

**ЭФФЕКТИВНАЯ
ПОДГОТОВКА
К ОГЭ**

ОГЭ

ИНФОРМАТИКА

УНИВЕРСАЛЬНЫЙ СПРАВОЧНИК

- Подробный теоретический материал
- Тренировочные задания



**ЭФФЕКТИВНАЯ
ПОДГОТОВКА
К ОГЭ**

ОГЭ

О.В. Дьячкова

ИНФОРМАТИКА

УНИВЕРСАЛЬНЫЙ СПРАВОЧНИК



МОСКВА
2019

УДК 373:004
ББК 32.81я721
Д93

Д93 **Дьячкова, Ольга Владимировна.**
ОГЭ. Информатика : универсальный справочник / О. В. Дьячкова. —
Москва : Эксмо, 2019. — 272 с. — (ОГЭ. Универсальный справочник).

ISBN 978-5-04-103754-3

Справочник предназначен для подготовки учащихся к ОГЭ по информатике. Пособие содержит подробный теоретический материал по всем темам, проверяемым экзаменом, а также тренировочные задания в форме ОГЭ. В конце справочника приводятся ответы.

Издание будет полезно учителям информатики, так как даёт возможность эффективно организовать учебный процесс и подготовку к экзамену.

УДК 373:004
ББК 32.81я721

ISBN 978-5-04-103754-3

© Дьячкова О.В., 2019
© Оформление. ООО «Издательство «Эксмо», 2019

СОДЕРЖАНИЕ

1. ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ

1.1. Представление информации	7
1.1.1. Информация. Язык как способ представления и передачи информации: естественные и формальные языки	7
Виды информации	9
Свойства информации	9
Языки представления информации.....	10
1.1.2. Формализация описания реальных объектов и процессов, моделирование объектов и процессов	11
1.1.3 Дискретная форма представления информации. Единицы измерения количества информации	15
Единицы измерения количества информации	15
Системы счисления.....	17
1.2. Передача информации	24
1.2.1. Процесс передачи информации, источник и приемник информации, сигнал, скорость передачи информации.....	24
1.2.2. Кодирование и декодирование информации	26
Кодирование информации	26
Декодирование информации	28
1.3. Обработка информации	30
1.3.1. Алгоритм, свойства алгоритмов, способы записи алгоритмов. Блок-схемы. Представление о программировании	30
Словесная запись алгоритма	31
Формальные исполнители алгоритма	31
Блок-схема	33
Алгоритмические языки	35
Псевдокод.....	35
Стандартная структура алгоритма	36
Описание величин и действия над ними.....	37
Команды учебного алгоритмического языка	38
1.3.2. Алгоритмические конструкции	40
Линейные алгоритмические конструкции	41
Алгоритмические конструкции ветвления.....	42
Циклические конструкции	46
Таблицы и массивы	49
1.3.3. Логические значения, операции, выражения.....	52
Алгебра логики, логические высказывания	52
Логические операции.....	54
Логические выражения.....	57
Основные законы алгебры логики	60
1.3.4. Разбиение задачи на подзадачи, вспомогательный алгоритм	64
1.3.5. Обрабатываемые объекты: цепочки символов, числа, списки, деревья.....	66
Цепочки символов	67
Списки	68
Деревья	69

1.4. Компьютер как универсальное устройство обработки информации	72
1.4.1. Основные компоненты компьютера и их функции.....	72
Периферийные устройства	74
1.4.2. Командное взаимодействие пользователя с компьютером, графический интерфейс пользователя.....	76
Окно папки.....	77
Диалоговое окно.....	78
1.4.3. Программное обеспечение, его структура. Программное обеспечение общего назначения.....	80
Системное ПО	80
Прикладное ПО.....	82
Инструментальное ПО.....	84
Тренировочные тестовые задания к разделу 1	85
Тема 1.1	85
Тема 1.2	87
Тема 1.3	89

2. ИНФОРМАЦИОННЫЕ И КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ

2.1. Основные устройства, используемые в ИКТ	102
2.1.1. Соединение блоков и устройств компьютера, других средств ИКТ; простейшие операции по управлению (включение и выключение, понимание сигналов о готовности и неполадке и т. д.); использование различных носителей информации, расходных материалов. Гигиенические, эргономические и технические условия безопасной эксплуатации средств ИКТ.....	102
Соединение блоков и устройств компьютера	102
Сигналы о готовности и неполадке компьютера	103
Начало и завершение работы с компьютером.....	103
2.1.2. Создание, именование, сохранение, удаление объектов, организация их семейств. Файлы и файловая система. Архивирование и разархивирование. Защита информации от компьютерных вирусов	105
Файлы. Файловая система.....	105
Архивирование и разархивирование	114
Защита информации от компьютерных вирусов.....	115
2.1.3. Оценка количественных параметров информационных объектов. Объем памяти, необходимый для хранения объектов.....	118
Количество информации в текстовом сообщении.....	120
Количество графической информации.....	122
Измерение объемов звуковой информации	123
2.1.4. Оценка количественных параметров информационных процессов. Скорость передачи и обработки объектов; стоимость информационных продуктов, услуг связи.....	124
2.2. Запись средствами ИКТ информации об объектах и о процессах окружающего мира	125
2.2.1. Запись изображений и звука с использованием различных устройств	125
Запись звука.....	125
Запись изображений	127
2.2.2. Запись текстовой информации с использованием различных устройств	127
Сканирование	127
Распознавание текста.....	128
2.2.3. Запись музыки с использованием различных устройств.....	130
2.2.4. Запись таблиц результатов измерений и опросов с использованием различных устройств	130

2.3. Создание и обработка информационных объектов	131
2.3.1. Создание текста посредством квалифицированного клавиатурного письма с использованием базовых средств текстовых редакторов. Работа с фрагментами текста. Страница. Абзацы, ссылки, заголовки, оглавления. Проверка правописания, словари. Включение в текст списков, таблиц, изображений, диаграмм, формул	131
Правила набора текста и составления документов	131
Запуск текстового редактора	133
Интерфейс MS Word	134
Работа с фрагментами текста	136
Форматирование текста	139
Организация списков	144
Параметры страницы	147
Стили и темы в оформлении документов Word	148
Автоматическое оглавление документа	148
Ссылки	150
Организация табличных данных	152
Вставка в текст различных объектов	156
Сервисные функции	160
2.3.2. Базы данных. Поиск данных в готовой базе. Создание записей в базе данных	161
Классификация баз данных	162
Таблицы реляционной БД	163
Связи между таблицами	165
Добавление, удаление, поиск, сортировка записей таблицы	167
Отбор записей таблицы	169
2.3.3. Рисунки и фотографии. Ввод изображений с помощью инструментов графического редактора, сканера, графического планшета, использование готовых графических объектов. Геометрические и стилиевые преобразования. Использование примитивов и шаблонов	172
Получение графических изображений	172
Создание графических изображений	174
Форматы графических файлов	178
2.4. Поиск информации	180
2.4.1. Компьютерные энциклопедии и справочники; информация в компьютерных сетях, некомпьютерных источниках информации. Компьютерные и некомпьютерные каталоги, поисковые машины, формулирование запросов	180
2.5. Проектирование и моделирование	184
2.5.1. Чертежи. Двумерная графика. Использование стандартных графических объектов и конструирование графических объектов: выделение, объединение, геометрические преобразования фрагментов и компонентов	184
Действия над графическими объектами	185
2.5.2. Диаграммы, планы, карты	188
2.5.3. Простейшие управляемые компьютерные модели	196
2.6. Математические инструменты, динамические (электронные) таблицы	197
2.6.1. Таблица как средство моделирования. Ввод данных в готовую таблицу, изменение данных, переход к графическому представлению	197
Основные элементы электронной таблицы	197
Ввод и редактирование данных	201
Вставка, удаление и очистка данных	204
Автозаполнение данных	207
Копирование и перемещение данных. Буфер обмена	209
Специальная вставка	210
Форматирование данных	211
2.6.2 Ввод математических формул и вычисления по ним	217
Простейшие расчеты	217
Адресация в формулах	219

Имена ячеек и диапазонов.....	223
Встроенные функции	223
2.6.3. Представление формульной зависимости в графическом виде.....	231
Типы диаграмм	231
Построение диаграммы	234
2.7. Организация информационной среды.....	237
2.7.1. Создание и обработка комплексных информационных объектов в виде печатного текста, веб-страницы, презентации с использованием шаблонов	237
Подготовка презентации.....	237
Начало работы с PowerPoint	238
Работа с файлами	241
Создание презентации	241
Вставка объектов на слайд	242
Оформление презентации	243
Использование анимации	244
Показ слайдов.....	247
Настройка времени презентации	247
Интерактивность презентаций	247
2.7.2. Электронная почта как средство связи; правила переписки, приложения к письмам, отправка и получение сообщения.....	248
Структура почтового ящика	248
Создание писем	249
Действия над письмами	251
2.7.3. Сохранение информационных объектов из компьютерных сетей и ссылок на них для индивидуального использования (в том числе из Интернета).....	252
Адреса веб-ресурсов	252
Работа в браузере	254
Сохранение веб-страниц	255
Закладки.....	256
2.7.4. Организация информации в среде коллективного использования информационных ресурсов. Примеры организации коллективного взаимодействия: форум, телеконференция, чат	257
Тренировочные тестовые задания к разделу 2.....	260
Тема 2.1	260
Тема 2.2	262
Тема 2.3	262
Тема 2.4	264
Тема 2.5	265
Тема 2.6	266
Тема 2.7	269
Ответы	271

1. ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ

- Знать:**
- виды информационных процессов;
 - примеры источников, приемников и каналов передачи информации;
 - принципы кодирования и декодирования информации;
 - условия возможности декодирования информации;
 - основные свойства алгоритма;
 - типы алгоритмических конструкций: следование, ветвление, цикл;
 - понятие вспомогательного алгоритма;
 - основные элементы математической логики;
 - различные формы представления упорядоченной информации;
 - основные характеристики компьютера;
 - функциональная схема устройства компьютера;
 - основные характеристики операционной системы;
 - способы организации пользовательского интерфейса;
 - общие характеристики программных средств.
- Уметь:**
- выделять основные информационные процессы в реальных системах;
 - оценивать числовые параметры информационных процессов;
 - выполнять равномерное и неравномерное кодирование, в том числе кодирование информационных цепочек;
 - выполнять и строить простые алгоритмы;
 - составлять блок-схемы решения задач;
 - создавать и преобразовывать логические выражения;
 - разбивать процесс решения задачи на этапы;
 - выполнять базовые операции над объектами: цепочками символов, числами, списками, деревьями;
 - оперировать компьютерными информационными объектами в наглядно-графической форме;
 - выбирать программные средства, предназначенные для работы с информацией определенного вида.

1.1. ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ

1.1.1. Информация. Язык как способ представления и передачи информации: естественные и формальные языки

Информация — одно из основных, фундаментальных понятий мира, наряду с такими основополагающими понятиями науки, как материя и энергия.

Понятие «информация» (от *лат.* *informatio* — разъяснение, изложение, осведомление; *informare* — придавать форму) вошло в широкое употребление в середине XX века. Несмотря на всю его важность, строгого определения информации, пригодного для всех областей науки, сегодня не существует.

В обыденной жизни под **информацией** понимают сведения о предметах, явлениях, фактах, действиях, процессах, передаваемые людьми устным, письменным или другим образом. Иными словами, это сведения об окружающем мире и процессах, протекающих в нем.

Сведения о внешнем мире человек воспринимает с помощью органов чувств (зрения, слуха, вкуса, обоняния, осязания). Таким образом, **информация** — это знания, сведения, которые человек получает из окружающего мира с помощью органов чувств. Такой подход к понятию «информация» можно назвать *субъективным*.

Однако понятие «информация» намного шире обыденного понимания. Так, с позиции *кибернетики** **информация** — это совокупность сигналов, воздействий или сведений, которые система или объект воспринимает извне (*входная информация*), выдает в окружающую среду (*выходная информация*) или хранит в себе (*внутренняя информация*).

Такое определение позволяет рассматривать с единой точки зрения самые разные процессы: передачу сообщений по техническим каналам связи, деятельность нервной системы человека и животных, работу вычислительных машин, различные процессы управления и т. д. Для кибернетического подхода важно, что информация используется для активного действия, управления, т. е. в целях сохранения, совершенствования, развития системы.

Например, система управления автомобилем реагирует на нажатие педали газа водителем и регулирует подачу топлива в двигатель. Термостат воспринимает изменение температуры в помещении и в соответствии с заданным режимом включает или отключает отопительные приборы. Нейроны человека в случае, например, ожога или укола передают нервные импульсы к мышцам, что вызывает их быстрое сокращение. На основе генетической информации, заложенной в луковице тюльпана, из нее вырастает цветок определенного цвета.

В *теории информации*** под информацией понимается не каждое сообщение, а лишь такое, которое содержит неизвестные для получателя факты и дополняет его представление об объекте или процессе. С точки зрения такого *вероятностного подхода* **информация** — это сведения об объектах и явлениях окружающей среды, которые уменьшают имеющуюся о них степень неопределенности и неполноты знаний. При таком подходе, например, формулировка теоремы Пифагора или название столицы Франции не являются информацией, если эти сведения уже известны получателю.

Функциональный подход определяет **информацию** как форму отражения и часто связывает ее с живыми системами. Отражение понимается как результат некоторого воздействия. Некоторый объект считается носителем информации о другом объекте (или внешнем мире), если в нем происходят изменения, отражающие воздействия на него этого второго объекта. Например, проводник нагревается и удлиняется, если он включен в электрическую цепь; металлы, находящиеся на воздухе, окисляются. Тем самым эти объекты передают информацию о внешней среде. Примерами отражения могут служить танец пчел, со-

* *Кибернетика*, по определению ее основоположника Норберта Винера, — наука о связи и управлении в машинах и животных, а также в обществе и человеческих существах.

** *Теория информации* — математическая теория систем связи и передачи информации. Ее создателем, так же как и основоположником цифровой связи, считается Клод Шеннон. Именно он впервые обосновал возможность применения двоичного кода для передачи информации.

общающих о найденном источнике пищи; распускание почек на растениях при весеннем потеплении; дрожь человека как реакция на холод или стресс; рельеф Земли как результат тектонических процессов в ее недрах.

В *компьютерной обработке данных* под **информацией** понимают последовательность символьных обозначений (букв, цифр, закодированных графических образов и звуков и т. п.), несущую смысловую нагрузку и представленную в доступном компьютеру виде.

Как правило, наиболее часто в *информатике* используется следующее определение.

Информация — это сведения об объектах и явлениях окружающей среды, которые являются объектом хранения, преобразования, передачи и использования.

Науку, изучающую структуру и общие свойства информации, а также методы ее представления, накопления, хранения, поиска, обработки, передачи и воспроизведения с помощью технических средств, называют **информатикой**.

Виды информации

Одно и то же информационное сообщение может быть представлено знаками, записанными на бумаге или высеченными в камне; звуковыми волнами при устном сообщении; нервными импульсами в процессе говорения; электрическими импульсами при телефонной передаче или телевизионном показе и многими другими способами.

Таким образом, информация различается по *форме представления*. Технические системы обрабатывают информацию в *сигнальном* представлении. Человек воспринимает информацию также в виде *знаков и образов*. Это могут быть числа, тексты, графическое представление (таблицы, рисунки, чертежи, схемы, фотографии), видеoinформация и др. Информация может быть представлена и звуковыми сигналами (устная речь, музыка), жестами, запахами, вкусами, световыми сигналами, электрическими и нервными импульсами, перепадами давления или температуры и др.

Соответственно, и *носителями информации* могут быть разные объекты. Среди них — материальные предметы (бумага, камень и т. д.); волны различной природы (звуковые, электромагнитные (световые, радиоволны), гравитационные); состояние вещества (давление, концентрация, температура).

По *способу восприятия человеком* различают информацию *визуальную* (зрительную), *аудиальную* (звуковую), *вкусовую*, *обонятельную*, *тактильную*. Значительную часть информации человек получает благодаря зрительному восприятию, меньшую — благодаря слуховому, остальные способы получения информации для человека менее важны. Поэтому в целом человек характеризуется *аудиовизуальным* восприятием информации.

Свойства информации

- **Объективность.** Информация не зависит от методов ее фиксации, чьего-либо мнения, суждения. Например, сообщение «до озера недалеко» несет субъективную информацию (зависящую от человека), а сообщение «до озера 4 км» —

объективную. Объективную информацию можно получить, например, с помощью различных измерительных приборов.

- **Достоверность.** Информация соответствует реальности, истинному состоянию. Недостоверной информация может стать в результате неточных измерений, помех либо в случае сознательного искажения. Намеренно искаженная информация называется *дезинформацией*.
- **Полнота.** Информация достаточна для понимания ситуации и принятия решения. Неполная информация может привести к неправильной оценке ситуации и ошибочным решениям.
- **Актуальность.** Информация важна, насущна именно в настоящий момент.
- **Ценность (полезность, значимость).** Информация значима для решения поставленной задачи. Избыточная, лишняя информация называется *шумом*. Разница между полезной информацией и шумом условна — в разное время, при разных обстоятельствах и для разных потребителей одна и та же информация может быть ценной или лишней.
- **Понятность (ясность).** Информация выражена способом, доступным получателю, и обеспечивает ясное и однозначное понимание.

Языки представления информации

Информацию можно представлять с помощью знаков. **Знаковые системы** — это наборы знаков определенного типа. Примерами знаковых систем являются разговорные языки, системы счисления, нотная грамота, математические формулы.

Каждая знаковая система строится на основе определенного **алфавита** — некоторого конечного упорядоченного набора знаков (символов или сигналов). Полное число символов алфавита называют **мощностью алфавита**. В знаковую систему включаются также и **правила выполнения операций** над знаками алфавита.

Язык — определенная знаковая система представления информации. Существуют *естественные* и *формальные* языки.

К **естественным языкам** относятся разговорные языки в устной и письменной форме, язык мимики и жестов и др. Разговорные языки начали формироваться еще в древнейшие времена для обмена информацией между людьми. На сегодняшний день существует несколько тысяч естественных языков — например, русский, английский, арабский, китайский и др.

В *устной речи* в качестве знаков языка используются различные звуки (фонемы). В основе *письменной речи* лежит алфавит — набор знаков (букв или иероглифов), которые человек различает по их начертанию. Алфавит русского языка называется кириллицей и содержит 33 знака, английский язык использует латиницу и содержит 26 знаков.

Из символов алфавита по правилам *грамматики* составляются *слова*, а из них по правилам *синтаксиса* — *предложения*.

Формальные языки — это специальные языки для различных областей человеческой деятельности. Они имеют жестко фиксированный алфавит и строгие правила грамматики и синтаксиса. Примерами формальных языков служат языки программирования, системы счисления, алгебра и другие языки математики, нотная запись, язык дорожных знаков.

Например, десятиричная система счисления — это знаковая система, в качестве алфавита которой используются арабские цифры, а выполнение арифметических операций над ними задается строгими правилами. Азбука Морзе представляет собой алфавит из двух знаков (точки и тире) и правила составления сигналов из этих знаков.

1.1.2. Формализация описания реальных объектов и процессов, моделирование объектов и процессов

Модель — искусственно созданный объект, который замещает исследуемый объект и отображает в более простом, уменьшенном виде структуру, свойства, взаимосвязи и отношения между его элементами.

Для каждой модели существует ее **прототип**, или **оригинал** — тот объект, который она замещает. Процесс создания модели называется **моделированием**. В процессе моделирования выделяются главные, наиболее существенные, свойства объекта.

Моделирование ставит целью понять сущность объекта, научиться им управлять, прогнозировать его состояние или действия.

Моделировать можно существующие предметы, явления, процессы, а также не существующие: объекты, которые планируется разработать, явления, которые могут и не произойти, и т. д.

По назначению различают модели научно-технические, исследовательские, обучающие, имитационные и др.

Научно-технические модели позволяют исследовать явления и процессы в лабораторных, а не в реальных условиях. **Исследовательские модели** дают возможность изучить потенциальные свойства или характеристики сооружений и механизмов до их воплощения в жизнь, чтобы избежать возможных ошибок. **Обучающие модели** и тренажеры используются для изучения или демонстрации свойств каких-либо объектов, процессов или явлений. **Имитационные модели** позволяют заменить (сымитировать) исследуемый объект другим со схожими свойствами.

Для исследования одного и того же объекта могут использоваться разные модели. Для исследования разных объектов может использоваться одна и та же модель.

По способу реализации модели подразделяют на *материальные* и *информационные*.

Материальные модели имеют реальное воплощение: макеты, копии, образцы.

Информационные модели представляют совокупность информации, характеризующей свойства и состояние объекта и его взаимосвязи с внешним миром. Примерами информационных моделей служат качественные описания, схемы и чертежи, таблицы и рисунки, химические формулы и географические карты, диаграммы и планы и т. д.

Информационные модели делятся на *описательные* (созданные на естественном языке) и *знаковые* (использующие формальный язык). Процесс построения информационных моделей с помощью формальных языков называют *формализацией*.

Этапы разработки формальной информационной модели:

1. Анализ исследуемого объекта и его свойств, выделение существенных свойств с точки зрения моделирования.
2. Выбор формы представления модели.
3. Формализация.
4. Анализ модели на непротиворечивость.
5. Анализ адекватности (соответствия) модели целям и задачам моделирования.

Математические модели — информационные модели в виде совокупности математических формул, отражающих взаимозависимости между параметрами объекта.

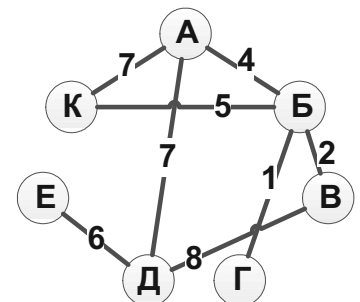
Расчеты для многих математических моделей проводят с помощью компьютеров. С помощью специальных программ исследуют объекты, которые невозможно, опасно либо дорого исследовать напрямую; процессы, которые происходят слишком медленно (или слишком быстро); явления, которые ранее не происходили, и т. д.

Пример 1

В таблице приведена протяженность дорог между населенными пунктами (отсутствие числа означает, что между соответствующими пунктами нет прямого сообщения). Определить длину кратчайшего пути из пункта Г в пункт Д (при условии, что можно передвигаться только по имеющимся дорогам).

	А	Б	В	Г	Д	Е	К
А		4			7		7
Б	4		2	1			5
В		2			8		
Г		1					
Д	7		8			6	
Е					6		
К	7	5					

Решение. Преобразуем форму представления исходных данных из табличной в графическую. Изобразим схему сообщения между населенными пунктами и нанесем протяженности путей сообщения.



Выпишем все возможные варианты перемещения из пункта Г в пункт Д и рассчитаем их протяженность:

$$\text{ГБАД} = 1 + 4 + 7 = 12$$

$$\text{ГВВД} = 1 + 2 + 8 = 11$$

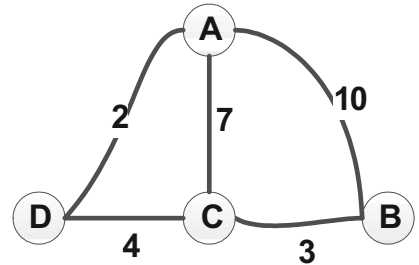
$$\text{ГБКАД} = 1 + 5 + 7 + 7 = 20$$

Таким образом, кратчайший путь ГВВД имеет длину 11.

Ответ: 11.

Пример 2

На схеме нарисованы дороги между четырьмя населенными пунктами и указана их протяженность. Определите, какие два пункта наиболее удалены друг от друга, указать кратчайшее расстояние между этими пунктами.



Решение. Выпишем все варианты сообщения между населенными пунктами. Для того чтобы не рассматривать симметричные варианты вида АСВ и ВСА, сведем все записи в таблицу и заполним в ней только один треугольник (правый верхний либо левый нижний).

	A	B	C	D
A		AB (10) ACB (10) ADCВ (9)	AC (7) ABC (13) ADC (6)	AD (2) ACD (11) ABCD (17)
B			BC (3) BAC (17) BADC (16)	BCD (7) BAD (12)
C				CD (4) CAD (9) CBAD (15)
D				

Выберем из имеющихся вариантов минимальные:

	A	B	C	D
A		ADCВ (9)	ADC (6)	AD (2)
B			BC (3)	BCD (7)
C				CD (4)
D				

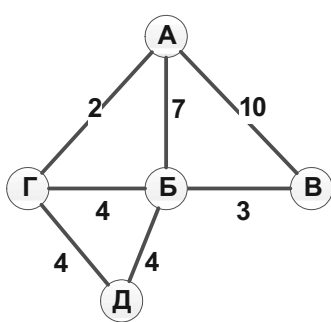
Осталось определить наибольшее расстояние из найденных — 9 для пути ADCB.

Ответ: 9.

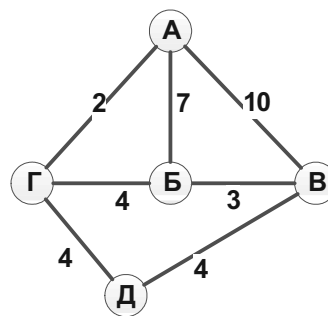
Пример 3

В таблице приведена протяженность дорог между пятью населенными пунктами. Указать схему, соответствующую таблице.

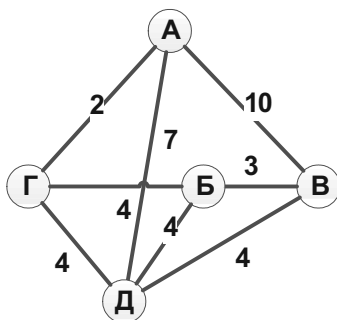
	А	Б	В	Г	Д
А		7	10	2	
Б	7		3	4	
В	10	3			4
Г	2	4			4
Д			4	4	



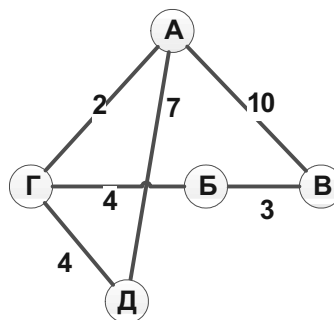
1)



2)



3)



4)

Решение. Все схемы отличаются друг от друга только наличием или отсутствием сообщений АБ, АД, БД, ВД. Остальные дороги одинаковы. Поэтому в первую очередь будем проверять совпадение данных для указанных дорог.

Схема 1 содержит путь БД, который не задан в таблице, также на схеме отсутствует дорога ВД (которая в таблице отмечена). Следовательно, схема 1 не отвечает условию задачи. На схемах 3 и 4 имеется маршрут АД, которого нет в таблице. Проверяем схему 2: на ней отмечены дороги АВ (протяженностью 7), АВ (10), АГ (2), БВ (3), БГ (4), ВД (4) и ГД (4). Все эти сведения полностью совпадают с табличными данными. Схема 2 полностью соответствует условию задачи.

Ответ: 2.

1.1.3. Дискретная форма представления информации.

Единицы измерения количества информации

Информация может быть представлена в *аналоговой* или *дискретной* форме.

Величина в **аналоговой форме** может принимать бесконечное множество значений. Примерами аналогового представления информации могут служить звук скрипки, картина художника, показатели температуры воздуха, уровня воды в реке.

Величина в **дискретной форме** может принимать только конечное множество значений. Примеры дискретного представления информации: цифровые показания часов или спидометра, текст в книге, изображение на экране монитора.

Величину в аналоговой форме представления информации можно преобразовать в величину в дискретной форме. Этот процесс называется **дискретизацией**.

Представление информации в компьютере дискретно. В процессах хранения, обработки и передачи информации в компьютере используется *двоичная знаковая система*. Ее алфавит состоит всего из двух знаков $\{0, 1\}$. Для удобства использования такого алфавита договорились называть любой из его знаков **бит** (от *англ.* bit — binary digit — двоичный знак). Поскольку один бит может принимать только одно из двух значений, то им выражают одно из двух взаимоисключающих понятий: да/нет, истина/ложь, включено/выключено.

Способ представления информации с помощью кода из двух знаков оказался наиболее значимым для развития техники. Двоичные числа удобно хранить, обрабатывать и передавать с помощью электронных устройств. Основным носителем информации в них являются элементы, которые могут находиться в одном из двух состояний: включено/выключено, высокий/низкий уровень напряжения или тока, наличие/отсутствие намагниченности материалов. Условно одно состояние обозначают через 1, а другое через 0. Каждый такой элемент способен хранить один двоичный разряд, или бит информации.

Любое информационное сообщение представляется последовательностью нулей и единиц (*цифрового кода*). Этот метод представления информации называется *двоичным кодированием*. Таким образом, двоичный код является универсальным средством кодирования информации. Благодаря двоичному кодированию все действия по обработке сообщений компьютером сводятся к совокупности простых действий над 0 и 1.

Единицы измерения количества информации

Бит — единица измерения количества информации, равная одному разряду в двоичной системе счисления. Это наименьшая единица измерения информации.

Основной единицей хранения и обработки цифровой информации принят *байт*.

Байт (англ. byte) — совокупность восьми двоичных разрядов (битов).

Соответственно, с помощью одного байта можно получить 256 ($= 2^8$) двоичных значений (от 00000000 до 11111111). В современных персональных компьютерах байт является наименьшей совокупностью битов, которую компьютер обрабатывает одномоментно.

На практике применяют более емкие, чем байт, единицы измерения объема сообщений и емкости носителей — *килобайты*, *мегабайты*, *гигабайты*, *терабайты*. Множителем при переходе к более емкой единице измерения выступает число 1024 ($= 2^{10}$).

Единицы измерения емкости носителей

Название	Условное обозначение	Соотношение с другими единицами
бит	бит	
килобит	Кбит	2^{10} бит = 1024 бит
мегабит	Мбит	2^{20} бит = 1024 Кбит
гигабит	Гбит	2^{30} бит = 1024 Мбит
терабит	Тбит	2^{40} бит = 1024 Гбит
байт	байт (Б)	8 бит
килобайт	Кбайт (Кб)	2^{10} байт = 1024 байт
мегабайт	Мбайт (Мб)	2^{20} байт = 1024 Кбайт
гигабайт	Гбайт (Гб)	2^{30} байт = 1024 Мбайт
терабайт	Тбайт (Тб)	2^{40} байт = 1024 Гбайт

Пример 1

Отчет включает 12 печатных страниц, каждая из которых содержит 40 строк текста по 60 символов. Для сохранения отчета была применена одна из кодировок Unicode, в которой каждый символ кодируется 16 битами. Определить информационный объем отчета.

Решение. Отчет содержит $12 \times 40 \times 60 = 28800$ символов. Для кодирования каждого из них потребуется 16 бит = 2 байта, следовательно, информационный объем отчета составляет 28800×2 байта = 57600 байт = 56,25 Кбайт.

Ответ: 56,25 Кбайт.

Пример 2

Расположить по порядку возрастания величины объемов памяти:
1000 байт; 12 Кбайт; 120 бит; 20 байт; 10 бит.

Решение. Сведем все величины к одной единице измерения — например, к битам.

$$1000 \text{ байт} = 1000 \times 8 \text{ бит}$$

$$12 \text{ Кбайт} = 12 \times 1024 \times 8 \text{ бит} = 12288 \times 8 \text{ бит}$$

$$120 \text{ бит} = 15 \times 8 \text{ бит}$$

20 байт = 20×8 бит

Таким образом, в порядке возрастания будут располагаться:

10 бит; 15×8 бит; 20×8 бит; 1000×8 бит; 12288×8 бит.

Или: 10 бит; 120 бит; 20 байт; 1000 байт; 12 Кбайт.

Ответ: 10 бит; 120 бит; 20 байт; 1000 байт; 12 Кбайт.

