рабочих областей можно использовать?
- 9°. Назовите роли, в которых пользователи могут получать доступ к общим документам в Microsoft Office Groove 2007. Объясните их возможности.
- 10°. Какие ограничения на файлы существуют в среде Документы Google?
- 11°. Файлы каких типов можно создавать и размещать в Документы Google?
- 12°. В чем может заключаться совместная работа над документом или проектом?



### Выполните задания

1. Запустите программу Microsoft Office Groove 2007:
  - 1) Настройте свою учетную запись.
  - 2) Создайте рабочую область Мир моих увлечений по типу Стандартная.
  - 3) Найдите в Интернете изображения по тематике вашего хобби и сохраните их в рабочей области.
  - 4) Создайте в отдельном документе список ссылок на веб-ресурсы, откуда были использованы изображения.
  - 5) Разместите документ в рабочей области.
  - 6) Установите контакты с несколькими вашими одноклассниками и пригласите их в вашу рабочую группу в роли участника.
  - 7) Пошлите участникам группы текстовое сообщение в чате с просьбой просмотреть материалы в вашей папке совместного доступа и выразить на вкладке Обсуждение свое мнение относительно вашего увлечения.
  - 8) Примите приглашение от двух ваших одноклассников относительно участия в их рабочих группах.
  - 9) Откройте рабочие области ваших одноклассников и просмотрите их материалы.
  - 10) Оставьте свои комментарии на вкладке Обсуждение.
  - 11) Просмотрите комментарии ваших одноклассников в вашей рабочей области.

- 12) Найдите среди одноклассников единомышленника по хобби.
- 13) Договоритесь с ним о создании общей презентации о вашем хобби.
- 14) Создайте презентацию из трех слайдов и сохраните ее в своей рабочей области.
- 15) Отправьте сообщение вашему соавтору с приглашением добавить к вашей презентации его слайды.
- 16) Добавьте к списку ссылок использованных ресурсов вашего соавтора свои ссылки и сохраните обновленный документ в его рабочей области.
- 17) Просмотрите вашу общую презентацию.
- 18) Отправьте сообщение с благодарностью о дружеской работе.
- 19) Закройте окно программы **Microsoft Office Groove 2007**.



2\*. Откройте свой аккаунт на сайте **Google**:

- 1) Зайдите на сервис **Документы Google**.
- 2) Создайте папку **Моя маленькая родина** и откройте к ней общий доступ всем пользователям.
- 3) Найдите в Интернете изображения интересных мест вашей местности и загрузите их в созданную папку.
- 4) Создайте в отдельном документе список ссылок на веб-ресурсы, откуда были использованы изображения, и разместите документ в общей папке.
- 5) Подготовьте три слайда презентации о вашем населенном пункте и сохраните ее в общей папке.
- 6) Отправьте участникам группы по электронной почте приглашения для совместной работы над презентацией, указав ссылку на нее в Интернете.
- 7) Примите аналогичные приглашения от двух ваших одноклассников.
- 8) Пересмотрите созданные презентации ваших одноклассников и добавьте к ним два своих слайда.
- 9) Добавляйте к списку ссылок использованных ресурсов ваших одноклассников свои ссылки.
- 10) Просмотрите общую презентацию в вашем хранилище документов.
- 11) Отправьте благодарность своим одноклассникам за дружескую работу, использовав службы мгновенного обмена сообщениями.
- 12) Закройте свой аккаунт и закройте сайт **Google**.

 **Практическая работа № 20. Разработка коллективного проекта**

*Внимание! Во время работы, с компьютером соблюдайте правила безопасности и санитарно-гигиенические нормы.*

Разработайте общий проект на одну из предложенных тем, посвященный окончанию школы: «Наша родная школа», «Наш 11-й класс», «Наши учителя», «Моя будущая профессия», «Куда пойти учиться?».