Системы счисления

Система счисления — совокупность обозначений, приемов и правил для записи чисел цифровыми знаками.

В зависимости от способов изображения чисел цифрами системы счисления делятся на *непозиционные* и *позиционные*.

Непозиционные системы счисления — такие, в которых количественное значение каждой цифры не зависит от занимаемой ею позиции в изображении числа.

Примером может служить *египетская система счисления* — в ней иероглифы (цифры), составляющие число, можно записывать сверху вниз, справа налево или вперемежку. Значение числа равно сумме значений цифр в его записи.

Переходной от непозиционных систем к позиционным служит *римская система счисления*. В ней позиция некоторых цифр уже меняет значение числа: например, в числе IX единицу нужно отнять от десяти, а в числе XI единицу нужно прибавить к десяти. Однако количественное значение самих цифр X и I от их позиции не зависит.

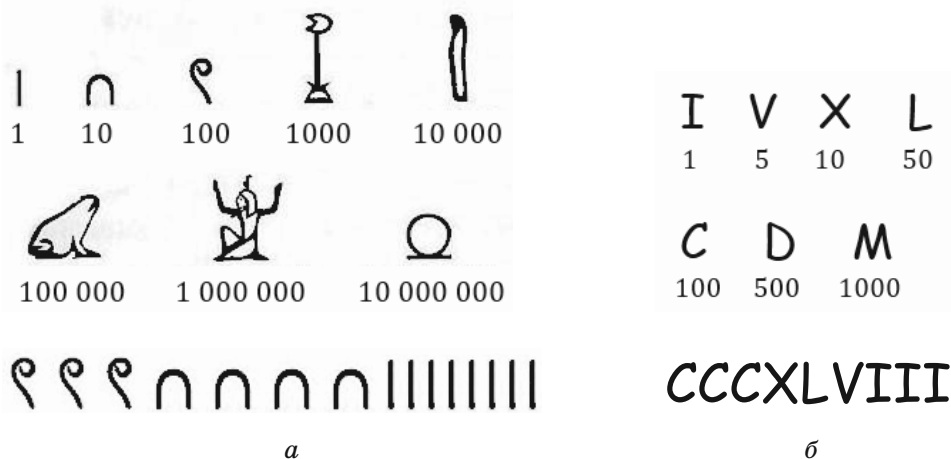


Рис. 1.1. Алфавит и изображение числа 348 в египетской (а) и римской (б) системах счисления

В римской системе цифры записываются слева направо в порядке убывания, и тогда их значения складываются. Если слева записана меньшая цифра, а справа — большая, то их значения вычитаются. Нежелательно записывать более трех одинаковых цифр подряд.

Например, для представления числа 348 в римской системе счисления надо выписать сначала число сотен, затем десятков и единиц: 300 — CCC, 40 — XL, 8 — VIII. Затем соединить эти записи: CCCXLVIII.

Аналогично для числа 1977: 1 тысяча — М, 900 — CM, 70 — LXX, 7 — VII.
Результат: MCMLXXVII.

Пример 3

Вычислить значение числа MMCDXXXIV, представленного в римской системе счисления.

Решение. Найдем позиции, в которых цифры записаны не по убыванию их значений: MM(CD)XXX(IV). Для расчета соответствующие значения нужно будет вычитать (из большего меньшее). Остальные значения нужно сложить:

$$1000 + 1000 + (500 - 100) + 10 + 10 + 10 + (5 - 1) = 2434.$$

Ответ: 2434.

В непозиционных системах очень трудно производить многие действия над числами, особенно умножение и деление, слишком громоздка запись для больших чисел. Поэтому широкое распространение получили позиционные системы счисления.

Позиционные системы счисления — такие, в которых количественное значение каждой цифры зависит от ее позиции в числе.

Количество знаков (цифр), используемых для изображения числа, называется **основанием** системы счисления (или *мощностью алфавита*). Систему с основанием 10 называют десятичной, с основанием 2 — двоичной, с основанием 16 — шестнадцатеричной, в общем случае: с основанием k — k -ичной.

Место цифры в числе называется **разрядом**, а количество цифр в числе — его **разрядностью**. Разряды целого числа нумеруются справа налево начиная с 0. Дробные разряды нумеруют слева направо начиная с -1 .

Примером позиционной системы счисления является используемая нами арабская *десятичная система счисления*. Иногда ее называют индо-арабской, поскольку она была придумана в Индии, а стала известна в Европе из арабских трактатов. Алфавит этой системы составляют 10 цифр — от 0 до 9. Каждая цифра в числе при перемещении справа налево в следующий разряд увеличивает свое значение в 10 раз. Чтобы определить значение числа, надо сложить произведения каждой его цифры на 10 в степени, равной разряду этого числа.

$$\begin{aligned} 348 &= 3 \cdot 10^2 + 4 \cdot 10^1 + 8 \cdot 10^0 \\ -348,17 &= -(3 \cdot 10^2 + 4 \cdot 10^1 + 8 \cdot 10^0 + 1 \cdot 10^{-1} + 7 \cdot 10^{-2}) \end{aligned}$$

Системы счисления могут иметь различные основания. Чтобы различать, в какой системе счисления записано число, принято указывать ее основание в виде *нижнего индекса* справа от числа. Сам индекс всегда представляется в десятичной системе. Для самой десятичной системы индекс указывают только тогда, когда используется какая-либо другая система:

316 — число в десятичной системе счисления,
316₈ — число в восьмеричной системе счисления.

Свойства записи чисел в позиционной системе счисления:

1. Для записи чисел в позиционной системе счисления с основанием k требуется k знаков (алфавит системы состоит из k цифр или букв).

2. Основание системы счисления, записанное в ней, всегда имеет вид 10 (читается «один ноль»).

3. С помощью n разрядов в позиционной системе счисления с основанием k могут быть записаны k^n чисел (от 0 до $k^n - 1$).

Если основание системы k больше 10, то цифры старше 10 при записи обозначают прописными буквами латинского алфавита: А, В, ..., Z. При этом цифре 10 соответствует знак А, цифре 11 — знак В и т. д.

Цифры шестнадцатеричной системы счисления

16-ричная система	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Соответствие в десятичной системе	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Информация в компьютере представлена в цифровой двоичной форме. В целях экономичного отображения двоичную информацию можно представлять в шестнадцатеричном виде. В программировании часто используется восьмеричная запись чисел.

Характеристика некоторых систем счисления

Основание системы счисления	Название системы счисления	Алфавит системы счисления
2	двоичная	0 1
3	троичная	0 1 2
8	восьмеричная	0 1 2 3 4 5 6 7
10	десятичная	0 1 2 3 4 5 6 7 8 9
16	шестнадцатеричная	0 1 2 3 4 5 6 7 8 9 A B C D E F

В общем виде число в позиционной системе счисления может быть представлено как последовательность символов алфавита (цифр), обозначенных через a_1, a_2, a_3 и т. д. Для числа А с количеством целых разрядов n и количеством дробных разрядов m запись имеет вид:

$$A = a_{n-1} a_{n-2} \dots a_2 a_1 a_0 a_{-1} a_{-2} \dots a_{-m}$$

Такая запись называется **свернутой записью** числа. Эту форму записи мы используем в повседневной жизни, поэтому ее называют также *естественной*.

Представление числа в виде многочлена называют **развернутой записью** числа:

$$A = a_{n-1} \cdot k^{n-1} + a_{n-2} \cdot k^{n-2} + \dots + a_1 \cdot k^1 + a_0 \cdot k^0 + a_{-1} \cdot k^{-1} + \dots + a_{-m} \cdot k^{-m}.$$

Развернутая запись числа задает правило для вычисления числа по его цифрам в k -ичной системе счисления. Для уменьшения количества вычислений пользуются *схемой Горнера*, которая получается путем поочередного выноса основания системы k за скобки:

$$A = (\dots((a_{n-1} \cdot k + a_{n-2}) \cdot k + a_{n-3}) \cdot k + \dots + a_1) \cdot k + a_0.$$

Пример 4

Сформировать развернутую запись числа 1234,5 в десятичной системе счисления и выполнить вычисления в соответствии с этой записью.

Решение. Перенумеруем разряды в целой части числа справа налево начиная с 0 (от 0 до 4). Дробный разряд в заданном числе всего один. Выпишем в соответствии с нумерацией разрядов сумму произведений значений цифр на степени разрядов. Для дробного разряда используем степень -1 :

$$1234,5 = 1 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0 + 5 \cdot 10^{-1}.$$

Рассчитаем значение полученного многочлена:
 $1000 + 200 + 30 + 4 + 0,5 = 1234,5$.

ПЕРЕВОД ЦЕЛЫХ ЧИСЕЛ ИЗ ОДНОЙ СИСТЕМЫ СЧИСЛЕНИЯ В ДРУГУЮ

Любое число может быть переведено из одной системы счисления в другую. Целую и дробную части числа переводят по разным правилам. Рассмотрим 3 варианта перевода целых чисел:

- из десятичной системы в десятичную;
- из десятичной системы в десятичную;
- из десятичной системы в другую десятичную.

ПЕРЕВОД ЦЕЛЫХ ЧИСЕЛ ИЗ ДЕСЯТИЧНОЙ СИСТЕМЫ В ДЕСЯТИЧНУЮ

Для перевода целых чисел из десятичной системы в десятичную следует:

- 1) превратить свернутую форму записи числа в развернутую;
- 2) выполнить арифметические действия развернутой записи.

Пример 5

Перевести двоичное число 101101 в десятичную систему счисления.

Решение. Для записи развернутой формы числа сначала перенумеруем все разряды числа справа налево начиная с 0:

$$\begin{array}{cccccc} 1 & 0 & 1 & 1 & 0 & 1 \\ 5 & 4 & 3 & 2 & 1 & 0 \end{array}$$

Представим двоичное число в развернутой форме. Для этого выпишем сумму произведений каждой цифры на основание системы (число 2) в степени ее разряда. Для удобства можно подписать основание 2 под каждым номером разряда.

$$\begin{array}{cccccc} 1 & 0 & 1 & 1 & 0 & 1_2 \\ 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{array} = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

Выполним арифметические действия в десятичной системе (учитывая, что $2^0 = 1$).

$$\begin{array}{cccccc} 1 & 0 & 1 & 1 & 0 & 1_2 \\ 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{array} = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = \\ = 32 + 0 + 8 + 4 + 0 + 1 = 45_{10}$$

Ответ: 45.

Пример 6

Перевести число $1A3_{16}$ в десятичную систему счисления.

Решение. Перенумеруем разряды справа налево и выпишем развернутую форму числа (учитывая, что основание системы равно 16). При этом заменим шестнадцатеричную цифру А на соответствующее ей в десятичной системе значение 10.

$$\begin{array}{ccc} 1 & A & 3 \\ 2 & 1 & 0 \end{array}$$

$$\begin{array}{r} 1 \quad \text{A} \quad 3_{16} \\ 16^2 \quad 16^1 \quad 16^0 \end{array} = 1 \cdot 16^2 + 10 \cdot 16^1 + 3 \cdot 16^0 = 256 + 160 + 48 = 464$$

Ответ: 464.

ПЕРЕВОД ЦЕЛЫХ ЧИСЕЛ ИЗ ДЕСЯТИЧНОЙ СИСТЕМЫ В НЕДЕСЯТИЧНУЮ

Для перевода целого числа из десятичной системы в недесятичную следует:

- 1) разделить это число на основание новой системы счисления с остатком до получения нулевого результата;
- 2) выписать остатки в обратной последовательности.

► Пример 7

Перевести число 29_{10} в двоичную систему счисления.

Решение. Разделим число 29 на 2 в столбик с остатком.

$$\begin{array}{r} \underline{29} \quad | \quad 2 \\ \underline{28} \quad 14 \\ \mathbf{1} \end{array}$$

Получившийся результат вновь разделим на 2 с остатком:

$$\begin{array}{r} \underline{29} \quad | \quad 2 \\ \underline{28} \quad \underline{14} \quad | \quad 2 \\ \mathbf{1} \quad \underline{14} \quad 7 \\ \mathbf{0} \end{array}$$

Продолжим деление до тех пор, пока не получим в результате 0:

$$\begin{array}{r} \underline{29} \quad | \quad 2 \\ \underline{28} \quad \underline{14} \quad | \quad 2 \\ \mathbf{1} \quad \underline{14} \quad \underline{7} \quad | \quad 2 \\ \mathbf{0} \quad \underline{6} \quad \underline{3} \quad | \quad 2 \\ \mathbf{1} \quad \underline{2} \quad \underline{1} \quad | \quad 2 \\ \mathbf{1} \quad \underline{1} \quad \underline{0} \quad | \quad 2 \\ \mathbf{1} \end{array}$$

Можно записать те же действия без деления в столбик:

Выражение	Частное	Остаток
$29 : 2$	14	1
$14 : 2$	7	0
$7 : 2$	3	1
$3 : 2$	1	1
$1 : 2$	0	1

Выпишем остатки в обратном порядке начиная с последнего: 11101. Таким образом, $29_{10} = 11101_2$.

Ответ: 11101.

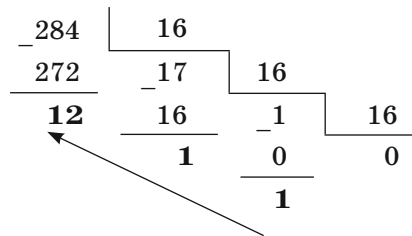
Пример 8

Перевести число 284_{10} в шестнадцатеричную систему счисления.

Решение. Разделим число 284 на 16 в столбик с остатком до получения нуля.

Выпишем остатки в обратной последовательности, заменяя те, что больше 10, на соответствующие шестнадцатеричные цифры (12 → С): $11C$. Таким образом, $284_{10} = 11C_{16}$.

Ответ: 11C.



Пример 9

Количество значащих нулей в двоичной записи десятичного числа 105 равно:

- 1) 2 2) 3 3) 4 4) 1

Решение. Преобразуем десятичную запись числа в двоичную, воспользовавшись последовательными делениями.

Выпишем полученные остатки в обратной последовательности: 1101001. Таким образом, $105_{10} = 1101001_2$. В двоичном числе всего 3 нуля.

Ответ: 2).

Выражение	Частное	Остаток
$105 : 2$	52	1
$52 : 2$	26	0
$26 : 2$	13	0
$13 : 2$	6	1
$6 : 2$	3	0
$3 : 2$	1	1
$1 : 2$	0	1

ПЕРЕВОД ЦЕЛЫХ ЧИСЕЛ ИЗ НЕДЕСЯТИЧНОЙ СИСТЕМЫ В НЕДЕСЯТИЧНУЮ

При работе с недесятичными системами наиболее часто возникает необходимость перевода значений между двоичной, восьмеричной и шестнадцатеричной системами счисления. Эта задача облегчается тем, что 8 и 16 являются степенями двойки: $8 = 2^3$; $16 = 2^4$. Поэтому каждая цифра восьмеричной системы представляется тройками двоичных разрядов (**триадами**), и каждая цифра шестнадцатеричной системы представляется четверками двоичных разрядов (**тетрадами**).

Для замены цифр на триады или тетрады можно использовать таблицы соответствия восьмеричных или шестнадцатеричных цифр записи в двоичной системе:

Двоичные триады	000	001	010	011	100	101	110	111
Восьмеричные цифры	0	1	2	3	4	5	6	7

Двоичные тетрады	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
16-ричные цифры	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
10-ричные цифры	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Для перевода целого числа из двоичной системы счисления в восьмеричную следует:

- 1) дополнить число слева незначащими нулями так, чтобы число цифр стало кратным 3;
- 2) разбить число на триады (группы по 3 цифры);
- 3) заменить каждую триаду соответствующей восьмеричной цифрой:

$$11101_2 \rightarrow 011101_2 \rightarrow \underbrace{011}_2 \underbrace{101}_2 \rightarrow 35_8$$

Для перевода целого числа из двоичной системы счисления в шестнадцатеричную нужно:

- 1) дополнить число слева незначащими нулями так, чтобы число цифр стало кратным 4;
- 2) разбить его на тетрады (группы по 4 цифры);
- 3) заменить каждую тетраду соответствующей шестнадцатеричной цифрой.

$$11101_2 \rightarrow \underbrace{00011101}_2 \rightarrow 0001 \ 1101_2 \rightarrow 1D_{16}$$

Для перевода целого числа из шестнадцатеричной или восьмеричной системы счисления в двоичную используются однотипные действия:

- 1) заменить каждую цифру числа соответствующей группой двоичных цифр (триадой для восьмеричной системы или тетрадой для шестнадцатеричной системы);
- 2) отбросить незначащие нули слева.

$$1D_{16} \rightarrow \underbrace{0001} \underbrace{1101}_2 \rightarrow 11101_2$$

$$35_8 \rightarrow \underbrace{011} \underbrace{101}_2 \rightarrow 11101_2$$

Для перевода целого числа между шестнадцатеричной и восьмеричной системами счисления можно использовать их связь с двоичной системой. Для этого следует:

- 1) представить число в двоичной системе;
- 2) выделить в нем тетрады (для шестнадцатеричной системы) или триады (для восьмеричной системы);
- 3) записать их цифрами необходимой системы счисления.

$$35_8 \rightarrow 11101_2 \rightarrow \underbrace{0011101}_2 \rightarrow 1D_{16}$$

$$1D_{16} \rightarrow 11101_2 \rightarrow \underbrace{011101}_2 \rightarrow 35_8$$

Пример 10

Перевести число 101110_2 в шестнадцатеричную систему счисления.

Решение. Способ 1. Поскольку в исходном числе разрядность (количество цифр) 6, что не кратно 4, дополним число слева двумя незначащими нулями до достижения кратности 4. Разобьем число на группы по 4 цифры (тетрады):

$$101110 = \underbrace{0010} \underbrace{1110}$$

По таблице соответствия находим шестнадцатеричные цифры: для первой тетрады — 2, для второй — E. Следовательно, $101110_2 = 2E_{16}$.

Способ 2. Для перевода числа без использования таблицы соответствия можно каждую из полученных тетрад представить сначала в десятичной системе, а затем заменить результат соответствующей шестнадцатеричной цифрой.

Первая тетрада дает десятичное значение 2, что соответствует шестнадцатеричной цифре 2:

$$0010_2 = 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 2_{10} = 2_{16}.$$

Вторая тетрада переводится в десятичное значение 14, которому соответствует шестнадцатеричная цифра E:

$$1110_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 14_{10} = E_{16}.$$

Таким образом, $101110_2 = 2E_{16}$.

Ответ: $2E_{16}$.

► Пример 11

Перевести число 612_8 в двоичную систему счисления.

Решение. Заменяем каждую цифру числа соответствующей двоичной триадой: $6 \rightarrow 110$, $1 \rightarrow 001$, $2 \rightarrow 010$. В результате получим: 110001010 . Таким образом, $612_8 = 110001010_2$.

Ответ: 110001010 .

1.2. ПЕРЕДАЧА ИНФОРМАЦИИ

1.2.1. Процесс передачи информации, источник и приемник информации, сигнал, скорость передачи информации

Информационный процесс — совокупность последовательных действий над информацией для получения какого-либо результата.

К информационным процессам относят:

- *сбор информации* — поиск и отбор необходимых сообщений из разных источников;
- *поиск информации* — нахождение информации в имеющихся информационных фондах;
- *обработка информации* — получение новых сообщений из уже имеющихся;
- *представление информации* — преобразование информации в форму, наиболее удобную для ее понимания и использования;
- *передача информации* — перемещение сообщений от источника к приемнику по каналу передачи;
- *хранение информации* — фиксирование сообщений на материальном носителе;
- *использование информации* — принятие на ее основе каких-либо решений;
- *защита информации* — предотвращение от случайной потери, повреждения, изменения или несанкционированного доступа к информации.

В процессе **передачи информации** всегда имеется несколько участников:

- тот, кто предоставляет информацию (выступает ее **источником**);
- тот, кто принимает информацию и является ее **получателем** (таких может быть несколько);
- **канал связи**, по которому передается информация.

Общую схему передачи информации (рис. 1.2) разработал основоположник цифровой связи (создатель теории информации) Клод Шеннон.

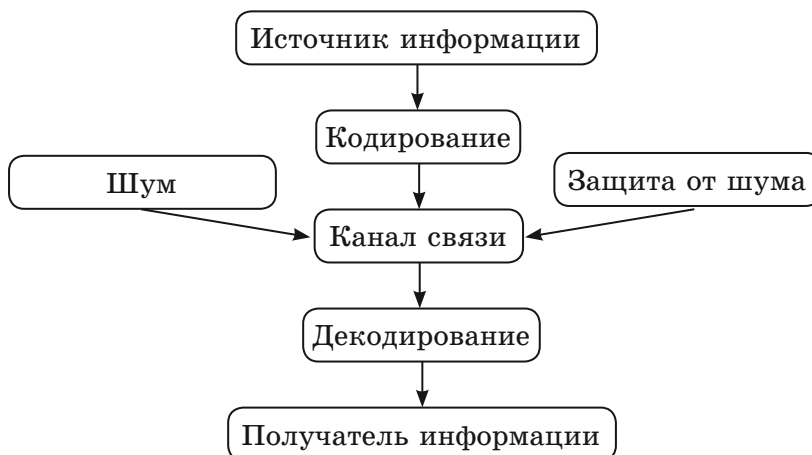


Рис. 1.2. Схема передачи информации

Передача информации означает ее перемещение в виде информационных сообщений в пространстве — от источника к приемнику. Передаваемое источником сообщение кодируется в передаваемый сигнал.

Источниками и приемниками информации могут быть живые существа или технические устройства. Каналами связи могут быть, например, электромагнитные, звуковые и световые волны.

Информационные сообщения передаются по каналам связи в форме сигналов. **Сигнал** — это изменение во времени некоторой физической величины (например, уровня напряжения). Именно изменения некоторых параметров (характеристик) сигнала отображают сообщение. Таким образом, сигналы являются материально-энергетической формой представления информации.

Сигналы могут быть *аналоговыми* (непрерывными) или *дискретными* (импульсными). Сигнал является **дискретным**, если его параметр может принимать только конечное число значений и существует лишь в конечное число моментов времени. В компьютерах используются сигналы, которые могут принимать только два дискретных значения — 0 и 1.

По *способу передачи сигналов* различают каналы *проводной связи* (например, кабельные) и каналы *радиосвязи* (например, спутниковые).

По *типу среды распространения* каналы связи делятся на *проводные*, *акустические*, *оптические*, *инфракрасные* и *радиоканалы*. Например, один из современных каналов передачи информации — световод (оптоволокно) — позволяет передавать сигналы лазеров на расстояние более 100 км без усиления.

Основной характеристикой каналов передачи информации является их *пропускная способность*, или скорость передачи по каналу информации.

Скорость передачи информации (информационных сообщений) — количество информации, переданное в единицу времени. Скорость передачи сообщений обычно измеряется в битах за секунду (бит/с). Кроме того, используются другие единицы: килобиты за секунду (Кбит/с), мегабиты за секунду (Мбит/с), байты за секунду (Б/с), килобайты за секунду (Кб/с).

Скорость передачи информации отображает, как быстро передается информация от источника к получателю — безотносительно к тому, по каким каналам происходит передача.

Пропускная способность канала — максимальное количество переданной или полученной по этому каналу информации за единицу времени. Таким образом, пропускная способность канала — максимально возможная скорость передачи информации по этому каналу. Например, пропускная способность современных оптоволоконных каналов — более 100 Мбит/с, т. е. в миллиарды раз выше, чем у нервной системы человека при чтении текстов.

Пропускная способность канала измеряется в тех же единицах, что и скорость передачи информации.

В сетях передачи данных по одному каналу может одновременно происходить огромное количество процессов передачи информации (от многих источников ко многим получателям). При этом скорость передачи информации для каждой конкретной пары «источник — получатель» может быть разной, а пропускная способность канала — величина, как правило, постоянная.

1.2.2. Кодирование и декодирование информации

Кодирование информации

Кодирование информации — процесс преобразования сигнала из формы, удобной для непосредственного использования информации, в форму, удобную для передачи, хранения или автоматической переработки.

Декодирование — обратный процесс восстановления информации из закодированного представления.

В процессах восприятия, передачи и хранения информации живыми организмами, человеком и техническими устройствами происходит кодирование информации. В этом случае информация, представленная в одной знаковой системе, преобразуется в другую. Каждый символ *исходного алфавита* представляется конечной последовательностью символов *кодированного алфавита*. Эта результирующая последовательность называется **информационным кодом** (*кодowym словом*, или просто *кодом*).

Примерами кодов являются последовательность букв в тексте, цифр в числе, двоичный компьютерный код и др.

Код состоит из определенного количества знаков (имеет определенную длину), которое называется **длиной кода**. Например, текстовое сообщение состоит из определенного количества букв, число — из определенного количества цифр.

Преобразование знаков или групп знаков одной знаковой системы в знаки или группы знаков другой знаковой системы называется **перекодированием**.

При кодировании один символ исходного сообщения может заменяться одним или несколькими символами нового кода, и наоборот — несколько символов исходного сообщения могут быть заменены одним символом в новом коде. Примером такой замены служат китайские иероглифы, которые обозначают целые слова и понятия.

Кодирование может быть *равномерным* и *неравномерным*. При **равномерном кодировании** все символы заменяются кодами равной длины; при **неравномерном кодировании** разные символы могут кодироваться кодами разной длины (это затрудняет декодирование). Неравномерный код называют еще **кодом переменной длины**.

Примером неравномерного кодирования является код *азбуки Морзе*. Длительное время он использовался для передачи сообщений по телеграфу. Кодовый алфавит включал точку, тире и паузу. При передаче по телеграфу точка означала кратковременный сигнал, тире — сигнал в 3 раза длиннее. Между сигналами букв одного слова делалась пауза длительностью одной точки, между словами — длительностью трех точек, между предложениями — длительностью семи точек.

Вначале код Морзе был создан для букв английского алфавита, цифр и знаков препинания. Принцип этого кода заключался в том, что часто встречающиеся буквы кодировались более простыми сочетаниями точек и тире. Это делало код компактным. Позже код был разработан и для символов других алфавитов, включая русский.

Коды Морзе для некоторых букв

Буква	А	Б	В	Г	Д	Е	И	К	О	Р	С	Т
Код Морзе	· —	— ···	· — —	— — ·	— · ·	·	· ·	— · —	— — —	· — ·	· · ·	—

В коде Морзе имеются кодовые слова, которые совпадают с началом других кодовых слов — например, код буквы А совпадает с началом кодов букв В, Л, П, Р и других; код буквы Т — с началом кодов букв Б, Г, Д, К, О и т. д. Это сделало бы код неоднозначным, если бы в нем использовались только точки и тире. Например, последовательность · — · — · — · — · — можно прочесть по-разному:

«ааааа» — если разбить ее · — · — · — · — · —

«арка» — если разбить · — · — · — · — · —

либо «арк», «рака», «ркаа». Чтобы избежать неоднозначности, код Морзе включает также паузы между кодами разных символов.

Пример 1

Для кодирования букв Р, И, К, П, А используется двоичное представление чисел 0, 1, 2, 3 и 4 соответственно (с сохранением одного незначащего нуля в случае одноразрядного представления). Закодировать последовательность букв ПАПРИКА. Результат записать шестнадцатеричным кодом.

Решение. Запишем двоичные коды указанных чисел (0 и 1 представим соответственно как 00 и 01):

Р	И	К	П	А
0	1	2	3	4
00	01	10	11	100

Закодируем последовательность букв ПАПРИКА — 1110011000110100.

Разобьем полученное представление на четверки (тетрады знаков) справа налево и переведем полученный набор чисел сначала в десятичный код:

1110 0110 0011 0100 → 14 6 3 4.

Полученную десятичную запись переведем в шестнадцатеричный код:

14 6 3 4 → E634.

Вместо перевода в десятичную систему счисления можно было воспользоваться таблицей соответствия шестнадцатеричных цифр и двоичной записи:

1110 → E; 0110 → 6; 0011 → 3; 0100 → 4.

Таким образом, результирующий код — E634₁₆.

Ответ: E634.

Декодирование информации

В зависимости от системы кодирования информационный код может или не может быть декодирован однозначно.

Равномерные коды всегда могут быть декодированы однозначно.

Для однозначного декодирования неравномерного кода важно, имеются ли в нем кодовые слова, которые являются одновременно началом других, более длинных кодовых слов.

Закодированное сообщение можно однозначно декодировать *с начала*, если выполняется **условие Фано**: никакое кодовое слово не является началом другого кодового слова.

Закодированное сообщение можно однозначно декодировать *с конца*, если выполняется **обратное условие Фано**: никакое кодовое слово не является окончанием другого кодового слова.

Неравномерные коды, для которых выполняется условие Фано, называются *префиксными*. **Префиксный код** — такой неравномерный код, в котором ни одно кодовое слово не является началом другого, более длинного слова. В таком случае кодовые слова можно записывать друг за другом без разделительного символа между ними.

Например, код Морзе не является префиксным — для него не выполняется условие Фано. Поэтому в кодовый алфавит Морзе, кроме точки и тире, входит также символ-разделитель — пауза длиной в тире. Без разделителя однозначно декодировать код Морзе в общем случае нельзя.

Условие Фано — это достаточное, но не необходимое условие однозначного декодирования.

► Пример 2

Для 5 букв латинского алфавита заданы их двоичные коды (для некоторых букв — из двух бит, для некоторых — из трех).

A	B	C	D	E
100	110	011	01	10

Определить, какой набор букв закодирован двоичной строкой 1000110110110, если известно, что все буквы в последовательности — разные.

Решение. В примере используется неравномерное кодирование (коды имеют разную длину). Условие Фано не выполняется: двоичный код символа D (01) служит началом кода символа C (011), код символа E (10) — началом кода символа A (100).

Обратное условие Фано также не выполняется: двоичный код символа E (10) служит окончанием кода символа B (110).

Следовательно, код можно раскодировать неоднозначно. Воспользуемся дополнительным условием — все буквы в последовательности разные. Рассмотрим все возможные варианты. Из них отбросим те, в которых получаются повторяющиеся буквы.

Начнем декодирование с начала цепочки. Первой буквой может быть A или E. Рассмотрим эти случаи отдельно.

1) Если первая буква E, то дальнейшую цепочку 00110110110 нельзя разложить на заданные коды букв (нет кода, начинающегося с 00).

2) Если первая буква A (код 100), то останется цепочка 0110110110. В ее начале может быть код букв C (011) или D (01).

3) Если вторая буква цепочки D, то остается цепочка 10110110 — она может начинаться только с кода буквы E (10) (так как буква A уже участвовала в коде). В таком случае оставшаяся часть 110110 будет содержать повторение кодов символа B, что противоречит условию. Следовательно, вторая буква цепочки — C.

4) Если вторая буква цепочки — C, остается цепочка 0110110. Она может начинаться только с кода символа D (01), поскольку символ C уже использован. Тогда дальнейший остаток — 10110. Он однозначно декодируется как EB. Ни один символ не повторяется.

Таким образом, декодированная последовательность символов — ACDEB.

Ответ: ACDEB.

Пример 3

В радиограмме использовались только 4 буквы из азбуки Морзе:

А	Б	Н	Д
· —	— · · ·	— ·	— · ·

Определите текст радиограммы, в которой было утеряно разбиение на буквы:
 — · · · · — — · — · · · — — · · —

Решение. Использованный код неравномерный, т. к. имеет разную длину для разных исходных символов. Проверим, является ли какое-либо кодовое слово началом другого кодового слова. Код буквы Н является началом кода буквы Д, и оба они являются началом кода буквы Б. Условие Фано не выполняется.