Для создания проекта:

1. Объединитесь в группу из 3-4 учеников.
2. Обсудите в группе тематику проекта, его структуру, составьте план разработки проекта, распределите роли участников проекта.
3. Подготовьте разные составляющие проекта коллективно - плакаты, буклеты, календари, текстовые документы, презентации, видеоролики, звуковые записи, публикации, стенды и др., использовав для их подготовки программное обеспечение, которое изучалось в курсе информатики.
4. Создайте средствами Microsoft Office Groove 2007 или Документы Google общий ресурс, где разместите составляющие проекта.
5. Просмотрите разработанные документы и внесите в них свои коррективы, обсудите результаты работ, спланируйте коллективную защиту проекта, используя средства коммуникаций для организации коллективной работы в группе.
6. Сохраните проект в общей папке.
7. Презентуйте свой проект своим одноклассникам.

# Содержание

Уважаемые одиннадцатиклассники! .....	3
<i>Глава 1. Информационные технологии в обучении</i>	
1.1. Программные средства учебного назначения и особенности их использования .....	4
<i>Практическая работа № 1. Использование программных средств при изучении физики, химии, биологии .....</i>	9
1.2. Математические процессоры. Программно-методический комплекс GRAN и его использование .....	10
<i>Практическая работа № 2. Автоматизация математических вычислений .....</i>	19
<i>Практическая работа № 3. Построение графиков функций и их анализ.....</i>	19
<i>Практическая работа № 4. Нахождение приближенных решений уравнений и их систем.....</i>	19
<i>Глава 2. Основы алгоритмизации и программирования</i>	
2.1. Модели и моделирование. Типы моделей .....	20
2.2. Алгоритмы и их свойства. Формы представления алгоритмов .....	25
2.3. Компьютерные программы и языки программирования. Этапы решения задач с использованием компьютера.....	31
2.4. Язык программирования Delphi и среда разработки Turbo Delphi 2006.....	40
<i>Практическая работа № 5. Создание и выполнение проекта. Использование элементов управления.....</i>	51
2.5. События и их обработчики .....	51
<i>Практическая работа № 6. Ввод и вывод данных. Работа с переменными .....</i>	62
2.6. Отладка программ .....	63
<i>Практическая работа № 7. Отладка программ.....</i>	68
2.7. Основные понятия математической логики .....	68
2.8. Ветвление в алгоритмах и программах.....	73
<i>Практическая работа № 8. Составление программ с ветвлениями .....</i>	83
2.9. Циклы в алгоритмах и в программах.....	84
<i>Практическая работа № 9. Программирование циклических вычислений .....</i>	94
2.10. Процедуры и функции пользователя и стандартные процедуры и функции в Delphi .....	95
<i>Практическая работа № 10. Использование подпрограмм .....</i>	101
2.11. Одномерные массивы .....	102
<i>Практическая работа № 11. Обработка одномерных массивов .....</i>	114
<i>Глава 3. Базы данных. Системы управления базами данных</i>	
3.1. Базы данных. Модель «сущность-связь» .....	115