Однако выполняется обратное условие Фано: никакой код не является окончанием другого кода. Следовательно, однозначная расшифровка сообщения возможна. Расшифровку нужно начинать с конца.

Последняя буква сообщения определяется однозначно — А: только ее код заканчивается на тире. Тогда предпоследняя буква — однозначно Н: только ее код заканчивается на «тире — точка». Так просматриваем сообщение до его начала. В результате получаем исходный текст — БАНДАНА.

Ответ: БАНДАНА.

1.3. ОБРАБОТКА ИНФОРМАЦИИ

1.3.1. Алгоритм, свойства алгоритмов, способы записи алгоритмов. Блок-схемы. Представление о программировании

Понятие алгоритма является одним из основных понятий вычислительной математики и информатики.

Алгоритм — строго определенная последовательность действий для некоторого исполнителя, приводящая к поставленной цели или заданному результату за конечное число шагов.

Любой алгоритм составляется в расчете на конкретного *исполнителя* с учетом его возможностей. **Исполнитель** — субъект, способный исполнять некоторый набор команд. Совокупность команд, которые исполнитель может понять и выполнить, называется **системой команд исполнителя**.

Для выполнения алгоритма исполнителю недостаточно только самого алгоритма. Выполнить алгоритм — значит применить его к решению конкретной задачи, т. е. выполнить запланированные действия по отношению к определенным входным данным. Поэтому исполнителю необходимо иметь **исходные (входные) данные** — те, что задаются до начала алгоритма.

В результате выполнения алгоритма исполнитель должен получить *искомый результат* — **выходные данные**, которые исполнитель выдает как результат выполненной работы. В процессе работы исполнитель может создавать и использовать данные, не являющиеся выходными, — **промежуточные данные** (рис. 1.3).



Рис. 1.3. Назначение алгоритма — преобразование исходных данных в искомый результат

Алгоритм должен обладать определенными свойствами. Наиболее важные **свойства алгоритма**:

- **Дискретность.** Процесс решения задачи должен быть разбит на последовательность отдельных шагов — простых действий, которые выполняются одно за другим в определенном порядке. Каждый шаг называется *командой* (инструкцией). Только после завершения одной команды можно перейти к выполнению следующей.
- **Конечность.** Исполнение алгоритма должно завершиться за конечное число шагов; при этом должен быть получен результат.
- **Понятность.** Каждая команда алгоритма должна быть понятна исполнителю. Алгоритм должен содержать только те команды, которые входят в систему команд его исполнителя.

- **Определенность (детерминированность).** Каждая команда алгоритма должна быть точно и однозначно определена. Также однозначно должно быть определено, какая команда будет выполняться на следующем шаге. Результат выполнения команды не должен зависеть ни от какой дополнительной информации. У исполнителя не должно быть возможности принять самостоятельное решение (т. е. он исполняет алгоритм *формально*, не вникая в его смысл). Благодаря этому любой исполнитель, имеющий необходимую систему команд, получит один и тот же результат на основании одних и тех же исходных данных, выполняя одну и ту же цепочку команд.
- **Массовость.** Алгоритм предназначен для решения не одной конкретной задачи, а целого класса задач, который определяется диапазоном возможных входных данных.

Способы представления алгоритмов:

- *словесная запись* (на естественном языке). Алгоритм записывается в виде последовательности пронумерованных команд, каждая из которых представляет собой произвольное изложение действия;
- *блок-схема* (графическое изображение). Алгоритм представляется с помощью специальных значков (геометрических фигур) — *блоков*;
- *формальные алгоритмические языки*. Для записи алгоритма используется специальная система обозначений (искусственный язык, называемый алгоритмическим);
- *псевдокод*. Запись алгоритма на основе синтеза алгоритмического и обычного языков. Базовые структуры алгоритма записываются строго с помощью элементов некоторого базового алгоритмического языка.

Словесная запись алгоритма

Произвольное изложение этапов алгоритма на естественном языке имеет свои недостатки. Словесные описания строго не формализуемы, поэтому может быть нарушено свойство определенности алгоритма: исполнитель может неточно понять описание этапа алгоритма. Словесная запись достаточно многословна. Сложные задачи трудно представить в словесной форме.

Пример 1

Записать в словесной форме правило деления обыкновенных дробей.

Решение.

Шаг 1. Числитель первой дроби умножить на знаменатель второй дроби.

Шаг 2. Знаменатель первой дроби умножить на числитель второй дроби.

Шаг 3. Записать дробь, числителем которой является результат выполнения шага 1, знаменателем — результат выполнения шага 2.

Описанный алгоритм применим к любым двум обыкновенным дробям. В результате его выполнения будут получены выходные данные — результат деления двух дробей (исходных данных).

Формальные исполнители алгоритма

Формальный исполнитель — это исполнитель, который выполняет все команды алгоритма строго в предписанной последовательности, не вникая в его

смысл, не внося ничего в алгоритм и ничего не отбрасывая. Обычно под формальным исполнителем понимают технические устройства, автоматы, роботов и т. п. Компьютер можно считать формальным исполнителем.

Программы на языке произвольного формального исполнителя могут состоять только из элементарных команд, которые входят в его систему (которые исполнитель «понимает»).

Исполнитель может иметь свою среду (например, систему координат, клеточное поле и др.). *Среда исполнителя* — это совокупность объектов, над которыми он может выполнять определенные действия (команды), и связей между этими объектами. Алгоритмы в этой среде выполняются исполнителем по шагам.

Пример 2

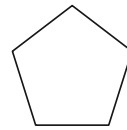
Исполнитель Крот имеет следующую систему команд:

1. вперед k — продвижение на указанное число шагов вперед;
2. поворот s — поворот на s градусов по часовой стрелке;
3. повторить m [команда1 ... командаN] — повторить m раз серию указанных команд.

Какой след оставит за собой исполнитель после выполнения следующей последовательности команд?

Повторить 5 [вперед 10 поворот 72]

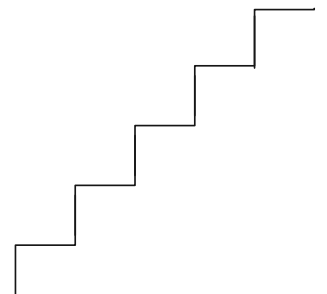
Решение. Команда вынуждает исполнителя 5 раз повторить набор действий: пройти 10 шагов вперед и повернуть на 72° по часовой стрелке. Так как поворот происходит на один и тот же угол, то за весь путь исполнитель повернет на $5 \times 72^\circ = 360^\circ$. Поскольку все отрезки пути одинаковой длины и сумма внешних углов любого многоугольника составляет 360° , то в результате будет оставлен след в форме правильного пятиугольника со стороной в 10 шагов исполнителя.



Заметим, что если увеличить количество повторов серии команд, то исполнитель будет повторно передвигаться по тем же отрезкам (произойдет повторное движение по тому же пятиугольнику).

Пример 3

В системе команд предыдущего исполнителя Крот сформировать алгоритм вычерчивания пятиступенчатой лестницы (длина ступеньки — 10 шагов исполнителя).



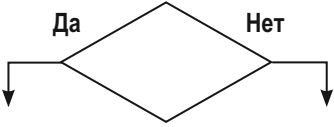
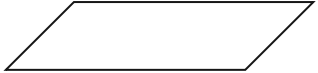
Решение. За каждый шаг цикла должно происходить 4 действия: движение вперед на 10 шагов исполнителя, поворот на 90° по часовой стрелке, еще 10 шагов вперед и поворот на 90° против часовой стрелки ($= 270^\circ$ по часовой). В результате за один шаг цикла формируется ломаная из двух отрезков длиной 10 под прямым углом. За пять таких шагов сформируется 5-ступенчатая лестница (ломаная будет содержать 10 звеньев).

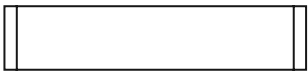

Повторить 5 [вперед 10 поворот 90 вперед 10 поворот 270]

Блок-схема

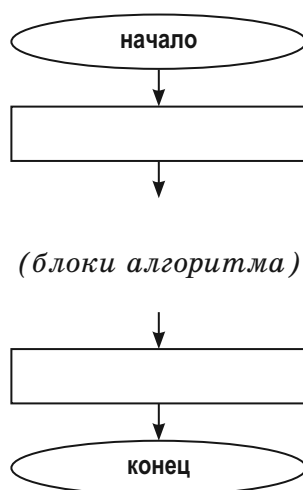
Блок-схема — наглядный способ представления алгоритма. Блок-схема отображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий. Определенному типу действия соответствует определенная геометрическая фигура блока. Линии, соединяющие блоки, определяют очередность выполнения действий. По умолчанию блоки соединяются сверху вниз и слева направо. Если последовательность выполнения блоков должна быть иной, используются направленные линии (стрелки).

Основные элементы блок-схемы алгоритма:

Внешний вид блока	Назначение блока
	<p>Функциональный блок (<i>операторный блок</i>, процесс). Служит для указания действия (шага) алгоритма. В прямоугольник входит одна направленная линия, из него выходит одна направленная линия. Внутри прямоугольника записывается команда, которая должна быть выполнена. Можно записывать несколько команд в одном блоке (для наглядности).</p>
	<p>Альтернативный блок (<i>условный блок</i>, условие). Служит для указания выбора одного из двух возможных действий. Внутри ромба размещается условие выбора (например, вопрос или сравнение). Условием может быть выражение, для которого возможно только одно из двух значений — либо «истина», либо «ложь».</p> <p>В ромб входит одна направленная линия. Из ромба выходят две направленные линии, каждая из которых подписана «Да» или «Нет». Если условие, записанное внутри ромба, будет верным (значение «истина», «да»), то управление будет передано по стрелке «Да», в противном случае — по стрелке «Нет».</p>
	<p>Блок начала / конца алгоритма (пуск/останов). Используется в начале и конце блок-схемы алгоритма</p>
	<p>Блок ввода/вывода. Служит для организации ввода исходных данных и вывода результирующих данных.</p>
	<p>Блок цикла. Предназначен для организации циклического процесса с параметром. Количество повторений (<i>итераций</i>) цикла и шаг изменения параметра должны быть известны. Внутри блока указываются (через запятую) начальное значение параметра цикла, конечное значение и шаг его изменения.</p>

Внешний вид блока	Назначение блока
	Блок предопределенного процесса (подпрограммы). Используется для указания обращения к отдельным модулям, вспомогательным алгоритмам, библиотечным подпрограммам.
	Блок печати Указывает вывод результатов на печать.

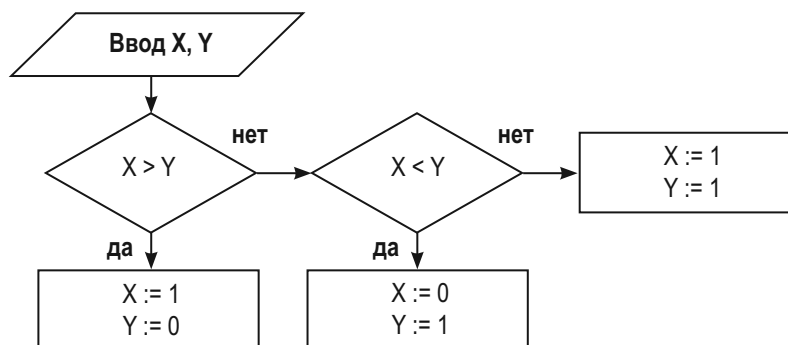
Общий вид блок-схемы алгоритма:



Пример 4

Алгоритм целочисленных преобразований представлен в виде фрагмента блок-схемы. Знаком $:=$ в нем обозначен оператор присваивания некоторого значения указанной переменной. Запись $X := 1$ означает, что переменная X принимает значение 1.

Определить результат работы алгоритма для исходных данных $X = 7, Y = 12$.



Решение. 1. Блок ввода данных определит исходные значения переменных X и Y (7 и 12 соответственно).

2. В первом условном блоке осуществляется сравнение значений X и Y. Поскольку условие, записанное в блоке, неверно ($7 < 12$), происходит переход по линии «нет».

3. Во втором условном блоке выполняется второе сравнение, которое для исходных данных оказывается верным. Происходит переход по линии «да».

4. Вычисляется результат выполнения алгоритма: $X := 0$, $Y := 1$.

Ответ: $X := 0$, $Y := 1$.

Алгоритмические языки

Алгоритмический язык — это искусственный язык (система обозначений), предназначенный для записи алгоритмов. Он позволяет представить алгоритм в виде текста, составленного по определенным правилам с использованием специальных *служебных* слов. Количество таких слов ограничено. Каждое служебное слово имеет точно определенный смысл, назначение и способ применения. При записи алгоритма служебные слова выделяют полужирным шрифтом или подчеркиванием.

В алгоритмическом языке используются формальные конструкции, но нет строгих синтаксических правил для записи команд. Различные алгоритмические языки различаются набором служебных слов и формой записи основных конструкций.

Алгоритмический язык, конструкции которого однозначно преобразуются в команды для компьютера, называется *языком программирования*. Текст алгоритма, записанный на языке программирования, называется *программой*.

Псевдокод

Псевдокод занимает промежуточное положение между естественным языком и языками программирования. Пример псевдокода — учебный алгоритмический язык. Алфавит учебного алгоритмического языка является открытым. Существенным достоинством этого языка является то, что его служебные слова, конструкции и правила записи алгоритма весьма схожи с теми, что применяются в распространенных языках программирования. Благодаря этому учебный алгоритмический язык позволяет легче освоить основы программирования.

Служебные слова учебного алгоритмического языка:

Категория	Служебные слова
общие	алг (заголовок алгоритма); нач (начало алгоритма); кон (конец алгоритма); арг (аргумент); рез (результат)
описание типов данных	цел (целый); сим (символьный); лит (литерный); лог (логический); вещ (вещественный); таб (таблица); длин (длина)
операции ввода/вывода	ввод ; вывод
логические операции	и ; или ; не

Категория	Служебные слова
логические значения	да; нет
операции ветвления	если; то; иначе; всё; при; выбор
операции цикла	нц (начало цикла); кц (конец цикла); пока; для; от; до

Стандартная структура алгоритма

Представление алгоритма на алгоритмическом языке (в том числе и языке программирования) состоит из двух частей. Первая часть — *заголовок* — задает название алгоритма и включает описание переменных, которые используются в нем. Вторая часть — *тело алгоритма* — содержит последовательность команд алгоритма.

Общий вид записи алгоритма на учебном алгоритмическом языке:

алг название (тип список имен аргументов и результатов; ...)

арг список имен

рез список имен

нач тип список имен промежуточных величин, ...

последовательность команд (тело алгоритма)

кон

В начале заголовка записывается служебное слово **алг**, после чего указывается имя алгоритма. Описание переменных, являющихся аргументами алгоритма и его результатами, приводится после названия в круглых скобках.

В следующих строках конкретизируют, какие именно переменные являются аргументами алгоритма (входными данными), а какие — его результатами (выходными данными). Для этого после служебного слова **арг** приводится список имен переменных-аргументов; в следующей строке после служебного слова **рез** приводится список имен переменных-результатов.

Между служебными словами **нач** и **кон** размещается *тело алгоритма* — конечная последовательность команд, выполнение которых предписывает алгоритм. Команды алгоритма записывают одну за одной в отдельных строках. В случае необходимости можно записать две или более команд в одной строке, тогда соседние команды разделяют точкой с запятой. Если в алгоритме применяются промежуточные переменные, их описание приводят в начальной строке тела алгоритма рядом со словом **нач**.

Примеры заголовков алгоритмов:

алг Объем_шара (**арг** вещ *Радиус*, **рез** вещ *Объем*)

алг Choice (**арг** цел *M, N*, **арг** лог *b*, **рез** вещ *Var1, Var2*)

В первом примере алгоритм имеет название *Объем_шара*, один вещественный аргумент *Радиус* и один вещественный результат *Объем*. Во втором примере алгоритм под названием *Choice* имеет три аргумента — целые *M* и *N* и логический *b*, а также два результата — вещественные *Var1* и *Var2*.

Пример алгоритма вычисления гипотенузы прямоугольного треугольника:

алг Гипотенуза (арг вещь a , b , рез вещь c)

нач

ввод a, b

$c := \text{sqrt}(a^2 + b^2)$

вывод c

кон

На вход алгоритму даются два вещественных аргумента a и b (величины катетов), результатом является вещественная переменная c (гипотенуза). Для ее расчета используется функция вычисления квадратного корня sqrt .

Описание величин и действия над ними

При описании алгоритма необходимо указать названия (обозначения) всех величин, которые будут в нем найдены или использованы.

При представлении алгоритма решения в виде блок-схемы выбранные обозначения величин приводятся отдельно от блок-схемы (как объяснение к ней). Если алгоритм представлен на языке программирования, то характеристика обрабатываемых величин включается в программу. Учебный алгоритмический язык также предусматривает описание величин, используемых в алгоритме.

Все величины в алгоритме разделяют на *постоянные (константы)* и *переменные*. **Константа** не может изменять свои значения в процессе работы алгоритма. Переменная может приобретать различные значения, которые сохраняются до тех пор, пока она не получит новое значение. Переменным величинам назначают имена. Таким образом, **переменная** — это именуемая величина, которая в процессе выполнения алгоритма может приобретать и хранить различные значения.

В алгоритмическом языке не существует специальных правил именования переменных. Однако их названия не должны совпадать со служебными словами алгоритмического языка. Во многих языках программирования для имен можно использовать только латинские буквы, цифры, знак подчеркивания. Имена обязательно должны начинаться с буквы, при этом строчные и прописные буквы в именах не различаются. В одном алгоритме не могут существовать разные объекты с одинаковыми именами. Все имена являются уникальными. Имена переменных и констант стараются выбирать так, чтобы они напоминали их смысл. Например, имена переменных и констант: S , $p12$, $result$, $umog$.

При представлении алгоритма на алгоритмическом языке именуются не только величины, но и сам алгоритм, и другие объекты. Имя алгоритма выбирают так же, как и имена переменных.

Величина — переменная, с которой связывается определенное множество значений. Этой величине присваивается имя (в языках программирования его называют *идентификатор*).

Значение — то, чему равна переменная в конкретный момент. Значение переменной можно задать двумя способами: присваиванием и с помощью процедуры ввода.

Тип переменной определяет диапазон всех значений, которые может принимать данная переменная, и допустимые для нее операции. Существует несколько predefined типов переменных. К *стандартным типам* относятся числовые, литерные и логические типы.

Числовой тип предназначен для обработки числовых данных. Различают целый и вещественный числовые типы. *Целый тип* в учебном алгоритмическом языке обозначается служебным словом *цел*, к нему относятся целые числа некоторого определенного диапазона. Они не могут иметь дробной части, даже нулевой. Число 123,0 является не целым, а вещественным числом. Вещественные величины относятся к *вещественному типу* данных и обозначаются в учебном алгоритмическом языке служебным словом *вещ*. Такие величины могут отображаться двумя способами: в форме с фиксированной запятой (например, 0,0511 или -712,3456) и с плавающей запятой (те же примеры: $5,11 \cdot 10^{-2}$ и $-7,123456 \cdot 10^2$).

Над числовыми данными можно выполнять арифметические операции и операции сравнения.

Обозначение операций:

сложение (+);
вычитание (-);
умножение (*);
деление (/);
возведение в степень (^);
больше (>);
меньше (<);
больше или равно (>=);
меньше или равно (<=);
равно (=);
не равно (\neq или <>).

Над целыми числами можно также выполнять две операции целочисленного деления *div* и *mod*. Операция *div* обозначает деление с точностью до целых чисел (остаток от деления игнорируется). Операция *mod* позволяет узнать остаток при делении с точностью до целых чисел. Например, результатом операции $100 \text{ div } 9$ будет число 11, а результатом $100 \text{ mod } 9$ — число 1.

Литерный тип представляет собой символы и строки, он дает возможность работать с текстом. *Литерные величины* — это произвольные последовательности символов. Эти последовательности заключаются в двойные кавычки: "результат", "sum_price". В качестве символов могут быть использованы буквы, цифры, знаки препинания, пробел и некоторые другие специальные знаки (возможными символами могут быть символы таблицы ASCII). В учебном алгоритмическом языке литерные величины обозначаются *лит*.

Над литерными величинами возможны операции сравнения и слияния. *Сравнение* литерных величин производится в соответствии с их упорядочением: "a" < "b", "b" < "c" и т. д. *Слияние (конкатенация)* литерных величин приводит к образованию новой величины: "пол" + "е" образует "поле".

Логический тип определяет логические переменные, которые могут принимать только два значения — *истина* (True) или *ложь* (False). Над логическими величинами можно выполнять все стандартные логические операции.

Команды учебного алгоритмического языка

Учебный алгоритмический язык использует следующие команды для реализации алгоритма:

- оператор присваивания :=
- операторы ввода/вывода:
ввод (список имен переменных)
вывод (список вывода)
- операторы ветвления
если... то... иначе... всё
выбор... при... иначе... всё
- операторы цикла
нц для... от... до... кц
нц пока... кц
нц ... до... кц

ОПЕРАЦИЯ ПРИСВАИВАНИЯ

Ко всем типам величин может быть применена **операция присваивания**, которая обозначается знаком «:=» и служит для вычисления выражения, стоящего справа, и присваивания его значения переменной, указанной слева. Например, если переменная H имела значение 12, а переменная M — значение 3, то после выполнения оператора присваивания $H := M + 10$ значение переменной H изменится и станет равным 13.

Вычисления в операторе присваивания выполняются справа налево: сначала необходимо вычислить значение выражения справа от знака присваивания. Поэтому допустимы конструкции вида $H := H + 10$. В этом случае сначала будет вычислено выражение в правой части ($12 + 10$), а его результат будет присвоен в качестве нового значения переменной H (значение 22).

Для оператора присваивания обязательно должны быть определены значения всех переменных в его правой части. Кроме того, типы данных в левой и правой части должны соответствовать друг другу.

ВВОД И ВЫВОД ДАННЫХ

В процессе работы алгоритма происходит обработка исходных данных для получения выходных (результатирующих) данных. В процессе этого преобразования могут быть найдены некоторые промежуточные результаты. Входные данные должны быть переданы алгоритму («введены»), а по окончании работы алгоритм должен вывести результат.

При записи алгоритма с помощью блок-схемы ввод и вывод данных отображаются с помощью блоков ввода/вывода (параллелограммов). При этом только указывается перечень данных для ввода или вывода, а сам процесс не детализируется.

Описание алгоритма средствами псевдокода может вовсе не предусматривать команды ввода или вывода данных. В заголовке алгоритма указывается, какие данные являются аргументами, какие — результатами работы алгоритма. Считается, что аргументы будут предоставлены до выполнения алгоритма, результаты будут выведены после его выполнения, и описывается лишь процесс превращения аргументов в результаты.

В записи алгоритма с помощью учебного алгоритмического языка для операций ввода/вывода используются команды **ввод** и **вывод**. После этих служебных слов указывается список ввода или вывода. Элементы этих списков перечисляются через запятую.

Список ввода может содержать только имена переменных. После выполнения команды **ввод** алгоритм получит значения перечисленных в списке переменных.

Список вывода может содержать имена переменных, константы и выражения. Если в списке вывода указано имя переменной, будет выведено ее значение. Если список вывода содержит выражение, будет выведен результат его вычисления. Текстовые константы следует записывать в списке вывода в кавычках (выводиться они будут без кавычек).

ПРИМЕР ОРГАНИЗАЦИИ ВВОДА/ВЫВОДА ДАННЫХ:

```
алг Расстояние (арг вещь  $v$ ,  $t$ , рез вещь  $S$ )
  нач
    ввод  $v$ ,  $t$ 
     $S := v * t$ 
    вывод "Путь ",  $S$ , " м"
  кон
```

Если при выполнении алгоритма ввести значения 20 и 10, то переменная v примет значение 20, а переменная t — значение 10. По окончании работы алгоритма будет выведен результат:

Путь 200 м

Тот же результат был бы получен, если бы изменить строку вывода на
вывод "Путь ", $v * t$, " м"

1.3.2. Алгоритмические конструкции

Различают три основных вида алгоритмов (базовые алгоритмические конструкции, или структуры): *линейные*, *с разветвлениями* и *с циклами*.

В самом простом случае алгоритм предписывает поочередное выполнение всех заданных действий независимо от значений входных данных. Например, чтобы умножить две обыкновенные дроби, необходимо перемножить отдельно их числители и знаменатели и записать их соответственно в числитель и знаменатель результата. Такие действия необходимо выполнять для умножения любых двух обыкновенных дробей.

Алгоритм, предписывающий одноразовое выполнение одной и той же последовательности действий при любых допустимых входных данных, называется **линейным (линейной структурой)**. Использование этой структуры возможно только для простых задач.

Для решения более сложных задач могут потребоваться алгоритмы, предусматривающие два возможных варианта действий. Выбор варианта зависит от некоторого условия. В таких случаях (когда алгоритм реализует выбор одного из альтернативных путей в зависимости от результатов проверки некоторого условия) говорят о ветвлении алгоритма. Например, для решения квадратного уравнения необходимо сначала найти значение дискриминанта, а затем, в зависимости от его знака, либо сообщить об отсутствии действительных корней (если дискриминант отрицательный), либо найти их по соответствующим формулам.

Алгоритм, предписывающий выполнение тех или других действий в зависимости от результата проверки условия, называется **разветвленным (структурой ветвления)**.

Хотя алгоритм ветвления содержит описание действий для обоих возможных вариантов, но при каждом его выполнении реализуется только один из них, какой именно — зависит от заданного набора входных данных. Следовательно, в отличие от линейного алгоритма, при реализации алгоритма с разветвлением будут выполнены не все действия, а только те, что выбраны по условию.

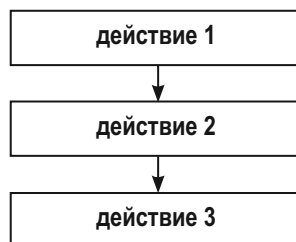
Третий вид алгоритмов (**с циклами**) обеспечивает многократное выполнение некоторой совокупности действий. Например, для вычисления разности двух чисел в столбик необходимо сначала вычесть последние цифры исходных чисел и записать последнюю цифру результата (если требуется, перенести единицу из предыдущего разряда). Затем аналогично следует вычислить разность предпоследних цифр чисел и так далее. Процедура повторяется, пока все цифры исходных чисел не будут исчерпаны. Количество повторений зависит от количества цифр в заданных числах.

Алгоритм, предписывающий повторное выполнение действий, называется **циклическим алгоритмом (алгоритмом с повторением, или структурой цикла)**. Повторяемое действие или группа действий называется *телом цикла*. Количество повторений тела цикла определяется поставленным условием, которое называется *условием цикла*. По результату проверки условия осуществляется выбор: еще раз повторить тело цикла или перейти к другим действиям.

Наличие возврата к ранее произведенным действиям является характерным отличием алгоритмов с циклами от линейных и разветвленных.

Линейные алгоритмические конструкции

В блок-схемах линейные алгоритмы представляют с помощью последовательности функциональных блоков:



В алгоритмическом языке линейным структурам соответствует последовательность команд языка:

Команда 1
Команда 2
Команда 3

У линейной структуры только один вход и только один выход, попасть извне в середину выполняемой последовательности команд невозможно.

► Пример 1

Определить значение целой переменной n после выполнения следующего алгоритма:

$m := 3$
 $n := 4$
 $m := 6 + m * n$
 $n := n + m / 3$

Решение. Первые две команды присваивания определяют начальные значения переменных. Для выполнения третьей команды сначала надо вычислить правую часть выражения: $6 + 3 \times 4 = 18$. Это значение будет присвоено переменной t . Для выполнения последней команды также сначала вычисляется правая часть выражения: $4 + 18/3 = 10$. Результат будет присвоен переменной n .

Ответ: 10.

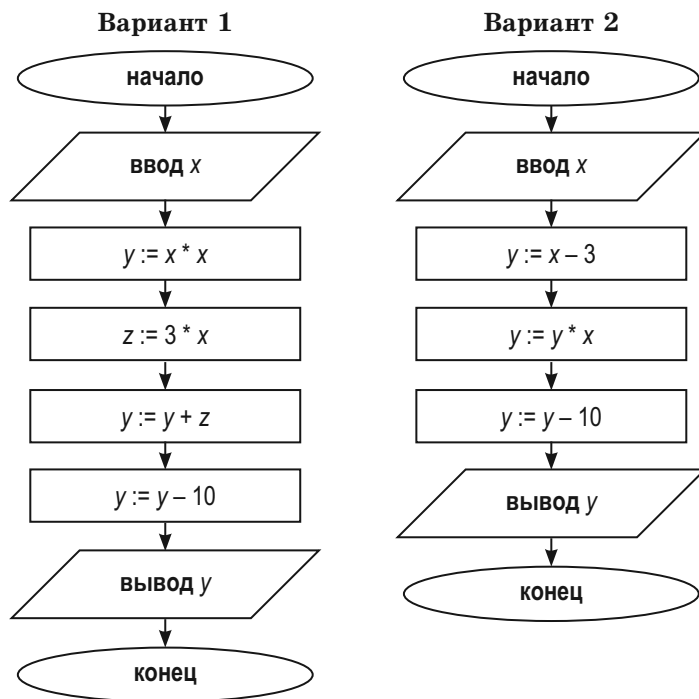
Даже для решения простых задач, ход которого не изменяется в зависимости от исходных данных, могут существовать разные варианты алгоритмов простой линейной структуры.

Пример 2

Вычислить значение выражения $x^2 - 3 \times x - 10$, используя только операции сложения и умножения.

Для решения задачи в той последовательности, что определяет само выражение, потребуется 2 операции умножения, 2 вычитания и 3 переменных (x , y , z). Однако можно вычислить то же выражение как $(x - 2) \times x - 10$. Такой алгоритм расчета потребует 1 умножение, 2 вычитания и 2 переменных (x , y).

Решение.



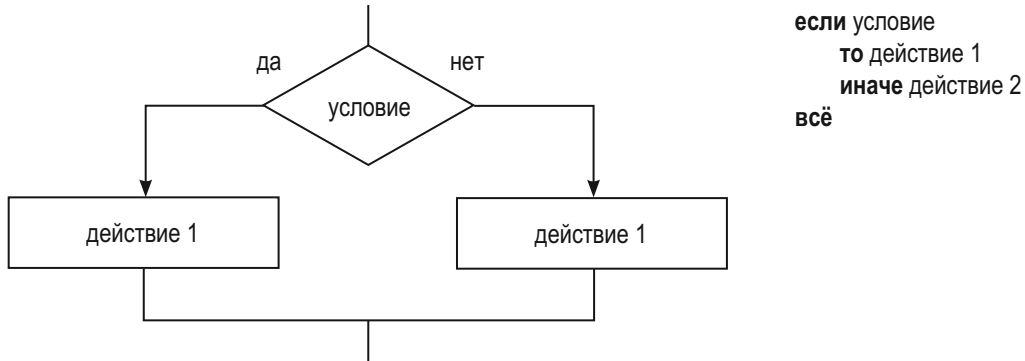
Алгоритмические конструкции ветвления

Для отображения базовой алгоритмической конструкции ветвления в блок-схемах используется альтернативный (условный) блок (фигура ромб), а в алгоритмических языках — команда **если**.

Существует две реализации структуры ветвления: *полная* и *неполная* (*краткая*). Обе формы ветвления являются замкнутыми: каждая из них имеет один вход и один выход.

Полная форма ветвления означает, что осуществляется выбор между двумя действиями. Если проверка условия дает результат «да», то выбирается действие 1; в противоположном случае (то есть если проверка условия дает результат «нет») — выбирается действие 2 (рис. 1.4).