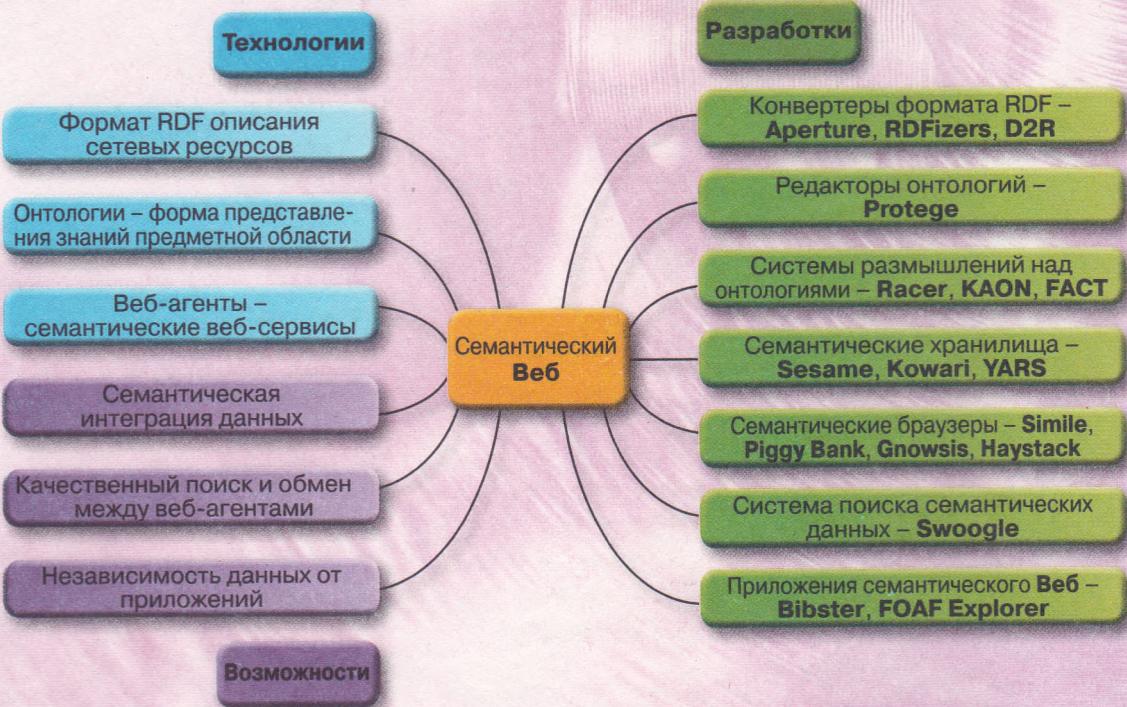


3.2. Модели данных. Основные понятия реляционной базы данных .....	121
3.3. Этапы создания базы данных. Понятие о системах управления базами данных.....	127
3.4. Система управления базами данных Microsoft Office Access 2007.....	131
3.5. Отображение модели «сущность—связь» на базу данных. Работа с таблицами базы данных.....	141
<i>Практическая работа № 12. Разработка модели «сущность—связь» заданной предметной области. Создание базы данных в среде СУБД.....</i>	152
3.6. Формы. Создание, редактирование и форматирование форм .....	153
3.7. Сортировка, поиск и фильтрация данных в базе данных .....	163
<i>Практическая работа № 13. Формы. Ввод данных. Сортировка, фильтрация и поиск данных .....</i>	168
3.8. Запросы в Access 2007.....	169
3.9. Отчеты. Создание отчетов в Access 2007.....	175
<i>Практическая работа № 14. Создание запросов и отчетов.....</i>	183
 <b>Глава 4. Информационные технологии персональной и коллективной коммуникации</b>	
4.1. Понятие о мультимедийных данных и их обработке .....	185
4.2. Добавление звуковых и видеообъектов к слайдовым презентациям .....	196
4.3. Создание и редактирование видеofilьмов (поточковых презентаций) .....	206
<i>Практическая работа № 15. Создание видеоклипа (поточковой презентации) .....</i>	217
4.4. Основы создания компьютерных публикаций. Программа Microsoft Office Publisher 2007.....	218
4.5. Особенности работы с текстовыми и графическими объектами в Publisher 2007. Создание бюллетеня и буклета.....	227
<i>Практическая работа № 16. Создание информационного бюллетеня и буклета .....</i>	234
4.6. Интегрированное использование средств обработки электронных документов.....	236
<i>Практическая работа № 17. Обработка данных в нескольких программных средах .....</i>	247
4.7. Типы веб-страниц. Классификация веб-сайтов .....	248
4.8. Этапы создания веб-сайтов. Основы веб-дизайна.....	254
4.9. Средства автоматизированной разработки веб-сайтов .....	261
<i>Практическая работа № 18. Автоматизированное создание веб-сайта .....</i>	272
4.10. Технологии и сервисы Веб 2.0. Веб-сообщества. Создание блогов .....	273
<i>Практическая работа № 19. Создание и ведение собственного блога .....</i>	289
4.11. Совместная работа над документами .....	290
<i>Практическая работа № 20. Разработка коллективного проекта .....</i>	301