Таким образом, полная форма команды **если** определяет две ветви команд: первая выполняется, если условие истинно, вторая — если условие ложно. Каждая ветвь в итоге ведет к общему выходу, так что работа алгоритма будет продолжаться при выборе любого пути.



```

если условие
  то действие 1
  иначе действие 2
все
  
```

Рис. 1.4. Полная форма ветвления
(блок-схема и команда алгоритмического языка)

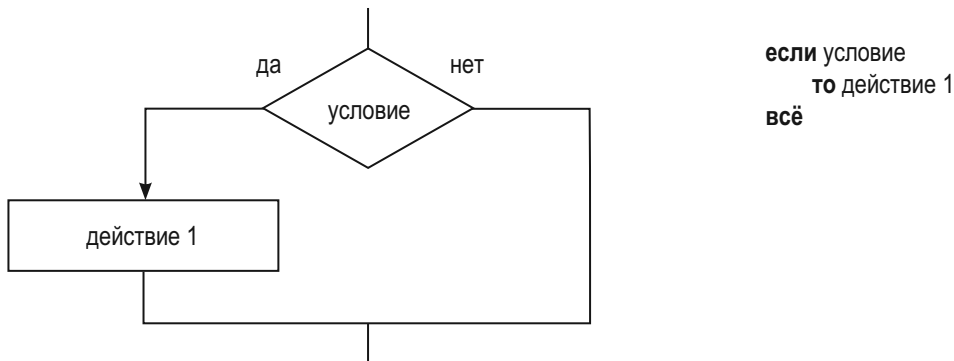
Пример 3

Для извлечения квадратного алгебраического корня из числа B необходимо проверить, положительно ли это число. Если проверка дает значение «истина» («да»), извлечь корень; если результат проверки — «ложь» («нет»), выдать соответствующее сообщение пользователю.

```

если  $B \geq 0$ 
  то вывод  $\sqrt{B}$ 
  иначе вывод "Отрицательное подкоренное выражение"
все
  
```

Краткая форма ветвления предполагает, что в случае истинности условия будет выполнена команда 1, а иначе — никакие действия не выполняются (рис. 1.5).



```

если условие
  то действие 1
все
  
```

Рис. 1.5. Краткая форма ветвления
(блок-схема и команда алгоритмического языка)

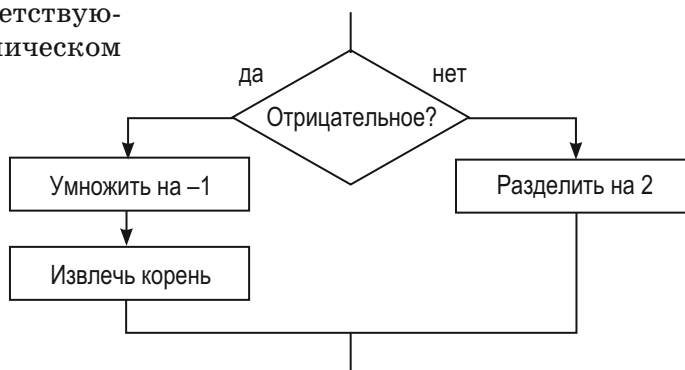
Пример 4

Для расчета значений гиперболы по формуле $y = 2/x$ необходимо проверить, что значение x не равно нулю. Если проверка условия дает значение «истина», — вычислить результат. В противном случае (если проверка условия дает значение «ложь», т. е. x равно нулю) — не выполнять никаких действий, поскольку деление на ноль запрещено.

```
если  $2/x \neq 0$   
то  $y := 2/x$   
всё
```

Пример 5

Для приведенного фрагмента блок-схемы алгоритма выбрать соответствующий ему фрагмент на алгоритмическом языке.



Варианты ответов:

- 1) **если** отрицательное
 то разделить на 2
 иначе умножить на -1 ; извлечь корень
 всё
- 2) **если** отрицательное
 то умножить на -1
 иначе разделить на 2
 всё
- 3) **если** отрицательное
 то извлечь корень; умножить на -1
 иначе разделить на 2
 всё
- 4) **если** отрицательное
 то умножить на -1 ; извлечь корень
 иначе разделить на 2
 всё

Решение. Ромбу на блок-схеме соответствует структура *ветвление* (команда **если**): проверяется условие «отрицательное». Если условие истинно, выполняется ветвь «да» (строка **то** на алгоритмическом языке) с двумя действиями: «умножить на -1 »; «извлечь корень». Если условие ложно, выполняется ветвь «нет» (строка **иначе** на алгоритмическом языке) с одним действием: «разделить на 2». После обоих вариантов структура завершается, что соответствует служебному слову **всё**. Такому положению соответствует вариант алгоритма 4.

Ответ: 4.

Для серии команд ветвления, следующих одна за другой (*множественное ветвление*), в алгоритмических языках существует специальная команда **выбор**. Она также имеет *полную* и *неполную (краткую)* формы.

Структура ветвления **выбор** предполагает поочередную проверку нескольких условий (одно за одним). Если проверяемое условие 1 истинно, выполняется

команда 1, если нет — переходят к проверке следующего условия. Если второе условие истинно — выполняется команда 2, если нет — проверяют следующее условие и т. д.

Полная и краткая формы структуры выбор различаются действиями после проверки последнего условия. Если проверка последнего условия выдает «ложь» («нет»), — в полной форме выполняют заданную команду; в краткой форме ничего не делают (рис. 1.6).

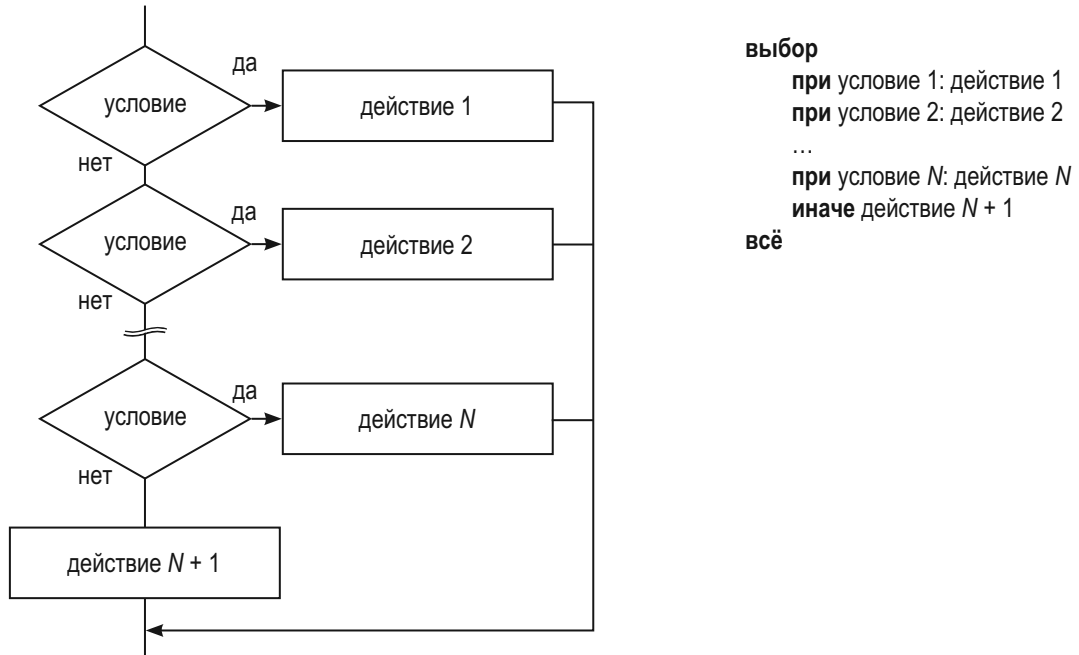


Рис. 1.6. Полная форма структуры **выбор**
 (блок-схема и команда алгоритмического языка)

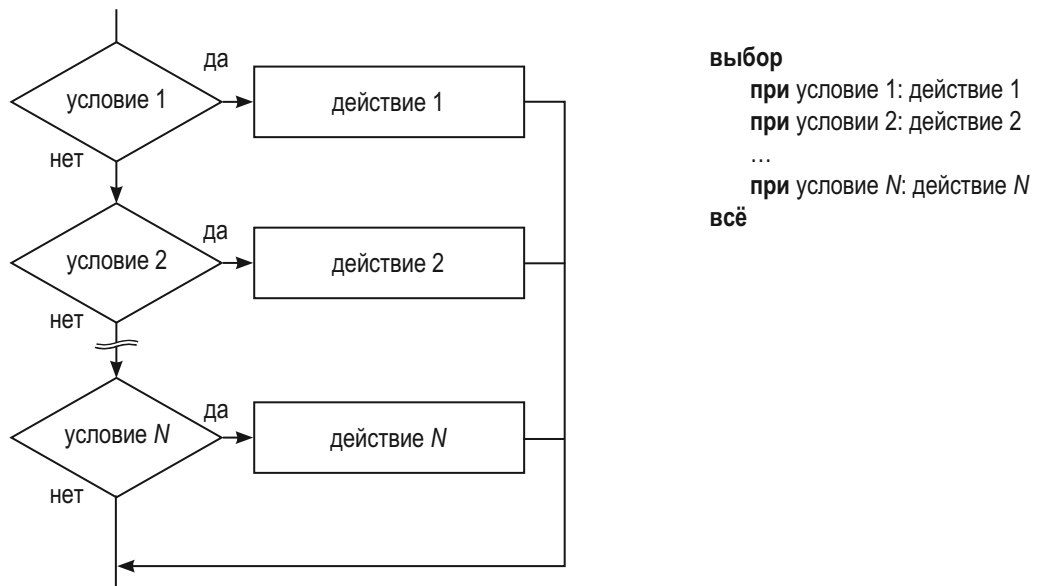


Рис. 1.7. Краткая форма структуры **выбор**
 (блок-схема и команда алгоритмического языка)

Во всех формах структур ветвления важную роль играют условия выбора. Часто в них используются операторы сравнения или другие отношения. Результатом условия может быть только одно из двух логических значений — *Ложь* или *Истина*. В языках программирования эти значения обычно записывают как *True* и *False*. В учебном алгоритмическом языке используют чаще значения «да» и «нет». В компьютерной форме эти значения обычно представлены битовыми значениями 0 и 1.

Простые условия обычно содержат только одну операцию — например, $X > 0$, $A \neq B$ или $s = \text{"отец"}$. Объединение нескольких простых условий образует *составное условное выражение* (или *составное условие*). Для объединения простых условий используются логические операторы:

- and* (логическое И),
- or* (логическое ИЛИ),
- xor* (исключающее ИЛИ),
- not* (отрицание).

Например, чтобы задать условие принадлежности числовой величины Z промежутку (10;20), следует записать составное условие $Z > 10$ and $Z < 20$.

Циклические конструкции

Базовая структура *цикл* (или *повторение*) обеспечивает многократное выполнение одних и тех же команд. Существует несколько разновидностей циклических структур. Любая из них содержит *тело цикла* (набор повторяющихся команд) и *заголовок цикла*, который определяет количество повторений тела цикла.

ЦИКЛ С ПРЕДУСЛОВИЕМ (ИЛИ ЦИКЛ «ПОКА»)

Заголовок этой структуры содержит условие, которое называется *предусловием*. Эта структура предписывает повторять тело цикла до тех пор, пока выполняется условие в его заголовке (т. е. пока оно остается истинным) (рис. 1.8).

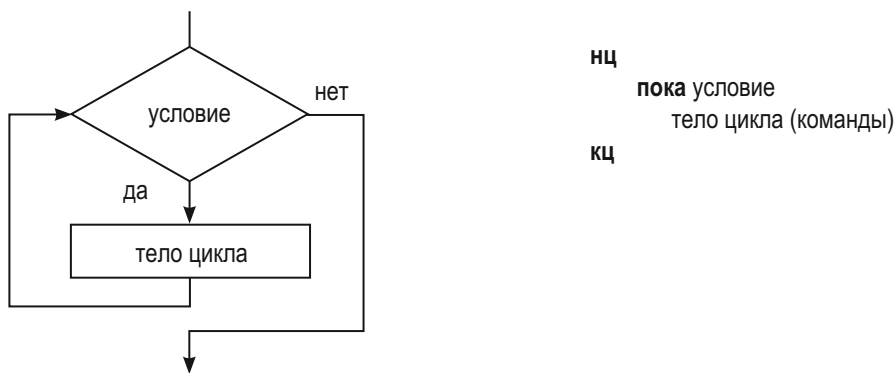


Рис. 1.8. Цикл с предусловием
(блок-схема и команда алгоритмического языка)

Работа цикла с предусловием начинается с проверки условия в его заголовке. Если условие истинно — выполняются команды тела цикла и происходит возврат к заголовку цикла. Снова проверяется условие заголовка (поскольку оно могло измениться в результате работы команд цикла) — если оно истинно, опять выполняется тело цикла. Так происходит до тех пор, пока проверка усло-

вия в заголовке не выдаст результат «нет». Тогда управление будет передано команде, следующей непосредственно за циклом.

Возможна ситуация, когда команды тела цикла не будут выполнены ни разу — если условие в заголовке сразу же выдает результат проверки «нет». Также возможна ситуация, когда условие заголовка всегда выдает положительный результат проверки — это приведет к бесконечному выполнению цикла, так называемому «зацикливанию» алгоритма. Таким образом, при создании цикла «пока» следует обращать особое внимание на формулировку условия в его заголовке.

Пример 6

Записать на алгоритмическом языке алгоритм получения остатка от деления целого числа a на целое число b с помощью вычитания.

Решение. Если число a меньше b , то остатком от деления служит само число a . В ином случае необходимо вычитать b из числа a до тех пор, пока результат не станет меньше b — он и будет остатком от деления.

```
нц
  пока  $a \geq b$ 
     $a := a - b$ 
кц
вывод  $a$ 
```

ЦИКЛ С ПОСТУСЛОВИЕМ (ИЛИ ЦИКЛ «ДО»)

Постусловие формулируется противоположным образом по отношению к предусловию. Цикл «до» предписывает повторять команды тела цикла до тех пор, пока не выполнится условие в его заголовке (т. е. пока оно остается ложным).

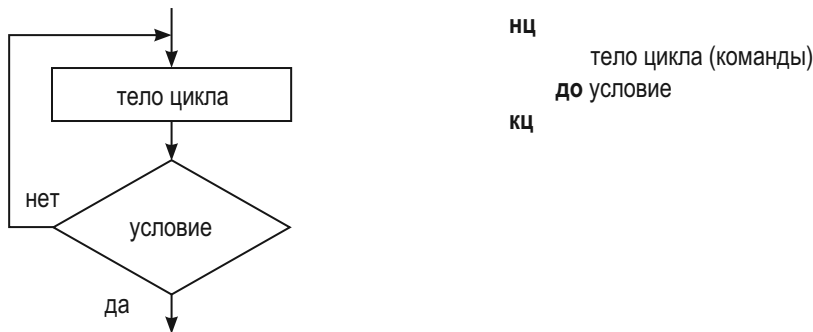


Рис. 1.9. Цикл с постусловием (блок-схема и команда алгоритмического языка)

Работа цикла с постусловием начинается с выполнения команд тела цикла. Затем проверяется условие цикла. Если условие ложно (проверка условия дает результат «нет») — происходит возврат к выполнению команд тела цикла. Затем снова проверяется условие (поскольку оно могло измениться в результате работы команд цикла). Так происходит до тех пор, пока проверка условия не выдаст результат «да». Тогда происходит выход из цикла и управление будет передано команде, следующей непосредственно за циклом.

В отличие от цикла с предусловием, тело цикла «до» всегда выполняется хотя бы один раз (до первой проверки). Тело цикла «пока» может не выполниться ни разу, если условие при первой же проверке выдаст «нет». Поэтому цикл с постусловием можно заменить циклом с предусловием, а наоборот — нет.

Пример 7

Записать алгоритм примера 6 с помощью цикла «до».

Решение.

```
нц
    a := a - b
до a < b
кц
вывод a
```

ЦИКЛ С ПАРАМЕТРОМ (ЦИКЛ СО СЧЕТЧИКОМ, ИЛИ ЦИКЛ «ДЛЯ»)

Эта структура предписывает повторять команды тела цикла для всех значений некоторой переменной (*параметра*, или *счетчика* цикла).

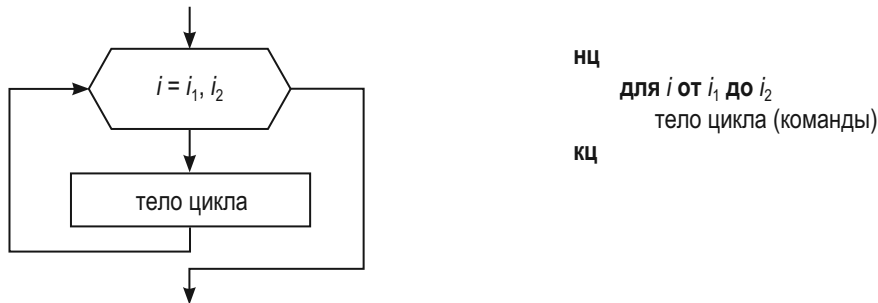


Рис. 1.10. Цикл со счетчиком (блок-схема и команда алгоритмического языка)

Имя параметра цикла (счетчика) указывают в заголовке после служебного слова *для*. Затем с помощью служебных слов *от* и *до* задают начальное и конечное значение этого параметра.

Работа цикла «для» начинается с присвоения параметру начального значения. Если оно не превышает конечного значения, выполняются команды тела цикла. Затем значение параметра цикла увеличивается на единицу. Если оно снова не превышает конечного значения, опять происходит выполнение тела цикла. Если же полученное значение параметра превысит конечное, цикл будет завершен и управление передано следующей за циклом команде.

Цикл со счетчиком всегда выполняется $i_1 - i_2 + 1$ раз.

Пример 8

Записать алгоритм вычисления факториала числа N .

Решение. Факториал числа вычисляется по формуле $N! = 1 \times 2 \times \dots \times N$. Следовательно, для расчета факториала надо организовать цикл со счетчиком, в котором перемножить последовательные целые числа от 2 до N . Значение $1!$, равное 1, можно присвоить результирующей переменной до цикла.

```
алг Факториал (арг цел N, рез цел factorial)
нач
    цел i
    factorial := 1
нц
    для i от 2 до N
        factorial := factorial * i
кц
вывод "N! = ", factorial
кон
```

Пример 9

Записать алгоритм сложения квадратов целых чисел от 1 до n включительно.

Решение.

алг Сумма_квадратов (арг цел n , рез цел S)

```
нач
  цел  $i$ 
   $S := 0$ 
нц
  для  $i$  от 1 до  $n$ 
     $S := S + i * i$ 
кц
вывод "S = ",  $S$ 
кон
```

Таблицы и массивы

Табличный способ организации данных позволяет удобно обрабатывать наборы однотипных данных. В языках программирования наборы табличных данных принято называть массивами. В алгоритмическом языке используют название «таблица».

Массив — конечный набор пронумерованных однотипных данных, имеющий имя.

Величины, которые входят в массив, называются его **элементами**. Количество элементов массива называется его **размерностью**.

На практике чаще всего используют массивы, содержащие числовые или литерные (символьные, текстовые) данные. Важным является требование однотипности данных.

Имя массива (таблицы) относится ко всему набору данных. Элементы массива различают по порядковому номеру, который называется **индексом**. Для обращения к элементу его индекс указывают в квадратных скобках сразу вслед за именем массива. Например, если массив имеет имя D , то третий его элемент обозначается $D[3]$. Такие имена называются *индексированными именами*. Индексы могут быть не только константами, но и переменными или целочисленными выражениями: $D[j]$, $D[k+1]$.

Различают *одномерные* и *многомерные* массивы.

Одномерный, или **линейный массив** представляет собой последовательность пронумерованных по порядку элементов. Одномерные таблицы называют иногда *векторами*. В них всегда только одна строка или один столбец.

В одномерных таблицах или массивах могут быть записаны данные, относящиеся только к какому-либо одному факту. Например, результаты соревнований спортсменов по одному виду спорта, или регулярные показания температуры в одной местности, или оценки учащихся по одному предмету, или названия альбомов одного исполнителя.

Одномерная таблица, содержащая расстояния от Солнца до планет Солнечной системы (в астрономических единицах), может выглядеть как табличная строка:

0,39	0,72	1	1,52	5,20	9,54	19,18	30,07	39,44
1	2	3	4	5	6	7	8	9
Меркурий	Венера	Земля	Марс	Юпитер	Сатурн	Уран	Нептун	Плутон

Названия планет в этой таблице отсутствуют (иначе они бы образовали вторую строку). Чтобы узнать расстояние от Солнца до Марса, надо обратиться к четвертому элементу этой таблицы. Если, например, в программе такой массив расстояний от Солнца был поименован как *Dist*, то для обращения к 4-му элементу массива надо указать *Dist[4]*.

Многомерные таблицы и массивы могут содержать данные о нескольких фактах — например, результаты соревнований спортсменов в нескольких видах спорта; измерения температуры, давления и влажности в разные месяцы в разных городах; оценки разных учащихся по разным предметам в разные дни; названия альбомов исполнителей, выпущенных в разные годы.

Пример многомерной таблицы, содержащей данные о некоторых странах Европы:

		Австрия 1	Бельгия 2	Велико- британия 3	Исландия 4	Испания 5	Норвегия 6	Финляндия 7	Франция 8	Швеция 9
Территория (тыс. кв. км)	3	83,8	30,5	244,8	103	504,8	387	337	551	450
Население (тыс. чел.)	1	8424	10431	62740	319	47270	4953	5388	65430	9449
Городское население (тыс. чел.)	2	4720	10120	55840	290	35930	3620	3390	47760	7840

В отличие от *индексированных* элементов массива обычные переменные иногда называют *скалярными*.

Для описания таблицы в алгоритмическом языке используется служебное слово **таб**. Необходимо указать по порядку: тип элементов таблицы, служебное слово **таб**, имя таблицы, а затем в квадратных скобках через двоеточие начальный и конечный номера элементов таблицы.

Общий вид описания таблицы:

тип элементов **таб** имя таблицы [наименьший индекс : наибольший индекс]

Например, для описания таблицы расстояний от Солнца необходимо учесть, что в ней содержатся нецелые числа, следовательно, тип данных таблицы — вещественный, и количество элементов равно 9:

вещ таб *Dist*[1:9]

Для присвоения значений элементам таблицы (массива) используют операторы присваивания или процедуру ввода данных, как и для обычных переменных.

Например, присвоение значений элементам таблицы *Dist* можно описать на алгоритмическом языке следующим образом:

алг Ввод_расстояний

нач

вещ таб *Dist*[1:9]

Dist[1] := 0,39; *Dist*[2] := 0,72; *Dist*[3] := 1

Dist[4] := 1,52; *Dist*[5] := 5,20; *Dist*[6] := 9,54

Dist[7] := 19,18; *Dist*[8] := 30,07; *Dist*[9] := 39,44

кон

Зачастую ввод данных в массив или задачи их обработки связаны с перебором элементов массива. Для этих целей обычно используют циклы с параметром (со счетчиком). В теле цикла индекс массива обозначают переменной, и эта же переменная выступает в роли параметра (счетчика) цикла. Для параметра в заголовке цикла указывают перечень значений индекса массива. Например, если необходимо обработать все 20 значений массива T , то следует записать цикл нц для i от 1 до 20 ... кц и в теле цикла задать обработку для элемента $T[i]$.

Пример 10

Таблица A содержит 14 элементов, значения которых совпадают с их номерами. Записать алгоритм заполнения значений таблицы.

Решение. Из условия задачи следует, что значения элементов таблицы — целые, это необходимо учесть при ее описании. Затем нужно создать цикл с параметром (например, k), изменяющимся от 1 до 14 (количество элементов таблицы). В теле этого цикла каждому элементу $A[k]$ присвоить значение k .

```
алг Ввод
нач
  цел  $k$ 
  цел таб  $A[1:14]$ 
нц
  для  $k$  от 1 до 14
     $A[k] := k$ 
кц
кон
```

Пример 11

По данным о территории некоторых стран Европы (см. первую строку примера многомерной таблицы) рассчитать суммарную площадь этих стран.

Решение. В алгоритме необходимо предусмотреть ввод вещественных данных в таблицу (например, $Terr$) из 9 элементов. Для будущего суммарного результата необходимо отвести переменную (например, S), первоначально присвоить этой переменной нулевое значение. Затем организовать цикл с параметром, изменяющимся от 1 до 9 (количество элементов в таблице) и поочередно сложить предыдущее значение S с очередным элементом таблицы $Terr[i]$. Завершить алгоритм выводом результата.

```
алг Площадь
нач
  вещ таб  $Terr[1:9]$ 
   $Terr[1] := 83,8; Terr[2] := 30,5; Terr[3] := 244,8$ 
   $Terr[4] := 103; Terr[5] := 504,8; Terr[6] := 387$ 
   $Terr[7] := 337; Terr[8] := 551; Terr[9] := 450$ 
  вещ  $S$ 
  цел  $i$ 
   $S := 0$ 
нц
  для  $i$  от 1 до 9
     $S := S + Terr[i]$ 
кц
  вывод  $S$ 
кон
```

Пример 12

По данным о территории некоторых стран Европы (см. первую строку примера многомерной таблицы) определить самую большую площадь страны.

Решение. Запишем только тело алгоритма, поскольку заголовок уже приведен в предыдущем примере. Отведем переменную *SMax* для будущего значения максимальной площади. Сначала присвоим переменной *SMax* площадь первой из стран. Затем в цикле сравним значение *SMax* с площадью каждой из стран со 2-й по 9-ю. Если обнаружится большее значение, запомним теперь его в переменной *SMax*. По окончании работы цикла (когда будут проверены все значения площадей) в переменной *SMax* будет сохранено самое большое значение из имеющихся.

```
вещ SMax
цел i
SMax := Terr[1]
нц
  для i от 2 до 9
    если SMax < Terr[i]
      то SMax := Terr[i]
    всё
кц
вывод SMax
```

1.3.3. Логические значения, операции, выражения

Алгебра логики, логические высказывания

Наука, изучающая формы, методы и законы правильного мышления, называется *логикой*. Она интересуется не содержанием мышления, а его формой, поэтому ее часто называют еще *формальной логикой*.

Форма мышления — это способ выражения мыслей или форма, по которой они строятся.

Форма, обозначающая какой-либо объект или отличающий его признак, называется *понятием*. Примеры понятий: «компьютер», «планета», «длина», «профессия».

Форма, утверждающая или отрицающая что-либо о свойствах понятий и отношений между ними, называется *утверждением* (*высказыванием*, *суждением*). Примеры логических утверждений: «Декодирование — процесс восстановления информации из закодированного представления»; «В двоичной системе используются две цифры: 0 и 1»; «Париж — столица Франции». Утверждения могут быть истинными или ложными. Так, высказывание «Шанхай — столица Франции» является ложным утверждением.

Форма, в которой из двух или нескольких высказываний получают новое утверждение, называется *умозаключением*. Пример умозаключения: «Периферийные устройства компьютера — это устройства для ввода или вывода информации. Сканер — устройство для переноса текста и изображений с бумаги в компьютер. Следовательно, сканер — периферийное устройство».

Правила, которые должны соблюдаться, чтобы на основании истинных суждений получить истинные выводы, — это законы мышления. Логика изучает эти законы и способы получения новых утверждений на основании уже имеющихся.

Математическая логика использует для установления истинности или ложности высказываний математические методы. Она пользуется специальным символьным языком, подобным языку математики, поэтому ее часто называют *символьной логикой*.

Алгебра логики — раздел математической логики, в котором методы алгебры используются в логических преобразованиях. Она изучает логические высказывания и методы установления их истинности или ложности с помощью алгебраических методов.

Логическое высказывание — это любое повествовательное предложение, в отношении которого можно однозначно утверждать, что его содержание истинно или ложно. Вопросительные и повелительные предложения не являются логическими высказываниями. Но и не каждое повествовательное предложение является логическим высказыванием. Например, суждение «Лето было очень дождливым» не является однозначным, для утверждения «Существует несколько Вселенных» нельзя однозначно определить истинность; поэтому такие предложения не являются логическими высказываниями (утверждениями).

Таким образом, отличительной особенностью логических высказываний является возможность принимать одно из двух значений — *истина* и *ложь*. Истинность или ложность высказывания определяется вне алгебры логики — с помощью наблюдений, научных исследований, практических опытов и т. п.

В алгебре логики различают простые высказывания и сложные (составные), составленные из нескольких простых. Если в высказывании нельзя выделить некую часть, которая не совпадает по смыслу с исходным высказыванием и сама является высказыванием, то оно называется *простым высказыванием*. *Простые высказывания* обычно обозначаются латинскими буквами А, В, С и т. д.

Сложные высказывания представляют собой объединение простых высказываний с помощью **логических связок**. В качестве логических связок используются слова «не», «и», «или», «тогда и только тогда», «если ... то». Истинность или ложность получаемых таким образом сложных высказываний определяется значением простых высказываний. Например, из простых высказываний «Офис фирмы находится в Мадриде» и «Офис фирмы находится в Берлине» можно составить сложные: «Офис фирмы находится в Мадриде или Берлине», «Офис фирмы находится в Мадриде и Берлине», «Если офис фирмы находится в Мадриде, то он находится в Берлине». Истинность первого из них означает, что офис фирмы находится в одном из названных городов или же имеются офисы в обоих городах. Ложность его означает, что ни в одном из этих городов офиса нет. Второе составное утверждение истинно тогда, когда в обоих городах имеется офис фирмы. Если же офис существует только в Берлине или только в Мадриде, — второе составное высказывание ложно.

В классической, двузначной алгебре логики логических значений всего два: *истина* (*True*) и *ложь* (*False*). Им соответствует цифровое представление — 1 и 0. Иногда эти значения записывают как «да» и «нет». Факт истинности или ложности некоторого высказывания А записывают соответственно как $A = 1$ или $A = 0$.

Логические операции

В алгебре логики логические связки рассматриваются как логические операции. Они имеют свои названия и обозначения. Результаты применения каждой операции к логическим высказываниям (истинным или ложным) можно представить в виде таблицы. В ней указывают все возможные сочетания значений исходных логических высказываний и истинность или ложность результата. Такие таблицы называют **таблицами истинности** операции. Обычно в них используют обозначения логических значений 0 и 1 или *ложь* и *истина*.

Основные логические операции — отрицание, конъюнкция, дизъюнкция, исключающая дизъюнкция, следование, эквивалентность.

Логическое отрицание (инверсия) — логическая операция, в результате которой из данного высказывания получается новое высказывание — *отрицание исходного*. Обозначается символически чертой сверху (\bar{A}) или условными обозначениями $\neg A$, *not A*, *не A* (читается «отрицание A», «не A», «A ложно», «неверно, что A»).

Высказывание $\neg A$ ложно, когда A истинно, и истинно, когда A ложно.

Таблица истинности операции отрицания

A	$\neg A$
<i>истина</i>	<i>ложь</i>
<i>ложь</i>	<i>истина</i>

или

A	$\neg A$
1	0
0	1

Если обозначить через A высказывание «Арбуз является ягодой», то $\neg A$ соответствует высказыванию «Арбуз не является ягодой» («Неверно, что арбуз — ягода»).

Отрицание является унарной операцией. *Унарная (одноместная) операция* — это операция, которая применяется к одному операнду.

Остальные логические операции являются двуместными (бинарными). *Бинарная (двуместная) операция* — это операция, которая выполняется над двумя операндами.

Логическое умножение (конъюнкция) — операция, соединяющая два или более высказываний при помощи связки «и». Эта связка символически обозначается с помощью знака \wedge и читается «A и B». Для обозначения конъюнкции также применяются знаки: $A \cdot B$, $A \& B$, $A \text{ и } B$, $A \text{ and } B$, а иногда между высказываниями не ставится никакого знака: AB .