## ИНТЕРНЕТ 2



## СЕМАНТИЧЕСКИЙ ВЕБ – ГЛОБАЛЬНАЯ БАЗА ЗНАНИЙ





## «ОБЛАЧНЫЕ» ВЫЧИСЛЕНИЯ



## ТЕХНОЛОГИИ БУДУЩЕГО

### Молекулярный компьютер

- в основе строения – бистабильные молекулы, могут находиться в двух стойких термодинамических состояниях
- размер молекулы – несколько нанометров
- время отклика –  $10^{-15}$  с

### ДНК-компьютер

- логические элементы – цепочки ДНК
- работоспособность в организме человека
- возможность определять заболевание и подавать лекарства к определенным органам

### Фотонный (оптический) компьютер

- все операции – манипуляция световыми потоками
- скорость выше скорости электрона
- производительность оптического процессора  $10^{13}$ – $10^{15}$  операций в секунду

### Квантовый компьютер

- основан на законах квантовой механики
- квантовые биты – кубиты – могут одновременно находиться в нескольких состояниях
- область применения – задачи перебора вариантов со многими итерациями

### Спинтронный компьютер

- спин – квантовая характеристика электрона, индекс вращательного движения элементарной частицы
- отсутствие энергозатрат на перенос частиц
- способность хранить данные при выключенном питании

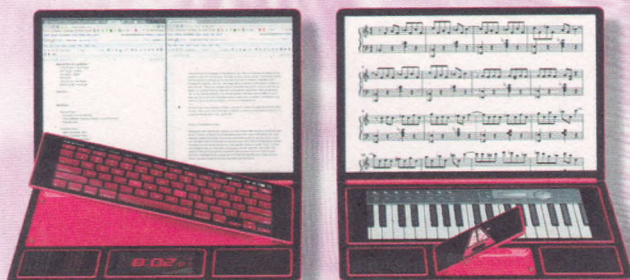
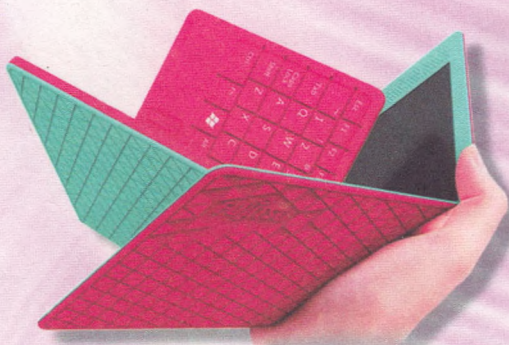


## КОМПЬЮТЕРЫ БУДУЩЕГО



**Flexbook** – раскладное сенсорное устройство – планшетный компьютер или электронная книга

Проект от компании **Dell**: компьютер с двумя пико-проекторами – для проекции изображения и для проекции клавиатуры



**Bento** – модульный «гибрид» ноутбука, планшета и смартфона

## ЭЛЕКТРОННЫЕ КНИГИ



Гибкая электронная книга от компании **Plastic Logic** из Кембриджа



Дисплей электронной книги **Reader** от компании **Skiff** воспроизводит данные на основе электронных чернил





Гибкая цветная электронная бумага от компании **Sony**



Планшетный компьютер от **Tablet XO-3** – инновационные технологии для обучения



Гибкая электронная книга **eRoll** на основе **OLED**-технологии

## НОВЕЙШИЕ УСТРОЙСТВА ВВОДА

Технология **Freespace** от компании **Hillcrest Labs**: управление курсором без контакта с поверхностью



**Deanmark AirMouse** – оптическая беспроводная мышь, роль левой и правой кнопок выполняют указательный и средний пальцы руки



Гибкая беспроводная клавиатура **Scosche freeKEY** со встроенным **Bluetooth**