Высказывание $A \wedge B$ истинно только тогда, когда оба высказывания A и B истинны. Высказывание $A \wedge B$ ложно только тогда, когда ложно хотя бы одно из высказываний A или B.

Таблица истинности операции конъюнкции

A	B	$A \wedge B$
<i>истина</i>	<i>ложь</i>	<i>ложь</i>
<i>ложь</i>	<i>истина</i>	<i>ложь</i>
<i>ложь</i>	<i>ложь</i>	<i>ложь</i>
<i>истина</i>	<i>истина</i>	<i>истина</i>

или

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

Например, высказывания «Лондон расположен севернее Лиссабона» и «Лондон расположен восточнее Лиссабона» истинны. Тогда истинным будет и составное логическое высказывание «Лондон расположен севернее и восточнее Лиссабона». Высказывания «Лондон расположен не севернее и восточнее Лиссабона», «Лондон расположен севернее и не восточнее Лиссабона», «Лондон расположен не севернее и не восточнее Лиссабона» — ложны.

Логическое сложение (дизъюнкция) — операция, соединяющая два или более высказываний при помощи связки «или». Эта связка символически обозначается с помощью знака \vee и читается « A или B ». Для обозначения дизъюнкции также применяются знаки: $A + B$, A или B , A or B , $A \mid B$.

Высказывание $A \vee B$ истинно только тогда, когда хотя бы одно из высказываний A или B истинно. Высказывание $A \vee B$ ложно только тогда, когда оба высказывания A и B ложны.

Таблица истинности операции дизъюнкции

А	В	$A \vee B$
<i>истина</i>	<i>ложь</i>	<i>истина</i>
<i>ложь</i>	<i>истина</i>	<i>истина</i>
<i>ложь</i>	<i>ложь</i>	<i>ложь</i>
<i>истина</i>	<i>истина</i>	<i>истина</i>

или

А	В	$A \vee B$
1	0	1
0	1	1
0	0	0
1	1	1

Например, высказывания «Виктор старше Ольги» и «Виктор — однофамилец Ольги» истинны. Тогда истинными будут и составные логические высказывания «Виктор старше Ольги или Виктор — однофамилец Ольги», «Виктор младше Ольги или Виктор — однофамилец Ольги», «Виктор старше Ольги или Виктор — не однофамилец Ольги». Высказывание «Виктор младше Ольги или Виктор — не однофамилец Ольги» — ложно, поскольку ложны оба составляющие его простые высказывания.

Исключающее сложение (исключающая дизъюнкция, строгая дизъюнкция, сложение по модулю два, дизъюнкция строго-разделительная) — логическая операция, соединяющая два высказывания при помощи связки «или», употребленной в исключаящем смысле (называется также *исключающее «или»*). Операция символически обозначается с помощью знака \oplus и читается «либо A , либо B ».

Высказывание $A \oplus B$ истинно только тогда, когда высказывания A и B имеют различные значения.

Таблица истинности операции строгой дизъюнкции

А	В	$A \oplus B$
<i>истина</i>	<i>ложь</i>	<i>истина</i>
<i>ложь</i>	<i>истина</i>	<i>истина</i>
<i>ложь</i>	<i>ложь</i>	<i>ложь</i>
<i>истина</i>	<i>истина</i>	<i>ложь</i>

или

А	В	$A \oplus B$
1	0	1
0	1	1
0	0	0
1	1	0

Например, результат исключаяющей дизъюнкции двух высказываний «Виктор не старше Ольги» и «Виктор младше Ольги» всегда будет истиной, кто бы из них не был старше.

Логическое следование (импликация) — логическая операция, соединяющая два высказывания при помощи связки «если... то» в сложное высказывание. Операция символически обозначается с помощью знака \rightarrow и читается «Если A , то B », « A влечет B », «из A следует B », « A имплицирует B ». Для обозначения импликации применяются также знаки \supset или \Rightarrow . Первое логическое высказывание является *условием* (посылкой), а второе — *следствием* (заключением).

Для операции импликации справедливо утверждение, что из лжи может следовать все что угодно, а из истины — только истина. Таким образом, импликация $A \rightarrow B$ ложна только тогда, когда A истинно, а B ложно (из истинного высказывания следует ложное). Во всех остальных случаях импликация истинна.

Таблица истинности операции импликации

А	В	$A \rightarrow B$
<i>истина</i>	<i>ложь</i>	<i>ложь</i>
<i>ложь</i>	<i>истина</i>	<i>истина</i>
<i>ложь</i>	<i>ложь</i>	<i>истина</i>
<i>истина</i>	<i>истина</i>	<i>истина</i>

или

А	В	$A \rightarrow B$
1	0	0
0	1	1
0	0	1
1	1	1

Для высказываний «Луна — спутник Земли» и «Сумма углов треугольника не равна 180° » (первое истинно, второе ложно) составное высказывание «Если Луна — спутник Земли, то сумма углов треугольника не равна 180° » будет ложным. Однако истинными будут высказывания «Если Луна — спутник Земли, то сумма углов треугольника равна 180° », «Если Луна — не спутник Земли, то сумма углов треугольника не равна 180° » и «Если Луна — не спутник Земли, то сумма углов треугольника равна 180° ». Этот пример наглядно демонстрирует, что в алгебре логики смысл высказываний не учитывается, а рассматриваются только их истинность или ложность.

Логическое равенство (эквивалентность, следование, двойная импликация, равнозначность) — логическая операция, позволяющая из двух высказываний A и B получить новое высказывание $A \equiv B$ (читается « A эквивалентно B »). Эта операция может быть выражена связками «тогда и только тогда», «необходимо и достаточно», «равносильно». Для обозначения эквивалентности применяются знаки \sim , \Leftrightarrow .

Если оба высказывания имеют различные логические значения, результатом операции эквивалентности всегда будет ложь. Если же оба простые высказывания ложны или оба истинны, то составное логическое высказывание всегда будет истинно.

Таблица истинности операции эквивалентности

А	В	$A \sim B$
<i>истина</i>	<i>ложь</i>	<i>ложь</i>
<i>ложь</i>	<i>истина</i>	<i>ложь</i>
<i>ложь</i>	<i>ложь</i>	<i>истина</i>
<i>истина</i>	<i>истина</i>	<i>истина</i>

или

А	В	$A \sim B$
1	0	0
0	1	0
0	0	1
1	1	1

Для высказываний «Линейное уравнение всегда имеет решение» и «Кит — млекопитающее» их эквивалентность всегда будет истиной, так как оба простые утверждения истинны.

Таким образом, сводная таблица истинности для всех основных логических операций имеет вид:

Логические переменные		Логические операции					
		отрицание	конъюнкция	дизъюнкция	исключающая дизъюнкция	следование	эквивалентность
А	В	$\neg A$	$A \wedge B$	$A \vee B$	$A \oplus B$	$A \rightarrow B$	$A \sim B$
1	0	0	0	1	1	0	0
0	1	1	0	1	1	1	0
0	0	1	0	0	0	1	1
1	1	0	1	1	0	1	1

Логические выражения

Логические высказывания могут быть записаны в виде формул (логических выражений). **Логические выражения** включают логические переменные, знаки логических операций, логические константы (*истина* и *ложь*) и скобки. Логические выражения принимают значения *истина* или *ложь*.

Правила построения логических выражений:

- любая логическая переменная или константа (*истина* и *ложь*) являются логическим выражением;
- если A — , то $\neg A$ — тоже логическое выражение;
- если A и B — логические выражения, то $A \wedge B$; $A \vee B$; $A \oplus B$; $A \rightarrow B$; $A \sim B$ — тоже логические выражения.

Например, $A \oplus \text{истина} \vee B \vee \text{ложь}$ — логическое выражение; $A \vee \oplus B \vee \text{ложь}$ не является логическим выражением.

Логическое выражение, принимающее значение *истина* при любых значениях входящих в него переменных, называется *тождественно-истинным выражением (тавтологией)*. Например, $A \vee B \vee \neg A$; $(A \wedge \neg A) \rightarrow B$.

Логическое выражение, принимающее значение *ложь* при любых значениях входящих в него переменных, называется *тождественно-ложным выражением* (*противоречием*). Например, $A \wedge \neg A$; $B \sim \neg B$.

Логическое выражение, принимающее как значение *ложь*, так и значение *истина* при разных значениях входящих в него переменных, называется *выполнимым выражением*.

ПРИОРИТЕТ ЛОГИЧЕСКИХ ОПЕРАЦИЙ

Для сложных логических выражений, содержащих несколько логических операций, определен порядок выполнения действий (*приоритет*): сначала операции отрицания, затем операции логического умножения, потом операции логического сложения и исключающего сложения, последними выполняются операции импликации и эквивалентности. Операции выполняются слева направо. Порядок выполнения может быть изменен с помощью скобок.

Приоритет выполнения логических операций в логических выражениях

1	\neg отрицание
2	\wedge конъюнкция
3	\vee дизъюнкция; \oplus исключающая дизъюнкция
4	\rightarrow следование; \sim эквивалентность

Пример 1

Определить порядок выполнения логических операций в выражении. Вычислить его значение, если $A = 1$, $B = 0$, $C = 1$.

$$\neg A \wedge C \vee (A \oplus B) \wedge B$$

Решение. Первыми вычисляются значения в скобках. Затем выполняются операции по приоритетам: самый высокий приоритет имеет операция отрицания, после нее, как в математике, следуют операции умножения, а затем сложения. Таким образом, порядок будет следующий:

$$\begin{array}{cccccc} & 2 & 3 & 5 & 1 & 4 \\ \neg A \wedge C \vee (A \oplus B) \wedge B \end{array}$$

Тогда значение выражения $\neg 1 \wedge 1 \vee (1 \oplus 0) \wedge 0$ после вычисления отрицания и выражения в скобках: $0 \wedge 1 \vee 1 \wedge 0$, после операций умножения: $0 \vee 0$. Итог: 0.

Ответ: логическое выражение ложно.

Пример 2

Для каких из приведенных слов истинно следующее высказывание?

(Вторая буква гласная) ИЛИ (Первая буква гласная) И НЕ (Длина имени не больше 5 букв)

- 1) Олег 2) Марианна 3) Светлана 4) Ольга.

Решение. 1. В соответствии с приоритетом выполнения операций сначала нужно вычислить результат операции отрицания: «Длина имени больше 5 букв». Такие имена — *Марианна, Светлана*.

2. Затем необходимо рассчитать результат конъюнкции (связка И) второго и третьего высказываний. Конъюнкция истинна только тогда, когда истинны оба высказывания. Следовательно, нужно выбрать имена, в которых и первая буква гласная, и длина имени больше 5 букв — таких имен среди вариантов нет.

3. Результат дизъюнкции (связка ИЛИ) истинен только тогда, когда истинно или одно, или второе, или одновременно оба простые высказывания. Первое простое высказывание истинно для имени *Марианна*, второе ложно для указанных имен. Таким образом, верный вариант ответа — 2.

Ответ: 2) Марианна.

СВЯЗЬ МЕЖДУ ЛОГИЧЕСКИМИ ОПЕРАЦИЯМИ

Между логическими операциями существует взаимосвязь. Операции исключающего «или», следования и эквивалентности можно выразить через операции отрицания, логического сложения и умножения, что отражено в следующей таблице.

Связь между логическими операциями

Исключающая дизъюнкция	$A \oplus B$	$(\bar{A} \wedge B) \vee (A \wedge \bar{B})$
Следование	$A \rightarrow B$	$\bar{A} \vee B$
Эквивалентность	$A \sim B$	$(\bar{A} \wedge \bar{B}) \vee (A \wedge B)$

Поэтому операции отрицания, логического сложения и логического умножения называют *основными логическими операциями*: их достаточно, чтобы построить любое логическое выражение.

Если логическое выражение содержит только операции отрицания, логического сложения и логического умножения, говорят, что выражение находится в *нормальной форме*.

РАВНОСИЛЬНЫЕ ЛОГИЧЕСКИЕ ВЫРАЖЕНИЯ

Логические выражения, значения которых совпадают для всех наборов входящих в них переменных, называются *равносильными*, или *эквивалентными*.

Чтобы убедиться, что два выражения равносильны, можно построить для них таблицы истинности. Если в таблицах совпадут все значения, значит, выражения равносильны.

Пример 3

Проверить равносильность выражений $A \sim B$ и $(\bar{A} \wedge \bar{B}) \vee (A \wedge B)$.

Решение. Для проверки следует создать таблицу истинности, содержащую столько строк, сколько возможно наборов значений переменных, входящих в выражение. Для двух переменных (A и B) количество наборов равно четырём. К двум столбцам для значений переменных (A и B) нужно присовокупить количество столбцов, равное количеству операций в выражении. Таким образом, необходимо создать таблицу, содержащую 4 строки и 7 столбцов.

Заполним первые 2 столбца (A и B) всеми сочетаниями значений переменных. Запишем в качестве заголовков столбцов все операции выражения в порядке их выполнения (в соответствии с приоритетами и скобками). Рассчитаем значения этих операций: сначала выражения в скобках, затем результат их сложения.

A	B	$\neg A$	$\neg B$	$\neg A \wedge \neg B$	$A \wedge B$	$(\neg A \wedge \neg B) \vee (A \wedge B)$
1	0	0	1	0	0	0
0	1	1	0	0	0	0
0	0	1	1	1	0	1
1	1	0	0	0	1	1

Последний столбец содержит результирующее значение выражения. Он совпадает с таблицей истинности для операции эквивалентности. Следовательно, выражения равносильны.

Основные законы алгебры логики

Для сложных логических выражений с большим числом переменных определение их истинности путем построения таблиц истинности становится громоздким. В таких случаях применяют способы упрощения выражений. Под упрощением понимают равносильное преобразование выражения к его нормальной форме.

Нормальная форма выражения содержит только операции отрицания, конъюнкции и дизъюнкции и не содержит отрицания выражений и двойных отрицаний.

Для упрощения используют равносильные преобразования, которые иначе называют **основными законами алгебры логики**.

Тождественные преобразования логических выражений

	Закон	Для дизъюнкции \vee	Для конъюнкции \wedge
1	переместительный (закон коммутативности)	$A \vee B = B \vee A$	$A \wedge B = B \wedge A$
2	сочетательный (закон ассоциативности)	$A \vee (B \vee C) = (A \vee B) \vee C$	$A \wedge (B \wedge C) = (A \wedge B) \wedge C$
3	распределительный (закон дистрибутивности)	$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$	$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$
4	де Моргана	$\overline{A \vee B} = \overline{A} \wedge \overline{B}$	$\overline{A \wedge B} = \overline{A} \vee \overline{B}$
5	идемпотентности	$A \vee A = A$	$A \wedge A = A$
6	поглощения	$A \vee (A \wedge B) = A$	$A \wedge (A \vee B) = A$
7	склеивания	$(A \wedge B) \vee (\overline{A} \wedge B) = B$	$(A \vee B) \wedge (\overline{A} \vee B) = B$
8	исключенного третьего	$A \vee \overline{A} = 1$	
9	непротиворечия		$A \wedge \overline{A} = 0$
10	исключения констант	$A \vee 0 = A \quad A \vee 1 = 1$	$A \wedge 1 = A \quad A \wedge 0 = 0$
11	двойного отрицания	$\overline{\overline{A}} = A$	

Для всех тождественных преобразований выполняется **закон двойственности**: если в формуле преобразования заменить конъюнкцию на дизъюнкцию, дизъюнкцию — на конъюнкцию, значения 1 — на 0, 0 — на 1, то закон, сформулированный для конъюнкции, примет форму аналогичного закона для дизъюнкции, и наоборот.

Прежде всего при равносильных преобразованиях избавляются от отрицания выражений, потом — от логических операций исключаящей дизъюнкции, следования и эквивалентности. Затем используют законы алгебры логики для уменьшения количества переменных в выражении.

Пример 4

Выбрать выражение, которое равносильно выражению $(A \wedge B) \vee (\bar{A} \wedge B)$.

- 1) A 2) $A \wedge B$ 3) $\bar{A} \wedge B$ 4) B

Решение. В соответствии с законом склеивания $(A \wedge B) \vee (\bar{A} \wedge B) = B$, следовательно, исходное выражение равносильно выражению B .

Ответ: 4) B .

ОПРЕДЕЛЕНИЕ ЗНАЧЕНИЙ ЛОГИЧЕСКИХ ВЫРАЖЕНИЙ

Выражения, которые принимают логические значения (*истина* или *ложь*) в результате выполнения операций сравнения (*больше* $>$, *меньше* $<$, *больше или равно* \geq , *меньше или равно* \leq , *равно* $=$, *не равно* \neq), также являются логическими выражениями. Кроме операций сравнения и логических операций такие выражения могут включать функции и алгебраические операции. Приоритет выполнения этих операций таков:

1. Вычисление значений функций.
2. Выполнение алгебраических операций (вначале возведение в степень, затем умножение и деление, после чего вычитание и сложение).
3. Выполнение операций сравнения (в порядке записи).
4. Выполнение логических операций (сначала операции отрицания, затем операции логического умножения, потом операции логического сложения, последними выполняются операции импликации и эквивалентности).

Если в логическом выражении используются скобки, то сначала выполняются заключенные в них операции.

Пример 5

Найти значение выражения:

$$y > 0 \vee A \vee \frac{|y - 6x|}{3} < \pi \wedge \neg(x^2 + y^3 > 2x \wedge \overline{A \vee B}) \wedge \neg B$$

для $x = 1$, $y = 5$, $A = \text{истина}$, $B = \text{ложь}$.

Решение. Подставим значения в выражение и вычислим его результат в соответствии с порядком операций.

1) $x^2 + y^3 > 2x$. Результат подстановки значений: $1^2 + 5^3 > 2$, т. е. $126 > 2$. Значение — *истина*.

2) $\overline{A \vee B}$. Результат подстановки в $A \vee B$: *истина* \vee *ложь*. Значение — *истина*. Значение $A \vee B$ — *ложь*.

Следовательно, значение выражения $(x^2 + y^3 > \wedge 2x \overline{A \vee B})$ равно

$$\text{истина} \wedge \text{ложь} = \text{ложь}.$$

3) $y > 0$. Результат подстановки: $5 > 0$. Значение — истина.

4) $\frac{|y - 6x|}{3} < \pi$. Результат подстановки: $\frac{|5 - 6|}{3} < \pi$. Значение — истина.

После этих вычислений получим:

$$\text{истина} \vee A \vee \text{истина} \wedge \neg \text{ложь} \wedge \neg B$$

Теперь надо выполнить сначала операции отрицания, затем — логического умножения и сложения.

5) $\neg \text{ложь}$. Значение — истина.

$\neg B$. Результат подстановки: $\neg \text{ложь}$. Значение — истина.

6) $\text{истина} \wedge \text{истина} \wedge \text{истина}$. Значение — истина.

7) $\text{истина} \vee A \vee \text{истина}$. Значение — истина.

Ответ: истина.

Пример 6

Для какого из приведенных ниже значений числа M истинно следующее выражение?

$$\neg M \geq 10 \wedge M > 3$$

1) 1

2) 2

3) 3

4) 4

Решение. В соответствии с приоритетами операций сначала следует выполнить операции сравнения, затем отрицания, а потом — конъюнкцию. Отрицанием высказывания $M \geq 10$ является высказывание $M < 10$. Получим выражение $M < 10 \wedge M > 3$. Для того чтобы это выражение (конъюнкция) было истинным, должны выполняться (т. е. быть истинными) оба неравенства. Следовательно, значение M должно быть больше 3, но меньше 10. Среди предложенных значений этому условию удовлетворяет только одно — число 4.

Ответ: 4) 4.

Задачи, подобные предыдущему примеру, можно решать и с помощью таблиц истинности.

Пример 7

Для какого из приведенных ниже значений числа M истинно следующее выражение?

$$\neg M \geq 10 \wedge M > 3$$

1) 1

2) 2

3) 3

4) 4

Решение. Составим таблицу истинности: все операции выражения укажем в столбцах таблицы, все предложенные значения M укажем в ее строках. Рассчитаем значения таблицы:

M	$M \geq 10$	$M > 3$	$\neg M \geq 10$	$\neg M \geq 10 \wedge M > 3$
1	0	0	1	0
2	0	0	1	0
3	0	0	1	0
4	0	1	1	1

Последний столбец содержит результат всего выражения. Истинным оно будет только для значения числа M , равного 4.

Ответ: 4) 4.

Пример 8

Для какого из указанных ниже значений числа M истинно следующее выражение?

$$5 \cdot M > 10 \wedge \neg(M + 2 < 9) \vee \neg(M > 0)$$

- 1) 1 2) 2 3) 6 4) 10

Решение. 1. Преобразуем все неравенства в выражении так, чтобы их левые части содержали только переменную M , получим: $M > 2 \wedge \neg(M < 7) \vee \neg(M > 0)$.

2. В соответствии с приоритетом выполнения операций выполним логическое отрицание: $M > 2 \wedge M \geq 7 \vee M \leq 0$.

3. Затем выполним операцию логического умножения (конъюнкцию) двух первых простых высказываний. Ее результат истинен только тогда, когда истинны оба простых высказывания. Это возможно только при $M \geq 7$.

4. Наконец, последней выполним дизъюнкцию — для ее истинности необходимо, чтобы хотя бы один из ее операндов был истинным: $M \geq 7, M \leq 0$. В предложенных ответах все числа положительные, значит, ответ 10.

Ответ: 4) 10.

Пример 9

В табличной форме представлены ежемесячные данные о продаже групп товаров за полгода. Сколько групп товаров демонстрировали рост продаж в весенние месяцы или вышли на уровень свыше 80% в июне?

Продажи, %								
№	Категория товаров	январь	февраль	март	апрель	май	июнь	июль
1.	Кондитерские изделия	85	67	76	61	44	53	91
2.	Молочные продукты	63	62	57	69	48	44	72
3.	Мясо/птица	81	73	67	99	98	51	41
4.	Напитки	41	45	66	96	66	81	56
5.	Приправы	73	95	60	99	73	43	51
6.	Рыбопродукты	62	50	75	70	86	90	70
7.	Фрукты	77	65	47	63	92	69	70
8.	Хлебобулочные изделия	62	45	74	65	81	70	45

Решение. Переформулируем условие задачи: необходимо найти группы товаров, для которых $(\text{Март} < \text{Апрель}) \wedge (\text{Апрель} < \text{Май}) \vee (\text{Июнь} > 80)$.

Введем обозначения:

$A = (\text{Март} < \text{Апрель})$

$B = (\text{Апрель} < \text{Май})$

$C = (\text{Июнь} > 80)$

Тогда выражение можно записать как $A \wedge B \vee C$.

Логическое выражение состоит из одной конъюнкции и одной дизъюнкции. Значение выражения конъюнкции истинно только тогда, когда истинны оба составляющие его простых выражения ($(\text{Март} < \text{Апрель})$ и $(\text{Апрель} < \text{Май})$). Значение выражения дизъюнкции будет истинным, если хотя бы одно из составляющих его простых высказываний будет истинным.

Составим таблицу истинности для исходных данных.

№	Категория товаров	март	апрель	май	июнь	A	B	C	$A \wedge B$	$A \wedge B \vee C$
1.	Кондитерские изделия	76	61	44	53	0	0	0	0	0
2.	Молочные продукты	57	69	48	44	1	0	0	0	0
3.	Мясо/птица	67	99	98	51	1	0	0	0	0
4.	Напитки	66	96	66	81	1	0	1	0	1
5.	Приправы	60	99	73	43	1	0	0	0	0
6.	Рыбпродукты	75	70	86	90	0	1	1	0	1
7.	Фрукты	47	63	92	69	1	1	0	1	1
8.	Хлебобулочные изделия	74	65	81	70	0	1	0	0	0

Логическому выражению удовлетворяют 3 записи — 4-я, 6-я и 7-я.

Ответ: 3.

1.3.4. Разбиение задачи на подзадачи, вспомогательный алгоритм

При создании алгоритмов и программ следует придерживаться некоторых определенных принципов работы, которые помогают рационально организовать работу, сделать алгоритмы и программы наглядными и понятными. Использование *структурного подхода* к программированию позволяет достичь этих целей, а также облегчает внесение изменений в алгоритмы и программы в случае такой необходимости.

Структурный подход предполагает составление алгоритмов и программ методом «сверху вниз» (нисходящее программирование). При этом задача сначала рассматривается в целом, затем разбивается на крупные блоки, те в свою очередь — на блоки поменьше и т. д. Эти блоки должны быть логически завершенными. Процесс разбиения продолжается до тех пор, пока задача не будет представлена в виде ряда простых задач, для которых легко составить алгоритм. Такой метод называется *методом пошаговой детализации*. Он предполагает постепенный переход от более общих задач к более частным и к их подзадачам.

Впоследствии алгоритмические блоки могут быть использованы в составе других алгоритмов, при решении других задач.

Алгоритм, который решает некоторую подзадачу из основной задачи и используется целиком в составе других алгоритмов, называется **вспомогательным алгоритмом**.

Такой подход облегчает решение сложных задач. Алгоритм решения поставленной задачи разделяется на основной алгоритм и вспомогательные.

Зачастую вспомогательный алгоритм выполняется многократно, отличаясь значениями параметров. Такой вспомогательный алгоритм, записанный на языке программирования, называют **подпрограммой**. Подпрограммы могут содержать обращения к другим подпрограммам.

При пошаговой детализации сначала реализуется (программируется) основной алгоритм, а вместо вспомогательных записывают сначала обращения к будущим подпрограммам. Сами подпрограммы составляются позже.

Существует и иной подход: сначала составляется множество подпрограмм, реализующих различные частные задачи, а затем создается основная программа, содержащая их вызовы. Обычно такие подпрограммы объединяют в *библиотеку подпрограмм*, которая может пополняться с течением времени. Такой подход называют методом программирования «снизу вверх».

Подпрограмма, так же как и основная программа, должна иметь свое (уникальное) имя. Подпрограммы имеют такую же структуру, что и основная программа. Отличия заключаются в оформлении заголовка подпрограммы: кроме имени (идентификатора) он содержит еще и перечень параметров, передаваемых из основной программы во вспомогательную. Такой заголовок называется *списком формальных параметров* и в общем случае не обязателен.

При вызове из основной программы некоторой подпрограммы указывают ее имя и перечень фактических значений, соответствующих ее формальным параметрам. При этом должны совпадать количество и типы соответствующих друг другу формальных и фактических параметров (по порядку следования в заголовке подпрограммы).

Подпрограмма может быть реализована в виде *функции* или *процедуры*. Различия между ними — в способах вызова (использования) и передачи вспомогательному алгоритму параметров. Подпрограмма-функция выработывает результат — вычисляет некоторое единственное значение, которое присваивается ее имени (идентификатору).

Одна программа может содержать несколько подпрограмм, а те, в свою очередь, могут включать свои подпрограммы, которые называются *вложенными*. Описание подпрограмм располагают внутри основной программы в ее описательной части.

Подпрограммы могут быть *стандартными* (входящими в состав языка программирования, чаще всего в виде библиотек подпрограмм) и *пользовательскими* — созданными непосредственно пользователем.

► Пример 1

Создать вспомогательный алгоритм вычисления длины отрезка по известным координатам его концов. Воспользоваться этим алгоритмом при решении задачи расчета периметра треугольника, заданного его вершинами.

Решение. Длина отрезка может быть рассчитана по формуле $l = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$, где (x_1, y_1) и (x_2, y_2) — координаты концов этого отрезка.

Таким образом, алгоритм будет иметь 5 параметров: 4 координаты точек и переменную для будущего результата. Обозначим рассчитываемую длину через l .

В заголовке алгоритма перечислим все 5 параметров, указав для них тип данных. В последующих строках перечислим отдельно входные параметры (аргументы, служебное слово **арг**) и выходной параметр (результат, служебное слово **рез**). Тело алгоритма будет содержать только одну команду — расчета длины отрезка.

```
алг Segment (вещ  $x_1, y_1, x_2, y_2, l$ )
  арг  $x_1, y_1, x_2, y_2$ 
  рез  $l$ 
нач
   $l := ((x_1 - x_2)^2 + (y_1 - y_2)^2)^{1/2}$ 
кон
```

В этом вспомогательном алгоритме отсутствуют команды ввода исходных и вывода результирующих данных. Это будет организовано в основном алгоритме. Например, для расчета длины отрезка d с координатами вершин $(0, 0)$ и $(2, 5)$ в основной программе надо будет указать команду `Segment(0, 0, 2, 5, d)`.

Составим основной алгоритм расчета периметра треугольника P с вершинами (a_1, b_1) , (a_2, b_2) и (a_3, b_3) . Для этого необходимо будет вычислить длины отрезков (l_1, l_2, l_3) трижды. Поэтому основной алгоритм будет трижды вызывать созданный вспомогательный *Segment*. Каждый раз при вызове будут указываться координаты вершин очередного отрезка и переменная для сохранения его длины.

Следует отметить, что если алгоритм будет записываться на языке программирования, необходимо будет описать используемый вспомогательный алгоритм (подпрограмму) в описательной (заголовочной) части основной программы.

```
алг Perimeter
нач
  вещ  $a_1, b_1, a_2, b_2, a_3, b_3$ 
  вещ  $l_1, l_2, l_3, P$ 
  ввод  $a_1, b_1, a_2, b_2, a_3, b_3$ 
  Segment( $a_1, b_1, a_2, b_2, l_1$ )
  Segment( $a_1, b_1, a_3, b_3, l_2$ )
  Segment( $a_2, b_2, a_3, b_3, l_3$ )
   $P := l_1 + l_2 + l_3$ 
  вывод  $P$ 
кон
```

1.3.5. Обрабатываемые объекты: цепочки символов, числа, списки, деревья

Одна из задач информатики — поиск таких форм представления информации, которые было бы удобно обрабатывать компьютеру. Для этого каждый объект представляется как некоторая абстрактная структура данных. **Абстрактная структура данных** описывает свойства и характеристики объекта, взаимосвязи его элементов и операции, которые могут быть произведены над классами таких объектов.

К базовым абстрактным структурам данных, которые используются в информатике, относятся:

- целые и вещественные числа,
- символы,
- логические значения.

Для их обработки в языках программирования существуют соответствующие *типы данных*. Для хранения данных создаются *переменные*. Конкретные данные выступают в качестве *значения переменной*. Каждая переменная однозначно определяется своим *именем (идентификатором)*. Тип данных, к которому относится переменная, говорит о том, какие действия можно производить над хранящимися в переменной данными.

Из базовых типов можно строить более сложные структуры (**составные типы** любой сложности). Существуют различные методы объединения простых типов в составные. Они разным образом задают правила построения структуры данных:

- каким образом эта структура организована (*логическая организация* структуры) — какие более элементарные объекты входят в ее состав и какие операции можно над ними выполнять;
- перечень операций, которые можно выполнять над структурой в целом.

Языки программирования описывают также и *физическое представление* структуры: каким образом данные будут храниться в памяти компьютера, каким образом будут кодироваться операции и т. п.

Таким образом, для объединения базовых структур в составную необходимо указать правила доступа к отдельным элементам этой структуры и операции над структурой в целом.

Основные методы конструирования **составных структур** позволяют создать такие объекты, как *таблицы, векторы, строки, последовательности*. Они называются **статическими**, для них обычно в большинстве языков программирования имеются соответствующие типы данных — одномерные и двумерные *массивы, строки, файлы, записи, множества*.

Более сложные структуры программисту предстоит создавать в программах самостоятельно, используя базовые структуры данных. Обычно их размер изменяется в ходе работы программы. Такие структуры называются **динамическими**. К ним обычно относят *списки, деревья (графы)* и др.

Правильный выбор структуры для решения задачи во многом определяет эффективность ее решения.

Цепочки символов

► Пример 1

Имеется некоторый алгоритм, который преобразует одну цепочку символов в другую по таким правилам: вычисляется длина исходной цепочки; значение суммы приписывается в конец этой цепочки. Затем в полученной цепочке каждая цифра, кроме 0, заменяется меньшей на единицу (9 заменяется на 8, 8 — на 7 и т. д.). Полученная последовательность символов записывается в обратном порядке.

Получившаяся цепочка символов и есть результат работы алгоритма. Например, если первоначально имелась цепочка символов 9ВАЛ, то результатом работы алгоритма будет ЗЛАВ8.

Какая последовательность символов получится, если применить к исходной цепочке ALL4U этот алгоритм дважды (то есть сначала применить алгоритм к исходной цепочке, а затем применить его к полученному результату)?

Решение. Выполним этот алгоритм дважды по шагам:

- 1) Вычислим длину цепочки: 5 символов.
- 2) Припишем длину в конец: ALL4U5.
- 3) Заменяем цифры на меньшие (5 — на 4, 4 — на 3): ALL3U4.
- 4) Перепишем цепочку в обратном порядке: 4U3LLA.

Применим алгоритм второй раз:

- 5) Вычислим длину цепочки: 6 символов.
- 6) Припишем длину в конец: 4U3LLA6.
- 7) Заменяем цифры на меньшие (6 — на 5, 4 — на 3, 3 — на 2): 3U2LLA5.
- 8) Перепишем цепочку в обратном порядке: 5ALL2U3.

Ответ: 5ALL2U3.

Списки

Список — это последовательность связанных элементов, для которых определены операции добавления элементов в произвольное место списка и удаление любого элемента. Список задается указателем на начало списка. Обычно элементы списка называют **узлами** списка.

Типовые операции над элементами списка:

- получение доступа к определенному узлу (обход списка);
- добавление узла перед или после указанного элемента;
- удаление заданного узла;
- поиск заданного элемента;
- объединение нескольких списков в один;
- разбиение одного списка на несколько списков;
- копирование списка и др.

Все элементы списка упорядочены. Для каждого узла, кроме первого, существует *предыдущий* узел; для каждого узла, кроме последнего, есть *следующий* узел. Обойти все элементы списка можно, только двигаясь последовательно от текущего элемента к следующему начиная с первого. Непосредственный доступ к произвольному элементу невозможен. Поэтому обработка такого односвязного списка не всегда удобна, поскольку отсутствует возможность движения в противоположную сторону — от конца списка к началу.

Существует несколько разновидностей списков, у которых определены дополнительные правила работы с начальным и конечным элементами.

Кольцо (кольцевой список) — список, имеющий дополнительную связь между последним и первым элементами (первый узел является следующим для последнего).

У некоторых видов списков добавление, удаление и доступ к значениям элементов всегда осуществляются только в крайних узлах (первом или последнем). К ним относятся *стеки* и *очереди*.

Стек — разновидность списка, в котором все включения и исключения узлов (и как правило, всякий доступ) производятся только на одном его конце. Этот концевой элемент называется *вершиной стека*. Иногда стек называют *списком LIFO* (Last In — First Out — последним вошел, первым вышел).

Механизм стека напоминает железнодорожный тупик, из которого первым может выехать только поезд, въехавший в него последним.

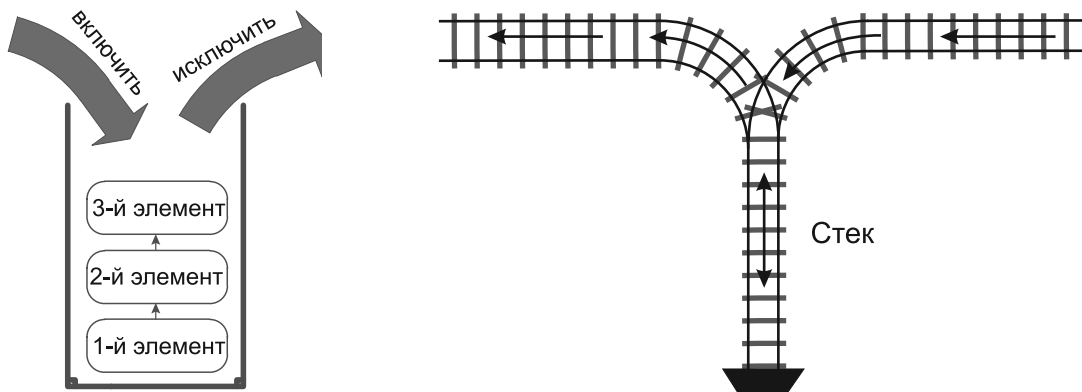


Рис. 1.11. Стек и аналогия с железнодорожным развездом

Очередь — разновидность списка, в котором все включения элементов производятся на одном его конце, а удаление узлов (и обычно всякий доступ) — на другом. Эти узлы называются *началом* и *концом очереди*. Иногда очередь называют *списком FIFO* (First In — First Out — первым вошел, первым вышел).

Механизм действия очереди действительно похож на бытовую очередь.

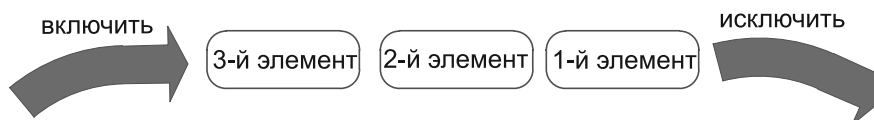


Рис. 1.12. Очередь

Деревья

Дерево — множество элементов (узлов), среди которых выделен один элемент (**корень**), а остальные элементы разбиты на группы так, что они также образуют деревья. При этом:

- а) корни поддеревьев являются потомками корня всего дерева;
- б) множества элементов поддеревьев не пересекаются друг с другом.

Узлы, которые не имеют ни одного потомка, называются **листьями** (или концевыми узлами).

Таким образом, каждый узел дерева является корнем некоторого поддерева, которое содержится в дереве, или листом. Образуется иерархическая структура узлов: все элементы связаны между собой отношениями вида «предок» — «потомок».

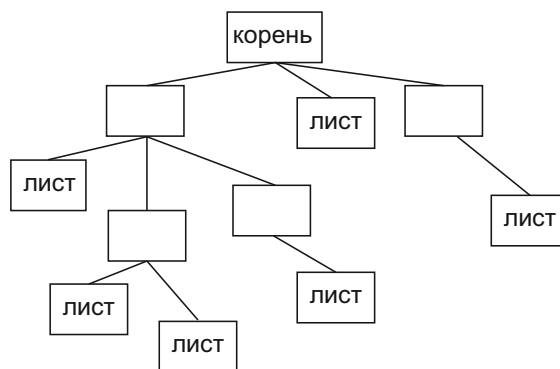


Рис. 1.13. Дерево

Двоичное (бинарное) дерево — дерево, в котором каждый узел может иметь не более двух потомков, являющихся корнями левого и правого поддеревьев.

Типовые операции над элементами дерева:

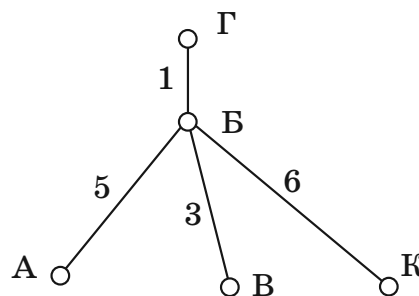
- добавление элемента в дерево,
- удаление элемента из дерева,
- обход дерева,
- поиск элемента в дереве.

Пример 2

Применим метод построения деревьев решений к задаче примера 1 (п. 1.1.2). В таблице приведены протяженность дорог между населенными пунктами (отсутствие числа означает, что между соответствующими пунктами нет прямого сообщения). Определите длину кратчайшего пути из пункта Г в пункт Д (при условии, что можно передвигаться только по имеющимся дорогам).

	А	Б	В	Г	Д	Е	К
А		4			7		7
Б	4		2	1			5
В		2			8		
Г		1					
Д	7		8			6	
Е					6		
К	7	5					

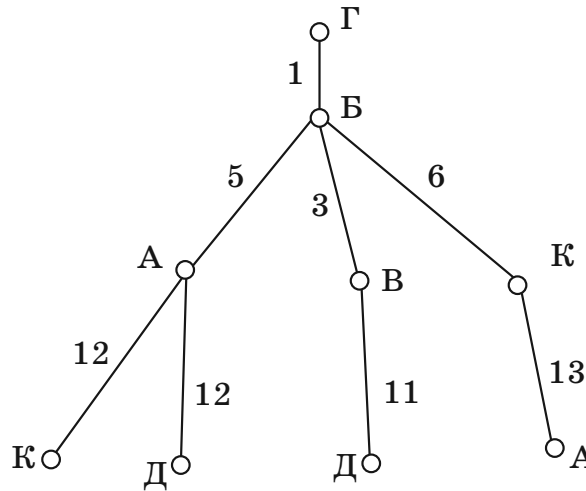
Решение. Построим дерево решения задачи. В качестве корня дерева примем стартовый пункт маршрута — Г. Из него существует только один путь — в пункт Б длиной 1. Добавим в дерево узел, подпишем его. Соединим оба узла линией (ребром) и припишем ему расстояние от Г до Б.



К узлу Б добавим поддерево, содержащее узлы всех доступных из Б населенных пунктов — А, В и К. Соединим эти узлы с Б и припишем каждому ребру суммарное расстояние от начала маршрута (пункта Г) до соответствующего пункта.

Для каждого из полученных листов дерева продолжим рассмотрение возможных маршрутов. Начнем с узла В, поскольку путь до него — самый короткий. Из пункта В достигим только пункт Д, добавим узел в дерево и рассчитаем расстояние до него ($3 + 8 = 11$). Достигнута конечная цель маршрута, но он может оказаться не самым коротким. Поэтому продолжим построение дерева но будем

обрывать его, как только на очередном пути будет получено расстояние больше найденного (>11).



Из пункта А достижимы два других пункта — Д и К. Расстояние до Д больше предыдущего, поэтому по-прежнему будем сравнивать длины всех маршрутов с 11. Расстояние до К (12) уже превышает эту величину, следовательно, эту ветвь дерева обрываем. Аналогично с поддеревом узла К — расстояние до доступного пункта А больше 11.

В результате получен самый короткий маршрут от Г до Д (длиной 11) — ГБВД.

Ответ: ГБВД.

Пример 3

У исполнителя Калькулятор в системе команд имеются команды:

1. прибавить 1
2. умножить на 3

Выполняя команду 1, исполнитель прибавляет 1 к имеющемуся числу. В результате команды 2 исполнитель умножает на 3 имеющееся у него число. Таким образом, последовательность команд 1122, примененная исполнителем к числу 3, приведет к результату 45:

прибавить 1; прибавить 1; умножить на 3; умножить на 3

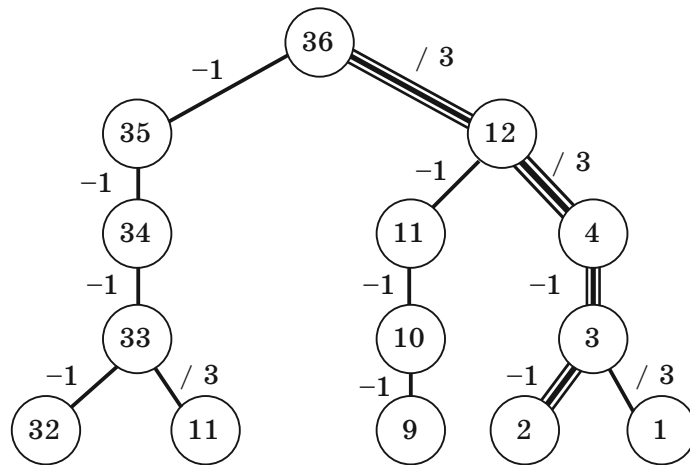
$$3 + 1 = 4; 4 + 1 = 5; 5 \times 3 = 15; 15 \times 3 = 45.$$

Требуется записать последовательность номеров команд, чтобы получить результат 36 из числа 2 не более чем за 4 шага.

Решение. Построим дерево поиска решений для обратной задачи — получения числа 2 из числа 36 с помощью двух обратных команд: разделить на 3; вычесть 1. Такой способ значительно эффективнее решения прямой задачи.

Корнем дерева будем считать узел с исходным числом 36. К нему можно применить одну из двух возможных операций — поэтому из корневого узла будет исходить два ребра. Поскольку не каждое число возможно разделить на 3, то некоторые узлы дерева будут иметь только одно ребро — для операции вычитания. Узлы, полученные вычитанием, будем располагать в левой ветви; узлы,

полученные делением, — в правой ветви. Каждому ребру припишем соответствующую операцию.



Поскольку по условию можно использовать для преобразований не более 4-х шагов, после 5-го уровня прекратим добавление узлов к дереву.

В итоге видим, что число 2 может быть получено только на одном пути (на рисунке он выделен жирной линией). Выпишем команды, приведшие к решению обратной задачи: делить на 3; делить на 3; вычесть 1; вычесть 1. Для решения прямой задачи следует выписать исходные команды в противоположном порядке: прибавить 1; прибавить 1; умножить на 3; умножить на 3. Номера этих команд: 1122.

Ответ: 1122.

1.4. КОМПЬЮТЕР КАК УНИВЕРСАЛЬНОЕ УСТРОЙСТВО ОБРАБОТКИ ИНФОРМАЦИИ

1.4.1. Основные компоненты компьютера и их функции

Компьютер — это электронное устройство для программной обработки информации.

Архитектура компьютера описывает его организацию и принципы функционирования его структурных элементов. Она включает в себя основные устройства компьютера и структуру связей между ними. Состав ПК еще называют *конфигурацией*.

Базовая конфигурация — минимальный состав компьютера, достаточный для начала работы с компьютером. В базовую конфигурацию обычно входят *системный блок, монитор (дисплей) и клавиатура*.

Системный (базовый) блок — это основной узел компьютерной системы; он содержит наиболее важные компоненты, осуществляющие обработку данных. Устройства, находящиеся внутри системного блока, называют *внутренними*, а устройства, подключаемые к нему снаружи, — *внешними*. Внешние дополни-

тельные устройства, предназначенные для ввода, вывода, обмена и длительного хранения данных, называют **периферийными**.

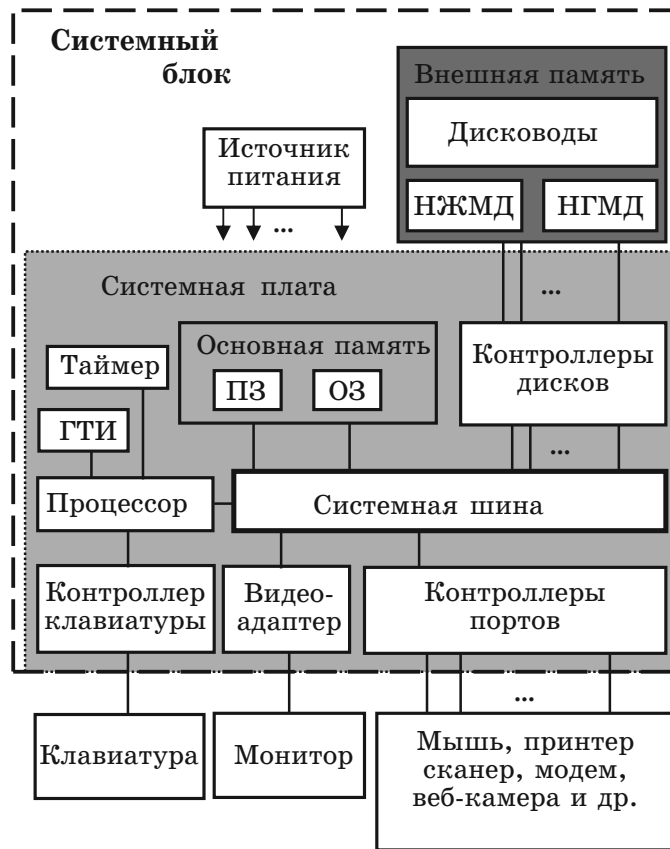


Рис. 1.14. Структурная схема ПК

Монитор (*дисплей*) компьютера предназначен для отображения информации, передаваемой в виде сигналов от *видеоконтроллера* (*видеокарты*).

Клавиатура — клавишное устройство, предназначенное для управления работой компьютера и ввода в него информации в виде алфавитно-цифровых символьных данных.

Системный блок содержит *материнскую плату*, *накопители* на магнитных и лазерных дисках, *блок питания* с вентилятором. В системном блоке также могут быть установлены *звуковая карта*, *видеокарта* и др.

Материнская (системная) плата — это сложная многослойная печатная плата, на которой располагаются все необходимые компоненты для работы компьютера. Она обеспечивает обмен информацией между устройствами с помощью различных шин. На ней расположены разъемы (*слоты*) для подключения разных устройств: *процессора*, *модулей памяти*, *адаптеров* и *контроллеров*, соединенных *системной шиной*. Материнская плата осуществляет основные функции по объединению этих компонентов компьютера в согласованно работающее устройство.

Процессор (центральный процессор, ЦП) выполняет все действия по обработке информации и управляет работой компьютера. Производительность процессора зависит от его частоты и разрядности. *Тактовая частота* — количество операций, которые процессор производит за секунду. Она измеряется в мегагерцах

(МГц) и гигагерцах (ГГц): 1 МГц означает выполнение 10^6 операций за секунду, 1 ГГц — 10^9 операций за секунду. *Разрядность* — длина двоичного кода, который процессор может обработать или передать целиком одновременно. Современные ПК обычно оснащены 32- или 64-разрядными процессорами; существуют процессоры с разрядностью 128 бит. Современные процессоры — *многоядерные*, они содержат несколько (до четырех) процессорных ядер в одном корпусе.

Основная память компьютера состоит из *оперативной памяти* (ОП, ОЗУ, оперативного запоминающего устройства) и *постоянной памяти* (ПП, ПЗУ, постоянного запоминающего устройства). **Оперативная память** — это набор микросхем, предназначенных для временного хранения данных, когда компьютер включен (после его выключения содержимое ОЗУ теряется). В ней сохраняются команды и промежуточные результаты, с которыми компьютер работает в данный момент. **Постоянная память** — это микросхема, предназначенная для длительного хранения данных, в том числе когда компьютер выключен. Она сохраняет постоянную информацию, которая записывается лишь один раз в заводских условиях и не может быть изменена пользователем. Самой важной характеристикой памяти является ее *объем*. Современным программам, например, требуется оперативная память объемом 128, 256 Мбайт и больше.

Обмен данными между отдельными элементами компьютера осуществляется через системную шину (*магистраль*). Шина — это кабель, состоящий из множества проводников. Обычно шина управляется специальной программой — *драйвером*.

Внешние устройства (клавиатура, монитор, дисководы, мышь и др.) подсоединяются к системной шине через **адаптеры** и **контроллеры**, которые обеспечивают функционирование этих устройств.

Устройства внешней памяти называются **накопителями**. Они предназначены для длительного сохранения информации. К ним относятся накопители на жестких, гибких и оптических дисках, флеш-память и др. *Накопитель на жестких магнитных дисках* (НЖМД, HDD — Hard Disk Drive, он же «винчестер») — основное устройство для долговременного хранения больших объемов данных и программ практически всех современных компьютеров. Одна из основных характеристик жесткого диска — *емкость* (количество данных, которые могут храниться накопителем; для современных устройств достигает нескольких терабайт). *Гибкие магнитные диски* были вытеснены компакт-дисками (*оптическими дисками*) и DVD, а затем — *флеш-памятью* (твердотельными носителями данных), которые имеют значительно большую емкость и надежность. В настоящее время существуют не только внутренние, но и внешние дисководы, имеющие удобное подключение к настольному ПК, ноутбуку, нетбуку.

Звуковая карта (звуковая плата) — это плата, которая позволяет работать на компьютере со звуком.

Видеокарта (графическая плата, видеоадаптер) — устройство, преобразующее изображение, находящееся в памяти компьютера, в видеосигнал для монитора.

Периферийные устройства

Периферийные устройства — устройства для ввода или вывода информации: *принтеры, клавиатуры, мыши, сканеры* и т. д. Подсоединение их к компьютеру производится через специальные разъемы — *порты ввода/вывода*. По способу передачи информации различают *последовательные* (информация передается последовательно) и *параллельные* (несколько битов информации передается

одновременно) порты. В настоящее время они вытесняются шиной *USB* и беспроводными технологиями передачи информации.

УСТРОЙСТВА ВВОДА ИНФОРМАЦИИ

Клавиатура. На сегодняшний день существует огромное количество различных клавиатур: мультимедийные и веб-клавиатуры, эргономичные и игровые, беспроводные и гибкие, виртуальные лазерные и др. По методу подключения к системному блоку различают проводные (все чаще подключаемые с помощью *USB*) и беспроводные клавиатуры.

Мышь — устройство управления манипуляторного типа. По сути, это датчик координат, определяющих положение указателя на экране. Перемещение мыши по плоской поверхности синхронизировано с перемещением графического объекта (указателя мыши) на экране монитора. Комбинация монитора и мыши обеспечивает наиболее современный тип интерфейса пользователя — *графический*. С помощью мыши пользователь изменяет свойства графических объектов и приводит в действие элементы управления компьютером.

Существуют мыши мало распространенного *механического (шарикового)* типа и современные — *оптического* типа, а также *беспроводные* мыши. В *шариковой* мыши ее движение передается в компьютер благодаря встроенному металлическому шарик, покрытому резиной, который вращается при перемещении мыши. В *оптической* — датчик улавливает свет, излучаемый встроенным диодом и отражаемый от поверхности стола. В *лазерных* мышках вместо диода установлен лазер, благодаря чему свет почти не рассеивается и достигается большая точность. *Беспроводные* мыши используют радиосвязь или инфракрасный порт.

Трекбол — встроенный в клавиатуру или мышь шарик, вращение которого вызывает перемещение курсора (по сути, это «перевернутая» шариковая мышь).

Сенсорная панель (тачпад) — сенсорная пластина, реагирующая на движение пальца пользователя по поверхности. Удар пальцем по поверхности тачпада воспринимается как нажатие кнопки.

Трекпойнт — специальная гибкая клавиша на клавиатуре, прогиб которой в нужном направлении перемещает курсор на экране дисплея.

Графический планшет — используется для рисования, а также для ввода рукописного текста с помощью специальной ручки.

Джойстик — рукоять с кнопкой. При вращении рукояти перемещается курсор на экране.

Сканер — устройство для переноса печатного текста и графических изображений (схем, рисунков, графиков, фотографий и др.) с бумаги в компьютер. Считывающая головка сканера равномерно движется над изображением, а специальное устройство преобразует его в цифровые коды.

Цифровая фотокамера — устройство для ввода фотоснимков в память компьютера.

Звуковая карта и микрофон — устройство для ввода звуковой информации.

УСТРОЙСТВА ВЫВОДА ИНФОРМАЦИИ

Монитор. Основным компонентом мониторов обычно является матрица *жидкокристаллических (ЖК) элементов*, реже — *электронно-лучевая трубка (ЭЛТ)*. Перспективными моделями считаются плазменные, проекционные и *OLED-мониторы* (в основе которых — органические светоизлучающие диоды).

Монитор подключается к компьютеру через устройство сопряжения — *видеоадаптер*. Основные параметры мониторов:

- *размер экрана* — длина его диагонали; измеряется в дюймах* (например, 15", 17", 19", 21", 22" и т. д.);
- *разрешение* (разрешающая способность) — число пикселей** по вертикали и горизонтали. Чем больше разрешение, тем выше качество изображения. Для размера экрана 17" ЖК-монитора оптимальным считается разрешение 1280 × 1024 пикселей;
- *время отклика* пикселей, или *инерционность* — измеряется в миллисекундах (лучшие мониторы имеют значение этого параметра около 2 мс);
- *угол обзора* — параметр, показывающий, на какой угол может отклониться взгляд человека без потери им видимости изображения на мониторе.

Принтер — устройство вывода информации на бумагу. Существует множество видов принтеров; чаще всего используют два вида:

- *струйные* — изображение формируется из капель чернил (тонера), которые выбрасывает печатающая головка принтера;
- *лазерные* — изображение создается лазерным лучом на светочувствительном барабане внутри принтера. Там, где луч подсвечивает поверхность барабана, возникает электрический заряд, который притягивает сухие частицы краски-тонера. Когда барабан касается бумаги, тонер переводится на нее, затем нагревается, плавится и фиксируется на бумаге.

Плоттер (графопостроитель) — устройство печати сложных графических изображений — чертежей, схем, графиков, карт, диаграмм;

Акустические колонки и наушники — устройство для прослушивания звука.

1.4.2. Командное взаимодействие пользователя с компьютером, графический интерфейс пользователя

Совокупность способов взаимодействия пользователя с программами и устройствами компьютера называется **пользовательским интерфейсом**. Он включает возможности задания команд (например, запуска программ), виды и способы вывода сообщений компьютера и т. п. Различают *командный интерфейс* (интерфейс командной строки, консольный режим, текстовый интерфейс) и *графический интерфейс*.

Командное взаимодействие с компьютером — одно из самых давних. Основным устройством управления в данном случае является клавиатура. Пользователь вводит специальные управляющие команды (текстовые инструкции) в определенную строку, называемую командной строкой. В начале такой строки выводятся специальные знаки — *приглашения*. Они указывают, что компьютер ждет команд от пользователя. Исполнение команды начинается после ее утверждения, например, нажатием клавиши Enter.

Графический интерфейс реализован с помощью графических средств. Это более сложный тип интерфейса, в котором в качестве органа управления кроме клавиатуры может использоваться мышь или другое устройство позиционирования.

* Один дюйм равен 2,54 см.

** *Пиксель* — наименьший логический элемент цифрового изображения. Он обладает определенным цветом, обычно имеет квадратную или круглую форму.

Характерным признаком графического интерфейса служат окна. *Окно* — это прямоугольная область экрана, где отображается информация программы. Каждая программа имеет хотя бы одно окно для взаимодействия с пользователем.

Современный графический интерфейс — *многооконный*, он предназначен для одновременной работы с несколькими программами и обмена данными между ними. Пользователь может работать с программами попеременно, переключаясь из одного окна в другое. Но в один момент времени можно вводить информацию только в одном окне, которое называется *активным* в текущий момент. Оно размещается поверх других окон.

Окно может находиться в одном из трех состояний:

- максимальном (развернутом на полный экран);
- нормальном (занимает часть экрана);
- минимальном (свернутом): окно не видно на экране, отображается только кнопка с названием программы или документа.

Существует несколько типов окон, вид которых определяются отображаемой в них информацией.

Основные типы окон:

- окно папки;
- окно приложения (программы);
- диалоговое окно;
- окно справочной системы (является разновидностью диалогового окна, но в нем используются гиперссылки для перехода к различным разделам справки).

Окно папки



Рис. 1.15. Окно папки Мой компьютер

Стандартные элементы, составляющие структуру окна любой папки:

- *строка заголовка* — горизонтальная полоса вдоль верхней границы окна, содержащая его заголовок. Захватывая эту строку мышью, можно перемещать окно;
- значок *системного (служебного) меню*, с помощью которого вызываются команды изменения размеров окна и его перемещения (находится в левом верхнем углу любого окна папки);

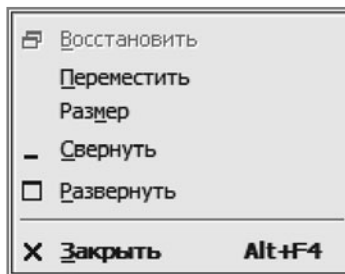


Рис. 1.16. Системное меню

- *строка меню*. Содержит выпадающие меню, команды которых позволяют проводить операции с содержимым окна или с окном в целом. Выбираются щелчком мыши;
- *панели инструментов*. Представляют собой наборы командных кнопок для наиболее часто используемых команд. Зачастую панель инструментов бывает настраиваемой — пользователь сам может разместить на ней необходимые ему командные кнопки;
- *адресная строка*. В ней указан путь доступа к текущей папке. Позволяет выполнить быстрый переход к другим папкам, набрав вручную их адрес или выбрав их из меню на правом краю строки;
- *рабочее поле* — область, где отображаются значки объектов, хранящихся в папке;
- *строка состояния* содержит информацию о режимах работы.

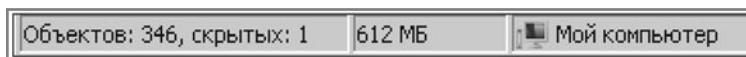


Рис. 1.17. Фрагмент строки состояния

Вертикальные и горизонтальные линейки прокрутки (скроллинг). Служат для просмотра документа по вертикали и горизонтали. Появляются, если размер окна слишком мал и не вмещает все объекты. Полоса прокрутки имеет «ползунок» и две концевые кнопки со стрелками.

Диалоговое окно

Диалоговое окно — это небольшое временное окно, которое отображается программой после определенного события или команды и служит для настройки параметров. Оно также выводит необходимые в процессе работы сообщения. Диалоговое окно содержит набор типовых объектов (элементов) управ-

ления. Перейти от одного элемента управления к другому можно с помощью клавиши Tab.

Наиболее часто встречаются элементы управления диалогового окна:

- *командные кнопки* имеют прямоугольную форму и служат для выполнения написанных на них команд. Выполнить команду можно щелчком мыши или клавишами Enter или Пробел по кнопке;
- *переключатели* имеют форму маленького круга и предназначены для выбора только одного из нескольких возможных вариантов. Вариант отмечается точкой внутри круга и выбирается щелчком мыши на кнопке или клавишей Пробел;
- *флажки* имеют квадратную форму и предназначены для включения или выключения режимов. При щелчке мышью на переключателе в его поле появляется специальный знак или, наоборот, этот знак исчезает — ;
- *текстовые поля* предназначены для ввода текстовых значений;
- *поля списка* служат для выбора одного варианта из предлагаемого перечня. Варианты открываются в виде выпадающего списка по щелчку на стрелке справа;
- *счетчик* предназначен для ввода числовых полей, значение можно установить с помощью кнопок со стрелками;
- *вкладки* расположены в окне одна под другой, так что видны только их ярлычки. Выбрать вкладку можно щелчком мыши по ярлычку;
- *ползунок* позволяет плавно изменять значение параметра.

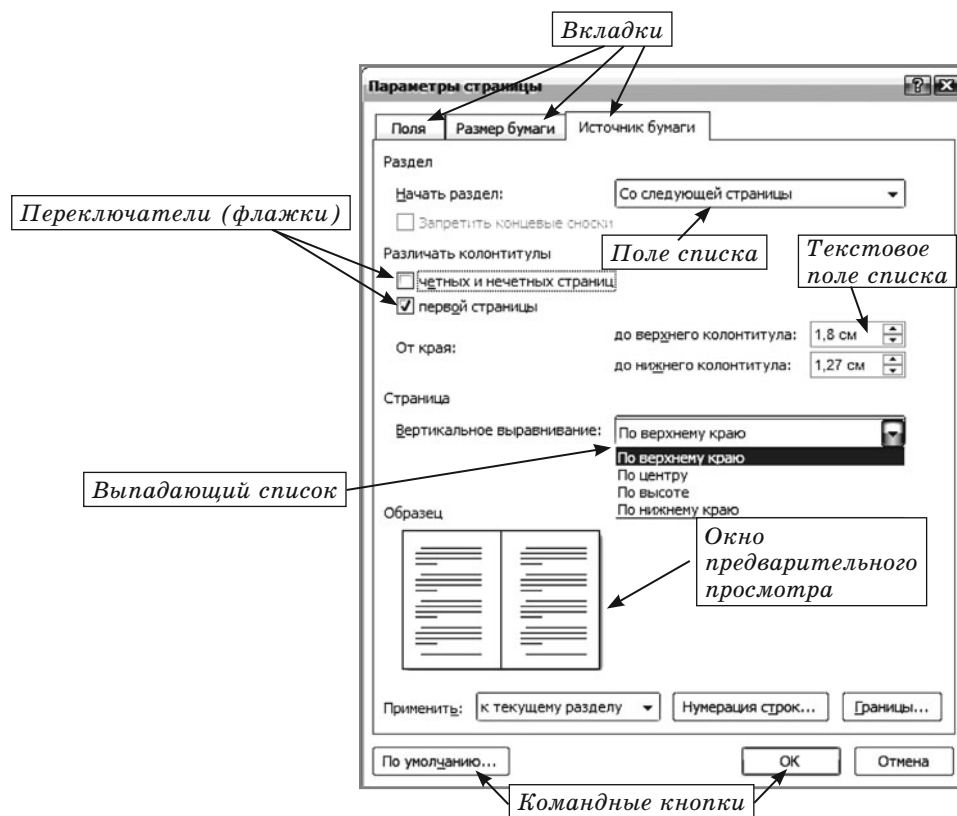


Рис. 1.18. Пример диалогового окна

Различия между кнопками *ОК*, *Отменить*, *Применить* в диалоговых окнах: *ОК* — сохраняет изменения и закрывает диалоговое окно; *Применить* — сохраняет изменения и оставляет диалоговое окно открытым; *Отменить* — отменяет изменения и закрывает диалоговое окно (аналог клавиши Esc).

В графическом интерфейсе различают *главное*, *системное*, *контекстное меню* и *меню приложения*. **Главное меню** предназначено для начала и окончания работы с компьютером, выбора программ для работы и проч. **Системное меню** позволяет управлять окнами — изменять их размеры, перемещать и т. д. **Контекстное меню** появляется при нажатии правой кнопки мыши. Оно содержит перечень свойств выделенного объекта и операций, разрешенных для него. Содержание контекстного меню зависит от позиции указателя мыши. **Меню приложения** предоставляет список команд конкретной программы.

1.4.3. Программное обеспечение, его структура.

Программное обеспечение общего назначения

Программное обеспечение (ПО) — это совокупность программ, предназначенных для решения определенных задач на компьютере.

В зависимости от области использования программ их можно разделить на 3 класса:

- *системное программное обеспечение (СПО)* — требуется для управления работой компьютера;
- *прикладное программное обеспечение (ППП — пакеты прикладных программ)* — необходимо пользователю для выполнения различных задач;
- *инструментальные системы программирования (ИСП)* — предназначены для создания новых компьютерных программ.



Рис. 1.19. Классы программных продуктов

Системное ПО

Системное программное обеспечение (СПО) состоит из двух компонентов:

- *базового программного обеспечения* (обычно поставляется вместе с компьютером);
- *сервисного программного обеспечения* (устанавливается дополнительно).

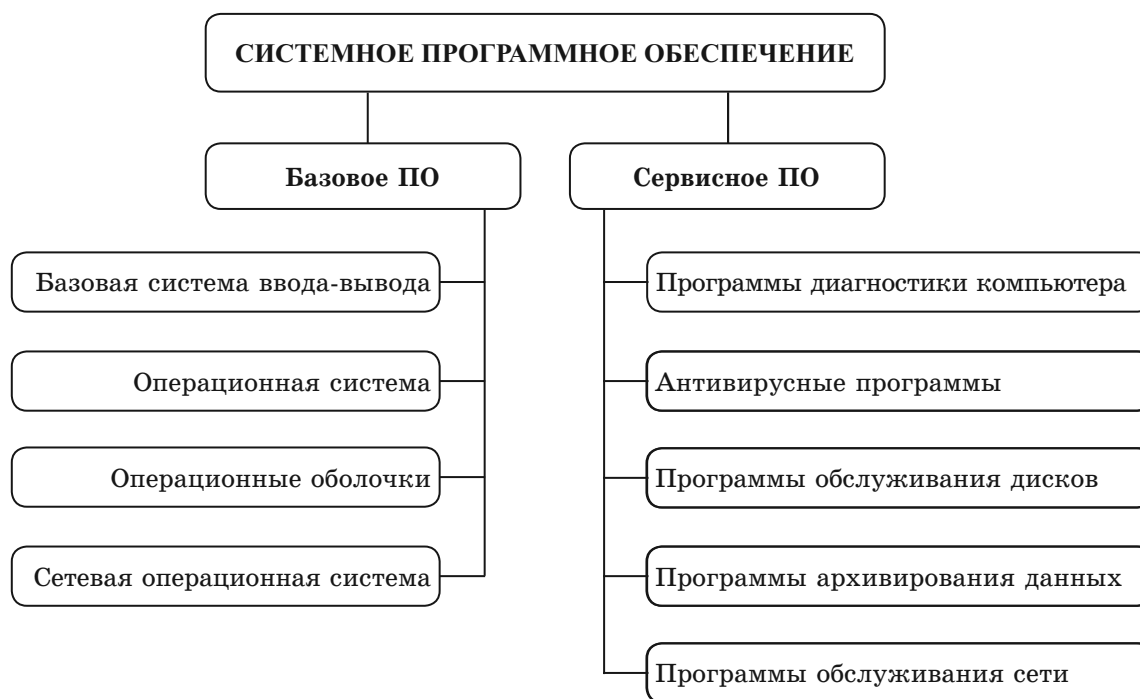


Рис. 1.20. Классификация системного программного обеспечения компьютера

Базовое программное обеспечение содержит минимальный набор программ, обеспечивающих работу компьютера. **Сервисное программное обеспечение** включает программы и программные комплексы, которые расширяют возможности базового ПО и организуют более удобную среду работы пользователя.

В базовое ПО входят: BIOS, операционная система и операционные оболочки (текстовые и графические).

BIOS отвечает за взаимодействие с основными аппаратными средствами и хранится в специальных микросхемах (ПЗУ).

Операционная система (ОС) — комплекс программ, предназначенный для управления выполнением пользовательских программ, планирования и управления вычислительными ресурсами. Операционная система загружается при включении компьютера и организует диалог с пользователем.

Сетевые операционные системы — комплекс программ, обеспечивающий обработку, передачу и хранение данных в сети.

Операционные оболочки — специальные программы для облегчения общения пользователя с командами операционной системы. Операционные оболочки могут иметь текстовый и графический варианты интерфейса.

Основное назначение сервисных, или служебных, программ (утилит) — проверка и настройка компьютерной системы. Обычно утилиты дополнительно устанавливаются на ПК и служат для расширения ее функций:

- диагностирование работоспособности компьютера;
- антивирусная защита;
- обслуживание дисков;
- архивирование данных;

- обслуживание сети;
- установка драйверов — программ, отвечающих за взаимодействие с конкретными устройствами.

ОПЕРАЦИОННАЯ СИСТЕМА

Операционная система (ОС) — главная составляющая системного программного обеспечения. Это комплекс программ, которые загружаются при включении компьютера и осуществляют управление компьютером, диалог с пользователем, запускают другие программы на выполнение.

ОС состоят из следующих категорий программ:

- *ядро* — основа ОС, выполняющая главные функции (загрузку компонентов ОС и поддержку выполнения компьютерных программ);
- программа управления файлами и папками (*файловая система*), служащая для классификации и просмотра информации;
- *драйверы*, которые позволяют ОС работать с аппаратурой: периферийными устройствами (монитор, клавиатура, мышь, принтеры и т. д.) и устройствами, входящими в состав системного блока (видеокарта, жесткий диск и т. д.). Без драйверов невозможно функционирование никаких компьютерных устройств.

Программы, предназначенные для работы под управлением данной операционной системы, принято называть *приложениями*. ОС обеспечивает интерфейсы (связь) между пользователем, приложениями и аппаратурой.

ОС осуществляет следующие операции:

- контролирует работоспособность аппаратного обеспечения ПК;
- выполняет процедуру начальной загрузки;
- управляет работой устройств ПК;
- управляет файловой системой;
- обеспечивает взаимодействие пользователя с ПК;
- загружает и выполняет прикладные программы;
- распределяет ресурсы ПК (оперативную память, процессорное время, периферийные устройства между прикладными программами и др.).

Сейчас на IBM-совместимые компьютеры устанавливаются ОС *Windows* и *Linux*, на персональные компьютеры Macintosh — *Mac OS*.

Прикладное ПО

Программное обеспечение прикладного уровня представляет собой прикладные программы (или их пакет), с помощью которых пользователь выполняет конкретные задания — от производственных до творческих, обучающих и развлекательных. Эти программные средства наиболее многочисленны.

ПРОГРАММЫ ОБЩЕГО НАЗНАЧЕНИЯ

Этот класс содержит широкий перечень программ для пользователей. Среди них — текстовые и графические редакторы, электронные таблицы, системы управления базами данных и пр.

Основные функции **текстовых редакторов и процессоров**: ввод и редактирование текстов в различных форматах, подготовка печатных документов —

статей, справок, отчетов, ведомостей и т. п. *Текстовые редакторы* позволяют вводить и редактировать тексты и могут содержать некоторые ограниченные возможности их форматирования (оформления). Примеры текстовых редакторов: Блокнот (Notepad), WordPad. *Текстовые процессоры* содержат большой перечень возможностей форматирования документов, а также средства объединения текста, графики, таблиц, формул и других объектов в итоговый документ. Часто текстовые процессоры включают различные средства автоматизации, составления оглавлений и указателей, проверки орфографии, подготовки шаблонов документов и др. Примеры текстовых процессоров: Microsoft Word, WordPerfect, OpenOffice.org Writer, AbiWord. *Издательские системы*, например Adobe InDesign, служат для подготовки и тиражирования газет, журналов, рекламных буклетов, проспектов, книг.

Табличные процессоры, или электронные таблицы предоставляют среду для обработки информации, представленной в виде таблиц, с использованием встроенных функций. Эти программы широко применяются в бухгалтерском учете, при анализе данных, всюду, где необходимо автоматизировать регулярно повторяющиеся вычисления достаточно больших объемов числовых данных. Кроме этого, такие программы помогают строить двух- или трехмерные графики, диаграммы. Примерами таких программ являются: Microsoft Excel, OpenOffice.org Calc, Lotus Symphony Spreadsheets и др.

Графические редакторы и процессоры предназначены для просмотра, обработки, редактирования, упорядочения и публикации цифровых фотографий, создания графических файлов, преобразования их из одного формата в другой и (или) обработки графических изображений — рисунков, чертежей, графиков, диаграмм, иллюстраций, в том числе и трехмерных. Пользователь может изменять палитру, масштаб, работать с многослойными изображениями, получать изображения со сканера и другой цифровой техники и т. д. Любой графический редактор включает в себя текстовый редактор и позволяет набирать тексты.

Системы управления базами данных (СУБД) обеспечивают организацию и хранение баз данных. *Базами данных* называют большие массивы данных о группе объектов с одинаковым набором свойств. СУБД обеспечивают ввод данных, поиск, сортировку записей, создание отчетов. Примерами СУБД являются Microsoft Access, Microsoft SQL Server, MySQL, Oracle, Sybase, Firebird, Interbase.

К средствам презентационной графики относятся специализированные программы для создания изображений и их показа на экране, подготовки слайдфильмов, мультфильмов, видеофильмов, их редактирования. Презентация может включать показ диаграмм и графиков.

Существуют программные средства, объединяющие возможности текстовых, графических редакторов, электронных таблиц, систем управления базами данных. Такие средства называются **интегрированными системами, или пакетами**. Они могут включать также органайзер, средства поддержки электронной почты, программу для создания презентаций и др. Эти средства предусматривают единые правила работы с программами (унифицированный интерфейс). Компоненты интегрированных пакетов могут работать отдельно друг от друга, но основные достоинства этих пакетов проявляются при их разумном сочетании друг с другом. Примерами интегрированных пакетов являются Microsoft Office, OpenOffice.org, Lotus SmartSuite, StarOffice, Corel WordPerfect Office.

К прикладному программному обеспечению относятся также:

- офисные пакеты;
- средства специализированного назначения;
- коммуникационные средства;
- программы мультимедиа;
- развлекательные и образовательные программы;
- системы искусственного интеллекта;
- игровые программы.

Офисные пакеты могут включать программы планирования (органайзеры), системы автоматического перевода, средства проверки орфографии и распознавания текста. К **специализированным (проблемно-ориентированным) средствам** относятся математические пакеты, системы деловой и научной графики, системы автоматизированного проектирования, бухгалтерские системы и т. д. **Коммуникационные средства** предназначены для взаимодействия пользователей в компьютерных сетях и включают браузеры (программы для просмотра содержимого веб-страниц), почтовые программы, средства скачивания файлов из сетей, программы для видеообщения и обмена текстовыми, голосовыми сообщениями, средства прослушивания онлайн-радио и просмотра онлайн-телевидения и др. Основное назначение **мультимедийных программ** — создание, обработка и проигрывание аудио- и видеoinформации. Современные **обучающие программы** обычно являются мультимедийными, включают не только звук и работу с микрофоном, но и отрывки из видеофильмов. **Системы искусственного интеллекта** реализуют отдельные функции интеллекта человека; к ним относятся экспертные системы для принятия решений в предметной области (медицина, математика, статистика и т. д.), системы анализа и распознавания речи и др. **Игровые программы** используются не только для отдыха и соревнований, но и для тренажерной тренировки навыков и умений, тренировки логического мышления, а также обучения.

Инструментальное ПО

Инструментальные системы, или системы программирования предназначены для создания новых программ. Они различаются в первую очередь по языкам программирования, которые они поддерживают.

В настоящее время широко распространены *системы визуального программирования*, которые позволяют создавать программы с использованием графического интерфейса.

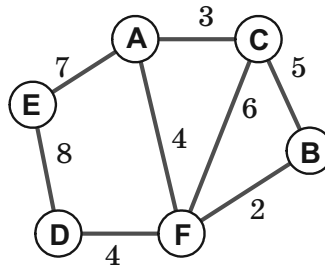
Тренировочные тестовые задания к разделу 1

Тема 1.1

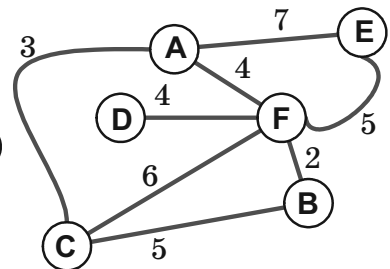
При выполнении заданий с выбором ответа обведите кружком номер правильного ответа в задании.

1. В таблице приведена протяженность путей для дорог между пятью населенными пунктами (отсутствие числа означает, что между соответствующими пунктами нет прямого сообщения). Укажите схему, соответствующую таблице.

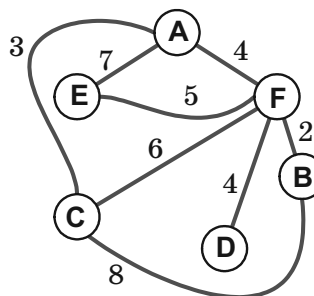
	A	B	C	D	E	F
A			3		7	4
B			8			2
C	3	8				6
D						4
E	7					5
F	4	2	6	4	5	



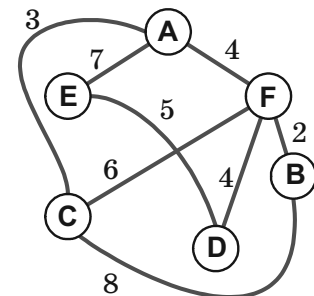
1)



2)



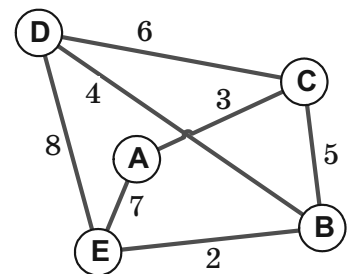
3)



4)

2. На схеме нарисованы дороги между пятью населенными пунктами и указана их протяженность. Найдите два пункта, наиболее удаленные друг от друга. Укажите кратчайшее расстояние между этими пунктами.

- 1) 9 2) 8 3) 26 4) 10



3. В таблице указана стоимость проезда между пятью городами (отсутствие числа означает, что между соответствующими городами нет прямого сообщения). Найдите стоимость самого дешевого маршрута между городами A и D.

- 1) 15 2) 13 3) 10 4) 12

	A	B	C	D	E
A			3		7
B			5	4	2
C	3	5		10	
D		4	10		8
E	7	2		8	

4. Выберите вариант, в котором величины объемов памяти расположены по возрастанию:
- 1) 768 байт; 12 Кбит; 20 Кбайт; 1 Мбит; 2000 бит
 - 2) 1 Мбит; 768 байт; 12 Кбит; 2000 бит; 20 Кбайт
 - 3) 768 байт; 20 Кбайт; 2000 бит; 12 Кбит; 1 Мбит
 - 4) 2000 бит; 768 байт; 12 Кбит; 20 Кбайт; 1 Мбит

Ответом к заданиям этой части является набор символов (букв или цифр), который следует записать в отведенном в задании поле для записи ответа.

5. Два текстовых сообщения одной кодировки объемом 1,2 Мбит и 720 Кбит объединили в одно, а затем удалили из него фрагмент объемом 80 Кбайт. Определите объем получившегося сообщения в байтах.

Ответ: _____.

6. Сколько мегабит информации содержится в сообщении объемом 2^{19} байт?

Ответ: _____.

7. Объем сообщения, содержащего 4096 символов, составил $1/256$ часть мегабайта. Какое количество байтов использовалось для кодирования одного символа?

Ответ: _____.

8. Определите, сколько единиц в двоичной записи десятичного числа 105.

Ответ: _____.

9. Переведите десятичное число 166 в двоичную систему счисления.

Ответ: _____.

10. Переведите десятичное число 805 в восьмеричную систему счисления.

Ответ: _____.

11. Переведите десятичное число 6851 в шестнадцатеричную систему счисления.

Ответ: _____.

12. Переведите число 110111011_2 из двоичной в десятичную систему счисления.

Ответ: _____.

13. Переведите число 765_8 из восьмеричной в десятичную систему счисления.

Ответ: _____.

14. Переведите число $14A_{16}$ из шестнадцатеричной в десятичную систему счисления.

Ответ: _____.

Тема 1.2

При выполнении заданий с выбором ответа обведите кружком номер правильного ответа в задании.

1. Во время шифрования русских слов вместо каждой буквы был записан ее номер в алфавите (без пробелов). Номера букв приведены в таблице.

НОМЕРА БУКВ ДЛЯ ШИФРОВКИ СООБЩЕНИЙ

А	1	Й	11	У	21	Э	31
Б	2	К	12	Ф	22	Ю	32
В	3	Л	13	Х	23	Я	33
Г	4	М	14	Ц	24		
Д	5	Н	15	Ч	25		
Е	6	О	16	Ш	26		
Ё	7	П	17	Щ	27		
Ж	8	Р	18	Ъ	28		
З	9	С	19	Ы	29		
И	10	Т	20	Ь	30		

Некоторые шифровки можно расшифровать несколькими способами. Например, 12118201 может означать «КАРТА», или «АБААЖТА», или «АУАЖТА» и т.д. Даны четыре шифровки. Только одна из них расшифровывается единственным способом. Определите эту шифровку.

- 1) 192036 2) 912063 3) 202319 4) 710215

2. Во время шифрования русских слов вместо каждой буквы был записан ее номер в алфавите (без пробелов). Номера букв приведены в таблице (см. условие задания 1). Некоторые шифровки можно расшифровать несколькими способами. Например, 14101818 может означать «МИРАЖ» или «МИРР», или «АГИАЖАЖ» и т. д.

Даны четыре шифровки. В каждом зашифрованном слове все буквы различны. Только одна из шифровок расшифровывается единственным способом. Определите эту шифровку.

- 1) 132013252 2) 913019133 3) 112312523 4) 514291317

Ответом к заданиям этой части является набор символов (букв или цифр), который следует записать в отведенном в задании поле для записи ответа.

3. Для кодирования букв О, К, Л, Б используется двоичное представление чисел 0, 1, 2 и 3 соответственно (для одnorазрядного представления добавляется один незначащий нуль в начале). Закодируйте последовательность букв КОЛОБОК таким способом и укажите количество единиц в полученном коде.

Ответ: _____.

4. Для кодирования букв А, И, У, Н, В используется двоичное представление чисел 0, 1, 2, 3 и 4 соответственно (для одноразрядного представления добавляется один незначащий нуль в начале). Укажите наибольшее число подряд идущих нулей в закодированной таким способом последовательности букв БАБУИН.

Ответ: _____.

5. Для кодирования букв А, К, М, С, Р используется двоичное представление:

А	К	М	С	Р
00	01	10	11	100

Закодируйте последовательность букв МАРАКАС таким способом и запишите результат шестнадцатеричным кодом.

Ответ: _____.

6. Во время шифрования русских слов вместо каждой буквы был записан ее номер в алфавите (без пробелов). Номера букв приведены в таблице (см. условие задания 1). Некоторые шифровки можно расшифровать несколькими способами. Например, 19171618 может означать «СПОР» или «СПАЕОАЖ», или «АЗАЁАЕР» и т. д.

Даны четыре шифровки:

206192 362061 241693 330117

Только одна из них расшифровывается единственным способом. Найдите ее и расшифруйте. Получившееся слово укажите в качестве ответа.

Ответ: _____.

7. В радиограмме использовались только 4 буквы из азбуки Морзе:

А К Н О
 • — — • — — • — — —

Известно, что в исходном тексте повторялась не более чем одна буква. Определите текст радиограммы, в которой было утеряно разбиение на буквы:

— • — — — — — • — • — • —

Ответ: _____.

8. В радиограмме использовались только 5 букв из азбуки Морзе:

А К Н О
 • — — • — — • — — —

Известно, что в исходном тексте повторялась не более чем одна буква. Определите текст радиограммы, в которой было утеряно разбиение на буквы:

— • • • • — • — — • —

Ответ: _____.

Тема 1.3

При выполнении заданий с выбором ответа обведите кружком номер правильного ответа в задании.

1. Исполнитель Оформитель наносит на полотно геометрические узоры, двигаясь над ним по командам. Команда **Сдвиг на (a, b)** (где a и b — целые числа) предписывает Оформителю переместиться из точки с координатами (x, y) в точку с координатами $(x + a, y + b)$. Если числа a и b положительные, соответствующие координаты увеличиваются, если отрицательные — уменьшаются.

Например, если Оформитель находится в точке с координатами $(1, 5)$, то команда **Сдвиг на $(2, -6)$** переместит Оформителя в точку с координатами $(3, -1)$.

Запись

Повторить k раз

команда1 команда2 команда3

Конец

означает, что последовательность команд *команда1 команда2 команда3* должна быть выполнена k раз подряд.

Оформителю был дан для исполнения алгоритм:

Повторить 3 раза

Сдвиг на $(2, 2)$ Сдвиг на $(-1, -3)$ Сдвиг на $(0, 6)$

Конец

Определите, в какой точке окажется Оформитель после выполнения этого алгоритма, если координаты его текущего местонахождения $(5, 6)$.

1) $(8, 2)$

2) $(7, 16)$

3) $(8, 21)$

4) $(-1, 21)$

2. Исполнитель Кузнечик прыгает по плоскости, выполняя команды. Команда **Прыгнуть на (a, b)** (где a и b — целые числа) предписывает Кузнечику прыгнуть из точки с координатами (x, y) в точку с координатами $(x + a, y + b)$. Если числа a и b положительные, соответствующие координаты увеличиваются, если отрицательные — уменьшаются.

Например, если Кузнечик находится в точке с координатами $(2, 4)$, то команда **Прыгнуть на $(1, -5)$** переместит Кузнечика в точку с координатами $(3, -1)$.

Запись

Повторить k раз [*команда1 команда2 команда3*]

означает, что последовательность команд *команда1 команда2 команда3* должна быть выполнена k раз подряд.

Кузнечику был дан для исполнения алгоритм:

Повторить 3 раза [**Прыгнуть на $(0, -1)$ Прыгнуть на $(-1, 3)$ Прыгнуть на $(2, -2)$**]

Какой одной командой можно заменить этот алгоритм, чтобы Кузнечик оказался в той же точке, что и после выполнения алгоритма?

1) Прыгнуть на $(1, -1)$

3) Прыгнуть на $(-1, 3)$

2) Прыгнуть на $(1, 0)$

4) Прыгнуть на $(3, 0)$

3. Исполнитель Монтажник перемещается над электросхемой, выполняя команды. Команда **Сместить на** (a, b) (где a и b — целые числа) предписывает Монтажнику переместиться из точки с координатами (x, y) в точку с координатами $(x + a, y + b)$. Если числа a и b положительные, соответствующие координаты увеличиваются, если отрицательные — уменьшаются.

Например, если Монтажник находится в точке с координатами $(2, 3)$, то команда **Сместить на** $(5, -1)$ переместит Монтажника в точку с координатами $(7, 2)$.

Запись

Повторить k раз
команда1 команда2 команда3
Конец

означает, что последовательность команд *команда1 команда2 команда3* должна быть выполнена k раз подряд.

Монтажнику был дан для исполнения алгоритм:

Сместить на $(3, 2)$
Повторить 5 раз
Сместить на $(-1, 9)$ **Сместить на** $(6, -2)$ **Сместить на** $(-5, -7)$
Конец

Какую команду нужно выполнить Монтажнику, чтобы вернуться в исходную точку, из которой он начал движение?

- | | |
|---------------------------------|----------------------------------|
| 1) Сместить на $(3, 2)$ | 3) Сместить на $(-3, -2)$ |
| 2) Сместить на $(1, -9)$ | 4) Сместить на $(2, 11)$ |

4. Исполнитель Черепашка перемещается по командам, оставляя за собой след в виде линии. Исполнитель имеет следующую систему команд:

Вперед k — продвижение на указанное число шагов вперед (k — целое число);

Поворот s — поворот на s градусов по часовой стрелке (s — целое число);

Повторить t [команда1 ... команда N] — повторить t раз серию указанных команд.

Какой след оставит за собой исполнитель после выполнения следующей последовательности команд?

Повторить 4 [Вперед 6 Поворот 120]

- | | |
|-------------------------------|------------------------------|
| 1) шестиугольник | 3) незамкнутая ломаная линия |
| 2) правильный четырехугольник | 4) правильный треугольник |

5. Исполнитель Черепашка перемещается по командам, оставляя за собой след в виде линии. Исполнитель имеет следующую систему команд:

Вперед k — продвижение на указанное число шагов вперед (k — целое число);

Поворот s — поворот на s градусов по часовой стрелке (s — целое число);

Повторить t [команда1 ... команда N] — повторить t раз серию указанных команд.

Какой след оставит за собой исполнитель после выполнения следующей последовательности команд?

Повторить 3 [Вперед 5 Поворот 120 Вперед 5 Поворот 60]

Вперед 5

- | | |
|--------------------|------------------------------|
| 1) треугольник | 3) пятиугольник |
| 2) четырехугольник | 4) незамкнутая ломаная линия |

6. Исполнитель Черепашка перемещается по бесконечному клетчатому полю. Исполнитель имеет следующую систему команд:

Вверх — продвижение на 1 шаг вверх;

Вниз — продвижение на 1 шаг вниз;

Влево — продвижение на 1 шаг влево;

Вправо — продвижение на 1 шаг вправо.

Черепашке был дан для исполнения алгоритм:

Вверх Вверх Вправо Вниз Вправо Вверх Вверх Влево

Укажите наименьшее возможное число команд, которые переместили бы Черепашку из той же начальной точки в ту же конечную точку, что и заданный алгоритм.

1) 4

2) 5

3) 6

4) 7

7. Исполнитель Чертежник перемещается, оставляя за собой след в виде линии. Чертежник может выполнять команду **Сдвиг на** (a, b) (где a и b — целые числа), которая предписывает ему переместиться из точки с координатами (x, y) в точку с координатами $(x + a, y + b)$. Если числа a и b положительные, соответствующие координаты увеличиваются, если отрицательные — уменьшаются.

Например, если Чертежник находится в точке с координатами $(3, 6)$, то команда **Сдвиг на** $(2, -1)$ переместит Чертежника в точку с координатами $(5, 5)$.

Запись

Повторить k раз

команда1 команда2 команда3

Конец

означает, что последовательность команд *команда1 команда2 команда3* должна быть выполнена k раз подряд.

Чертежнику был дан для исполнения алгоритм:

Повторить 4 раза

Сдвиг на $(3, 3)$ **Сдвиг на** $(3, -3)$

Конец

Сдвиг на $(0, 12)$

Какую фигуру нарисует Чертежник?

1) треугольник

2) четырехугольник

3) девятиугольник

4) незамкнутая ломаная линия

8. Исполнитель Чертежник перемещается, оставляя за собой след в виде линии. Чертежник может выполнять команду **Смещение на** (a, b) (где a и b — целые числа), которая предписывает ему переместиться из точки с координатами (x, y) в точку с координатами $(x + a, y + b)$. Если числа a и b положительные, соответствующие координаты увеличиваются, если отрицательные — уменьшаются.

Например, если Чертежник находится в точке с координатами $(0, 5)$, то команда **Смещение на** $(3, -1)$ переместит Чертежника в точку с координатами $(3, 4)$.

Запись **Повторить k раз** [команда1 команда2 команда3] означает, что последовательность команд команда1 команда2 команда3 должна быть выполнена k раз подряд.

Чертежнику был дан для исполнения алгоритм:

Повторить 3 раза [Смещение на (3, 4) Смещение на (3, -3)]

Выберите команду, которую нужно дописать в конец алгоритма, чтобы нарисованная фигура стала замкнутой ломаной линией.

- | | |
|--------------------------|------------------------|
| 1) Смещение на (-6, -1) | 3) Смещение на (9, -1) |
| 2) Смещение на (-18, -3) | 4) Смещение на (3, 18) |

Ответом к заданиям этой части является набор символов (букв или цифр), который следует записать в отведенном в задании поле для записи ответа.

9. В приведенном ниже алгоритме используются целочисленные переменные m и n . Символ «:=» означает оператор присваивания, знаки «+», «-», «*», «/» — операции сложения, вычитания, умножения и деления соответственно. Порядок выполнения операций соответствует правилам арифметики.

Определите значение переменной n после выполнения данного алгоритма.

```
n := 10
m := 24 - 2*n
n := m / 2 * n
```

Ответ: _____.

10. В приведенном ниже алгоритме используются целочисленные переменные a и b . Символ «:=» означает оператор присваивания, символы «+», «-», «*», «/», «mod» — операции сложения, вычитания, умножения, деления и остатка от целочисленного деления соответственно. Порядок выполнения операций соответствует правилам арифметики.

Определите значение переменной b после выполнения данного алгоритма.

```
b := 9
a := 3*b + 5
b := (b + a) mod 3
a := a - b
b := a
```

Ответ: _____.

11. В приведенном ниже алгоритме используются целочисленные переменные m и n . В алгоритме использованы такие знаки операций:

Обозначение	Тип операции
:=	Присваивание
+	Сложение
-	Вычитание
*	Умножение
/	Деление
div	Деление нацело

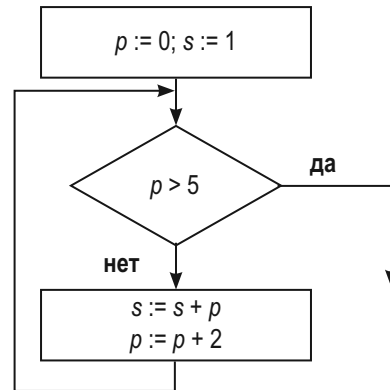
Порядок выполнения операций соответствует правилам арифметики.

Определите значение переменной n после выполнения данного алгоритма.

$m := 5$
 $n := 46 - m * 3$
 $m := n \text{ div } 6 + 11$
 $n := (n - m) * (18 - m)$

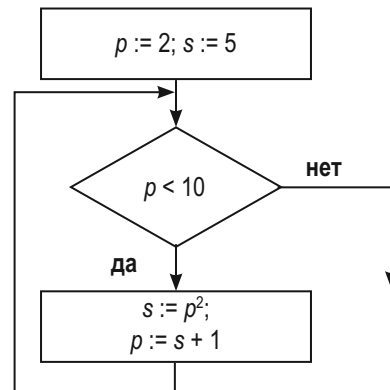
Ответ: _____.

12. Определите значение переменной s после выполнения алгоритма, представленного блок-схемой.



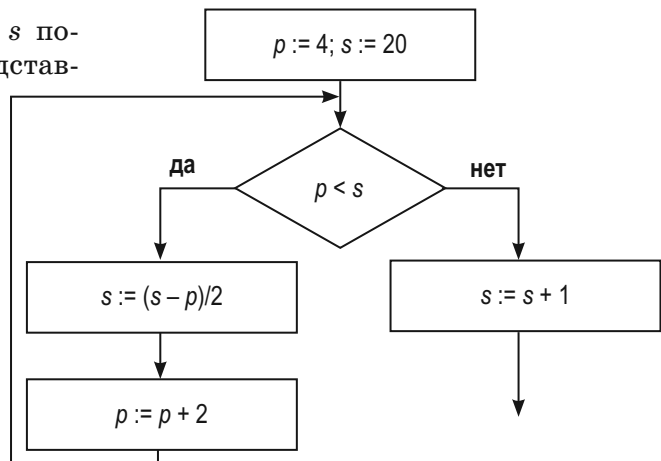
Ответ: _____.

13. Определите значение переменной s после выполнения алгоритма, представленного блок-схемой.



Ответ: _____.

14. Определите значение переменной s после выполнения алгоритма, представленного блок-схемой.



Ответ: _____.

15. Определите, что будет выведено в результате работы приведенного алгоритма.

```
алг Great
нач
  цел  $a, b, c, m$ 
   $a := 10$ 
   $b := 12$ 
   $c := 4$ 
  нц
    для  $m$  от 1 до 2
      если  $a > c$ 
        то  $c := a$ 
      всё
      если  $b > c$ 
        то  $c := b$ 
      всё
       $c := c - 1$ 
    кц
  вывод  $c$ 
кон
```

Ответ: _____.

16. Определите, что будет выведено в результате работы приведенного алгоритма.

```
алг Рост
нач
  цел  $k, s$ 
   $s := 4$ 
   $k := 8$ 
  нц
    пока  $k < 30$ 
       $s := (s + 1) * 2$ 
       $k := k + 10$ 
    кц
  вывод  $s$ 
кон
```

Ответ: _____.

17. Определите, что будет выведено в результате работы приведенного алгоритма.

```
алг Sigma
нач
  цел  $i, r, s$ 
   $r := 3$ 
   $s := 4$ 
  нц
    для  $i$  от 2 до 5
       $s := (2 * r + 10) / s$ 
       $r := r * r / 3$ 
    кц
   $s := s * s - s$ 
  вывод  $s$ 
кон
```

Ответ: _____.

18. Определите, в каком из алгоритмов верно реализована проверка того, является ли число c положительным, отрицательным или равно нулю.

```
1) если  $c = 0$ 
    то  $s :=$  "число равно нулю"
    всё
    если  $c > 0$ 
        то  $s :=$  "число положительно"
        иначе  $s :=$  "число отрицательно"
    всё
    вывод  $s$ 
```

```
2) если  $c > 0$ 
    то  $s :=$  "число положительно"
    иначе  $s :=$  "число отрицательно"
    всё
    если  $c = 0$ 
        то  $s :=$  "число равно нулю"
    всё
    вывод  $s$ 
```

```
3) если  $c = 0$ 
    то  $s :=$  "число равно нулю"
    всё
    если  $c > 0$ 
        то  $s :=$  "число положительно"
    всё
    если  $c < 0$ 
        то  $s :=$  "число отрицательно"
    всё
    вывод  $s$ 
```

19. В приведенном фрагменте алгоритма обрабатываются элементы таблицы A . Чему будут равны элементы этой таблицы по окончании работы алгоритма?

```
нц
  для  $i := 1$  до 8
     $A[i] := i + 2;$ 
кц
нц
  для  $i := 1$  до 8
     $A[i] := A[9 - i];$ 
кц
```

1) 8 7 6 5 4 3 2 1

2) 10 9 8 7 6 5 4 3

3) 10 9 8 7 7 8 9 10

4) 9 8 7 6 7 8 9 10

20. В приведенном алгоритме обрабатываются элементы таблицы Dat , содержащей количество выигрывшей семи команд. Что будет напечатано в результате работы алгоритма?


```

алг
нач
  цел таб  $Da\{1:7\}$ 
  цел  $k, m, f$ 
   $Da\{1\} := 9; Da\{2\} := 14$ 
   $Da\{3\} := 11; Da\{4\} := 12$ 
   $Da\{5\} := 7; Da\{6\} := 6$ 
   $Da\{7\} := 8$ 
   $f := 0$ 
  нц
    для  $k$  от 1 до 7
      если  $Da\{k\} < 10$ 
        то  $f := f + 1$ 
      всё
    кц
  вывод  $f$ 
кон

```

Ответ: _____.

21. В приведенном алгоритме обрабатываются элементы таблицы Z , содержащей ежемесячные данные о количестве совершенных телефонных звонков. Что будет напечатано в результате работы алгоритма?

```

алг
нач
  цел таб  $Z\{1:12\}$ 
  цел  $i, t, p$ 
   $Z\{1\} := 105; Z\{2\} := 214$ 
   $Z\{3\} := 268; Z\{4\} := 98$ 
   $Z\{5\} := 118; Z\{6\} := 216$ 
   $Z\{7\} := 277; Z\{8\} := 148$ 
   $Z\{9\} := 219; Z\{10\} := 285$ 
   $Z\{11\} := 150; Z\{12\} := 307$ 
   $p := 1; t := Z\{1\}$ 
  нц
    для  $i$  от 2 до 12
      если  $Z\{i\} < t$ 
        то  $t := Z\{i\}; p := i$ 
      всё
    кц
  вывод  $p$ 
кон

```

Ответ: _____.

22. Определите, для какого из указанных значений числа Z истинно выражение.

$$Z < 3 \vee \neg (Z \geq 0) \wedge \neg (Z + 2 \leq 10)$$

- 1) 1 2) 5 3) 8 4) 10

23. Определите, для какого из указанных значений числа Z ложно выражение.

$$(2*Z + 3 < 9) \wedge \neg (Z / 2 < 3) \vee \neg (Z \geq 6)$$

- 1) 1 2) 6 3) 3 4) 8

24. Определите, для какого из указанных значений числа X истинно выражение.

$$\text{НЕ } (X / 2 < 5) \text{ И НЕ } (2*X < 3) \text{ ИЛИ НЕ } (X < 9)$$

- 1) 2 2) 10 3) 6 4) 4

25. Для какого из приведенных чисел истинно выражение?

$$\text{НЕ (Все цифры равны)} \text{ ИЛИ НЕ (Вторая цифра четная)} \text{ И НЕ (Первая цифра равна 2)}$$

- 1) 27289 2) 1111 3) 28125 4) 420372

26. Для какого из приведенных слов истинно высказывание?

$$\text{НЕ (Число букв равно 5)} \text{ И (Первая буква согласная)} \text{ ИЛИ НЕ (Предпоследняя буква согласная)}$$

- 1) ПРИМЕР 3) ОТВЕТСТВЕННОСТЬ
2) СПОРТ 4) АЛГОРИТМ

27. Определите последнюю операцию при расчете значения логического выражения для значений $x = 5, y = 4$:

$$\neg (2*x < 11) \wedge y \geq 2 \vee y < 9$$

- 1) Ложь \wedge Истина 3) \neg Ложь
2) Ложь \vee Истина 4) Истина \vee Ложь

28. Высказывание «Только одно из чисел X, Y, Z положительно» соответствует логическому выражению:

- 1) $(X > 0) \text{ ИЛИ } (Y < 0) \text{ ИЛИ } (Z < 0)$
2) $(X > 0) \text{ И НЕ } (Y > 0) \cdot (Z > 0)$
3) $(X > 0) \text{ И } (Y \leq 0) \text{ И } (Z \leq 0) \text{ ИЛИ } (X \leq 0) \text{ И } (Y > 0) \text{ И } (Z \leq 0) \text{ ИЛИ } (X \leq 0) \text{ И } (Y \leq 0) \text{ И } (Z > 0)$
4) $((X > 0) \text{ ИЛИ } (Y \leq 0) \text{ ИЛИ } (Z \leq 0)) \text{ И } ((X \leq 0) \text{ ИЛИ } (Y > 0) \text{ ИЛИ } (Z \leq 0)) \text{ И } ((X \leq 0) \text{ ИЛИ } (Y \leq 0) \text{ ИЛИ } (Z > 0))$

29. В результате работы алгоритма исходная цепочка символов преобразуется в другую по следующим правилам.

Вычисляется длина исходной цепочки и приписывается в ее конец. Затем приписывается исходная цепочка. Из полученной последовательности выбрасывается два первых и один последний символ. Получившаяся цепочка символов и есть результат работы алгоритма. Например, если первоначально имелась цепочка символов *туча*, то результатом работы алгоритма будет *ча4туч*.

Примените дважды алгоритм к исходной цепочке ЗИМА.

Ответ: _____.

30. В результате работы алгоритма исходная цепочка символов преобразуется в другую по следующим правилам.

Перед каждым символом исходной цепочки записываются порядковые номера (перед первым символом — 1, перед следующим — 2 и т. д.). Получившаяся цепочка записывается в обратном порядке. Из полученной последовательности символов выбрасывается два начальных и два последних символа. Получившаяся цепочка символов и есть результат работы алгоритма. Например, если первоначально имелась цепочка символов *y2me*, то результатом работы алгоритма будет *m322*.

Примените дважды алгоритм к исходной цепочке 7ЧУДО.

Ответ: _____.

31. В результате работы алгоритма исходная цепочка символов преобразуется в другую по следующим правилам.

Все цифры исходной числовой цепочки, стоящие на четных позициях, заменяются нулями. Затем первая и последняя цифра меняются местами. Из полученной последовательности удаляется второй символ. Получившаяся цепочка символов и есть результат работы алгоритма. Например, если первоначально имелась цепочка символов *12345*, то результатом работы алгоритма будет *5301*.

Примените дважды алгоритм к исходной цепочке 134679.

Ответ: _____.

32. В таблице приведена стоимость проезда между станциями. Если на пересечении соответствующей строки и столбца с названиями станций нет числа, значит, прямого сообщения между этими станциями нет. Стоимость проезда по маршруту с пересадками складывается из стоимости поездок, составляющих маршрут.

Найдите маршрут между станциями А и В, стоимость которого не больше 9. Если таких несколько, укажите любой из них.

	А	В	С	Д	Е
А		10	5		
В	10		4		2
С	5	4			1
Д					2
Е		2	1	2	

Ответ: _____.

33. В таблице приведена стоимость проезда между станциями. Если на пересечении соответствующей строки и столбца с названиями станций нет числа, значит, прямого сообщения между этими станциями нет. Стоимость проезда по маршруту с пересадками складывается из стоимости поездок, составляющих маршрут.

Определите, с какой из двух станций — С или Е — дешевле всего добраться до станции В не более чем с двумя пересадками. Выпишите получившийся маршрут.

	А	В	С	Д	Е	F
А		10		2		
В	10				14	9
С				1		
Д	2		1			2
Е		14				4
F		9		2	4	

Ответ: _____.

34. В таблице приведены расстояния по железнодорожному пути между станциями. Если на пересечении соответствующей строки и столбца с названиями станций нет числа, значит, прямого сообщения между этими станциями нет.

Найдите самый короткий маршрут с двумя пересадками.

	А	В	С	Д	Е
А			5		3
В				2	9
С	5			1	
Д		2	1		
Е	3	9			

Ответ: _____.

35. У исполнителя Калькулятор в системе команд имеются 2 команды:

- 1) вычесть 1
- 2) умножить на 2

Выполняя команду 1, исполнитель вычитает 1 из имеющегося числа. В результате команды 2 исполнитель умножает на 2 имеющееся у него число. Таким образом, последовательность команд 2212, примененная исполнителем к числу 5, приведет к результату 38:

умножить на 2; умножить на 2; вычесть 1; умножить на 2

$$5 * 2 = 10; 10 * 2 = 20; 20 - 1 = 19; 19 * 2 = 38.$$

Запишите последовательность номеров команд, чтобы получить результат 31 из числа 9 не более чем за 4 шага. Если таких последовательностей несколько, выпишите любую из них.

Ответ: _____.

36. У исполнителя Калькулятор в системе команд имеются 2 команды:

1) вычесть 1

2) умножить на 4

Выполняя команду 1, исполнитель вычитает 1 из имеющегося числа. В результате команды 2 исполнитель умножает на 4 имеющееся у него число. Таким образом, последовательность команд 1122, примененная исполнителем к числу 3, приведет к результату 16:

вычесть 1; вычесть 1; умножить на 4; умножить на 4

$$3 - 1 = 2; \quad 2 - 1 = 1; \quad 1 * 4 = 4; \quad 4 * 4 = 16.$$

Запишите последовательность номеров команд, чтобы получить результат 42 из числа 3 не более чем за 5 шагов. Если таких последовательностей несколько, выпишите любую из них.

Ответ: _____.

37. У исполнителя Калькулятор в системе команд имеются 2 команды:

1) умножить на 3

2) вычесть 4

Выполняя команду 1, исполнитель умножает на 3 имеющееся число. В результате команды 2 исполнитель вычитает 4 из имеющегося у него числа. Таким образом, последовательность команд 2111, примененная исполнителем к числу 7, приведет к результату 81:

вычесть 4; умножить на 3; умножить на 3; умножить на 3

$$7 - 4 = 3; \quad 3 * 3 = 9; \quad 9 * 3 = 27; \quad 27 * 3 = 81.$$

Запишите последовательность номеров команд, чтобы получить результат 38 из числа 6 не более чем за 5 шагов. Если таких последовательностей несколько, выпишите любую из них.

Ответ: _____.

2. ИНФОРМАЦИОННЫЕ И КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ

- Знать:**
- правила подсоединения блоков и устройств компьютера;
 - основные сигналы о готовности и неполадке при включении компьютера;
 - правила безопасной эксплуатации средств ИКТ;
 - принципы организации файловой системы;
 - понятие процесса дискретизации звука и графики;
 - основные звуковые и графические форматы;
 - способы создания звуковых и графических файлов с помощью компьютера;
 - возможности форматирования текстового документа;
 - способы автоматизации редактирования документа;
 - модели представления данных в современных базах данных;
 - структуру таблиц и виды связей между таблицами в базах данных;
 - принципы конструирования сложных графических объектов из простых;
 - инструменты графического редактора для выполнения базовых операций по созданию изображений;
 - способы задания поисковых запросов к поисковым серверам;
 - виды программ, предназначенных для компьютерного черчения;
 - типы информационных моделей;
 - этапы разработки информационной модели;
 - области применения табличного информационного моделирования;
 - структуру электронных таблиц, адресацию ячеек электронных таблиц, форматы данных в электронных таблицах;
 - правила создания формульных выражений в электронных таблицах;
 - категории функций и типы диаграмм;
 - принципы создания презентаций;
 - назначение и функции коммуникационных технологий;
 - структуру адресов веб-ресурсов;
 - правила организации переписки, общения в сети в режиме реального времени.
- Уметь:**
- организовывать сеанс работы на компьютере;
 - организовывать работу с файлами и другими объектами файловой системы;
 - архивировать и разархивировать информацию;
 - защищать информацию от компьютерных вирусов;
 - оценивать количественные показатели информационных объектов и процессов;
 - записывать текстовую, графическую, звуковую информацию;
 - определять основные параметры мультимедийных данных;
 - создавать текстовые документы, выполнять основные операции с фрагментами текста;
 - проводить отбор информации в базах данных;
 - осуществлять ввод графических изображений и их преобразование;
 - выделять в сложных графических объектах простые (графические примитивы);
 - создавать сложные графические объекты с помощью инструментов растрового и векторного графических редакторов;
 - осуществлять поиск информации с помощью поисковых систем;
 - формулировать сложные поисковые запросы;
 - определять результаты применения логических операторов в языках поисковых запросов;

- создавать рисунки, чертежи, графические представления реальных объектов;
- использовать стандартные графические объекты и конструировать новые;
- производить геометрические преобразования и операции обработки графических объектов;
- разрабатывать формальные информационные модели;
- оперировать элементами электронных таблиц;
- вводить и редактировать данные в электронных таблицах;
- определять необходимые форматы данных и форматировать данные в электронных таблицах;
- использовать различную адресацию ячеек и трансляцию формул;
- проводить расчеты с использованием функций Excel;
- строить диаграммы на основании данных электронных таблиц;
- создавать презентации, содержащие различные объекты;
- настраивать анимацию, интерактивность, график показа презентации;
- передавать и получать информацию с помощью электронных средств связи;
- создавать личные коллекции информационных объектов;
- организовывать общение по сети в режиме реального времени.

2.1. ОСНОВНЫЕ УСТРОЙСТВА, ИСПОЛЬЗУЕМЫЕ В ИКТ

- ### 2.1.1. Соединение блоков и устройств компьютера, других средств ИКТ; простейшие операции по управлению (включение и выключение, понимание сигналов о готовности и неполадке и т. д.); использование различных носителей информации, расходных материалов. Гигиенические, эргономические и технические условия безопасной эксплуатации средств ИКТ

Соединение блоков и устройств компьютера

Для подключения устройств к персональному компьютеру необходимо:

1. Подсоединить все внешние устройства к системному блоку с помощью кабелей, прилагающихся к устройствам.
2. Подключить к сети 220 В кабели всех внешних устройств.
3. Подключить кабель электропитания системного блока (сетевой провод) к соответствующему разъему на системном блоке, затем подключить его к сети 220 В.
4. Включить кнопки питания на всех внешних устройствах, планируемых для использования во время работы, — принтере, сканере, звуковых колонках, модеме и т. д. — в соответствии с инструкциями к устройствам. Затем включить электропитание монитора.
5. Если корпус компьютера имеет общий выключатель питания на задней стенке системного блока, включить его. Это нужно сделать только при первом включении, а далее следует пользоваться кнопками лицевой панели.
6. Для включения питания системного блока нажать и отпустить кнопку Power.

Сигналы о готовности и неполадке компьютера

При правильном включении электропитания должен засветиться индикатор питания на передней панели компьютера. Процессор начнет выполнять программу BIOS (базовая система ввода-вывода) материнской платы, которая производит тестирование устройств компьютера и загрузку операционной системы с жесткого диска в оперативную память. Проверка называется POST (Power On Self Test).

Когда тест POST удачно пройден, из динамика компьютера раздается один короткий звуковой сигнал.

Если компьютер не реагирует после включения, в первую очередь необходимо проверить питание компьютера. Следует убедиться, что включены сетевой провод (как в сам компьютер, так и в электросеть) и выключатель на задней стенке компьютера.

Если во время самотестирования при включении компьютера обнаруживается сбой, загрузка прекращается и выводится сообщение об ошибке. При этом динамик системного блока может выдавать последовательности разных по частоте, длине и количеству звуковых сигналов. У каждого звукового сигнала есть свое значение, различающееся для разных версий BIOS. Эти сигналы облегчают поиск неисправности компьютера.

Начало и завершение работы с компьютером

НАЧАЛО РАБОТЫ С ОС WINDOWS

Как только компьютер загрузится, на экране появится окно приветствия *Добро пожаловать*, где нужно выбрать пользователя по его имени или картинке (фотографии). Затем следует ввести пароль (если он был задан во время установки и создания учетных записей пользователя). Символы пароля в поле набора заменяются точками, что позволяет защитить его от рассекречивания во время ввода.

Необходимо помнить, что язык раскладки клавиатуры должен совпадать с тем, который был задан при установке имени и пароля. Для переключения набора на русском, английском или ином (установленном на компьютере) языке в окне приветствия имеется индикатор переключения языков.

После нажатия клавиши Enter откроется персональный *Рабочий стол Windows*. Если компьютер содержит только одну учетную запись без пароля, ОС Windows пропускает окно *Добро пожаловать* и автоматически открывает *Рабочий стол*.

ПЕРЕКЛЮЧЕНИЕ МЕЖДУ ПОЛЬЗОВАТЕЛЯМИ

Быстрое переключение между пользователями позволяет нескольким пользователям одновременно входить в систему. Однако вводить информацию при помощи клавиатуры в определенный момент времени может только один активный пользователь. Когда пользователь «возвращается» в систему, все его настройки Windows автоматически восстанавливаются.

Переключение на сеанс другого пользователя без завершения текущего сеанса возможно следующим образом. Необходимо открыть *Главное меню* и нажать кнопку *Выход из системы*. В появившемся диалоговом окне *Выход из Windows*

следует щелкнуть по кнопке *Смена пользователя* и далее начать сеанс другого пользователя через диалоговое окно приветствия. В окне *Выход из Windows* можно и отказаться от переключения — с помощью кнопки *Отмена*.

ЗАВЕРШЕНИЕ СЕАНСА

Для завершения сеанса работы данного пользователя необходимо открыть диалоговое окно *Выход из Windows* и нажать кнопку *Выход*.

ЗАВЕРШЕНИЕ РАБОТЫ

ВЫКЛЮЧЕНИЕ

Операционная система Windows готовится к выключению путем сохранения настроек сессии и отключения сетевых, модемных и прочих соединений.

Общий порядок выключения компьютера следующий:

- завершить работу программ;
- выключить системный блок: в меню кнопки *Пуск* нажать кнопку *Выключение*. В появившемся диалоговом окне *Выключить компьютер* щелкнуть по кнопке *Выключение*;
- выключить внешние устройства компьютера.

В диалоговом окне *Выключить компьютер* кроме кнопки *Выключение* также имеются следующие кнопки: *Спящий режим*, *Ждущий режим* и *Перезагрузка*.

СПЯЩИЙ РЕЖИМ

Спящий режим сохраняет сессию в отдельном файле на жестком диске перед тем, как выключить питание. Этот режим позволяет выключить компьютер даже в том случае, если на нем запущено много программ и открыто много документов. Когда компьютер включится, *Рабочий стол* будет восстановлен с исходными настройками.

ЖДУЩИЙ РЕЖИМ

Ждущий режим выключает монитор, останавливает работу жестких дисков и вентиляторов, а также переходит в режим низкого энергопотребления. В этом режиме индикатор на корпусе компьютера может мигать или принять желтый цвет. Как только активируются клавиатура или мышь, в течение нескольких секунд компьютер выводится из режима ожидания и возвращается в нормальный рабочий режим. При переходе в ждущий режим данные измененных файлов не сохраняются, поэтому в случае аварийного отключения электропитания все изменения в файлах теряются.

Ждущий режим предназначен не для выключения компьютера, а лишь для временной приостановки его работы, и переходят в него в случае непродолжительного перерыва в работе для экономии энергопотребления (особенно полезно для ноутбуков). На компьютере со 108-клавишной клавиатурой можно перейти в ждущий режим с помощью клавиши *Sleep*.

ПЕРЕЗАГРУЗКА

Вариант *Перезагрузка* завершает работу пользователя, выключает и автоматически перегружает ОС Windows. Эта опция закрывает все программы и просит пользователя сохранить изменения. Перезагрузку обычно осуществляют

в случае установки нового устройства или программного обеспечения, которое требует перезагрузки компьютера. Также данная опция может помочь при неполадках в системе Windows и в случае «зависания» Windows.

БЕЗОПАСНАЯ ЭКСПЛУАТАЦИЯ КОМПЬЮТЕРА

При работе с компьютером необходимо соблюдать следующие меры безопасности:

- для подключения электропитания использовать только сетевые кабели;
- использовать для электропитания компьютера и периферийных устройств только электросети переменного тока напряжением 220 В, частотой 50 Гц с заземлением;
- подключение и отключение периферийных устройств выполнять только при отключенных от сети компьютере и периферийных устройствах;
- не использовать поврежденные сетевые кабели и розетки;
- не перекрывать вентиляционные отверстия компьютера и периферийных устройств;
- не вскрывать корпус монитора, принтера, других периферийных устройств, а также корпус блока питания компьютера;
- перед открытием корпуса компьютера отсоединять его от сети;
- не оставлять без присмотра включенные компьютер и периферийные устройства;
- не допускать попадания на компьютер и периферийные устройства жидкостей и сыпучих веществ;
- ограничивать время работы за компьютером.

2.1.2. Создание, именование, сохранение, удаление объектов, организация их семейств. Файлы и файловая система. Архивирование и разархивирование. Защита информации от компьютерных вирусов

Файлы. Файловая система

Информация, имеющаяся на компьютере, хранится на дисках в *файлах*.

Под файлом понимают логически связанную совокупность данных или программ, для размещения которой во внешней памяти выделяется именованная область.

Файлы могут содержать тексты, аудиозаписи, рисунки, видеофильмы, программы и т. д.

Часть операционной системы (ОС), работающая с файлами и позволяющая найти свободное место при записи нового файла во внешней памяти, а также обеспечивающая хранение данных и доступ к конкретному файлу, называется файловой системой.

Файловая система определяет схему записи информации, содержащейся в файлах, на физический диск.

Диски (гибкие, жесткие, портативные, CD, DVD и т. п.), хранящие файлы, используют разные принципы записи и чтения информации. Диски должны быть отформатированы. Форматирование жесткого диска выполняется в три этапа. Сначала проводят форматирование низкого уровня, затем на диске организуют разделы, после чего проводят форматирование высокого уровня.

Первый этап — *форматирование низкого уровня* — выполняется еще на заводе, и изготовитель указывает только форматную емкость диска. Физическое форматирование всегда выполняется одинаково, независимо от свойств операционной системы.

В накопителях на жестких дисках данные записываются и считываются универсальными головками чтения/записи с поверхности вращающихся магнитных дисков, разбитых на дорожки. **Дорожка** — это одно «кольцо» данных на одной стороне диска. Дорожка записи на диске слишком велика, чтобы использовать ее в качестве единицы хранения информации. Поэтому дорожки на диске разбивают (в процессе форматирования низкого уровня) на нумерованные отрезки, называемые **секторами**. Стандартный размер поля данных сектора для жесткого диска — 512 байт, для оптического — 2048 байт. Количество секторов может быть разным в зависимости от плотности дорожек и типа накопителя. Например, дорожка гибких дисков может содержать от 8 до 36 секторов, а дорожка жесткого диска — от 380 до 700.

Группы секторов, рассматриваемые операционной системой как одно целое, объединяются в кластеры. **Кластер** является наименьшей единицей адресации при обращении к данным. Каждый кластер имеет свой номер. В отличие от размера сектора, размер кластера (число секторов, от 1 до 128) строго не фиксирован. Он выбирается кратным степени числа 2.

При записи файла операционная система выделяет кластеру некоторое пространство на диске. Это пространство выделяется целыми кластерами, так что каждый файл, в зависимости от его размера, занимает один или несколько кластеров. Операционная система сама определяет, какие именно кластеры соответствуют данному файлу и в какие реальные сектора диска при этом производится запись.

Файлы на диске записываются в свободные кластеры, поэтому фрагменты одного и того же файла могут находиться в разных местах диска. Данные о том, в каком месте диска записан тот или иной файл, хранятся в *файловой системе*. Производительность компьютера является наилучшей, когда фрагменты файла занимают кластеры, идущие на диске подряд, т. к. операция поиска физических координат файла при его большой фрагментации довольно затруднительна.

Самой распространенной на сегодняшний день является файловая система NTFS (New Technology File System), хотя иногда еще используется файловая система FAT32. Размер кластера в системе NTFS не зависит от размера диска, и для больших дисков эта система работает значительно эффективнее. Кроме того, она обеспечивает больший уровень быстродействия, чем FAT32, средства защиты, шифрования, восстановления и сжатия информации, контроль над доступом и ряд других возможностей, недоступных ни в одной версии файловой системы FAT.

На втором этапе форматирования на жестком диске создают один или несколько **разделов**. В большинстве случаев при установке на компьютер нового жесткого диска на нем создается один раздел.

Создание нескольких разделов дает возможность разделить информационное пространство диска. Это необходимо, если предполагается использовать на одном компьютере несколько операционных систем (один раздел поддерживает одну файловую систему).

Кроме того, рекомендуется создавать несколько разделов и в целях безопасности: желательно иметь как минимум два раздела — один для хранения служебных файлов операционной системы и прикладных программ, другой — для хранения данных (документов, фотографий, коллекций музыки и фильмов и т. д.), с которыми работает пользователь. Для создания на жестком диске разделов можно использовать как стандартные программы, так и отдельные утилиты.

В каждом разделе можно создать несколько *логических дисков*. Они будут восприниматься как отдельные устройства. Каждое из них обозначается буквой латинского алфавита.

В операционной системе Windows для наименования дисков используются одна буква и двоеточие:

A: — первый дисковод гибких дисков;

B: — второй дисковод (если установлен);

C: — первый жесткий диск (или первый том первого физического жесткого диска). Если на компьютере установлена только одна операционная система (например, Windows), она находится, как правило, именно на этом диске;

D:, ..., Z: — эти буквы выделены для других дисков, томов, устройств CD/DVD, USB-дисков, сетевых, съемных дисков и переносных устройств. Операционная система Windows присваивает буквенные обозначения поочередно (их можно изменять с помощью специальной функции *Управление компьютером*).

После создания разделов необходимо выполнить форматирование высокого уровня, при котором создаются структуры для работы с файлами и данными — оглавление диска и две копии таблицы размещения файлов.

В дальнейшем обслуживание дисков производится с помощью специальных сервисных программ — *утилит*.

ДЕФРАГМЕНТАЦИЯ ФАЙЛОВ

Процесс разбиения файла на фрагменты при записи на диск называется *фрагментацией*. Файлы, которые располагаются в цепочках из несмежных кластеров, называются *фрагментированными*. При большой степени фрагментации замедляется доступ к файлам. *Дефрагментация* — перераспределение файлов на диске, при котором они располагаются в непрерывных областях. Для нее существуют специальные программы — *утилиты дефрагментации*. Дефрагментация чаще всего требуется для файловых систем FAT.

ПРОВЕРКА ДИСКА

Существуют также *утилиты проверки диска*, которые проверяют правильность информации в таблицах распределения файлов и осуществляют поиск сбойных блоков диска. Один из видов нарушений файловой структуры — *потерянные кластеры*: отдельные кластеры или их цепочки, помеченные как занятые, но не принадлежащие ни одному файлу. Возможны также *пересечения цепочек кластеров* — ситуации, когда два файла ссылаются на одни и те же кластеры. Для исправления подобных ошибок программы проверки используют информацию из копий таблиц размещения файлов.

Пример 1

Определите емкость жесткого диска, содержащего 2^{28} кластеров. Количество секторов в кластере — 16.

Решение. Поскольку размер одного сектора 512 байт, емкость диска будет равна:
 $512 \cdot 16 \cdot 2^{28} = 2^9 \cdot 2^4 \cdot 2^{28} = 2^{42} = 2 \cdot 2^{40}$

$2 \cdot 2^{40}$ байт = 2 Тбайт

Ответ: 2 Тбайт.

Пример 2

На жесткий диск записаны 2 файла, имеющие размеры 1375 байт и 162 Кбайт. Сколько кластеров заняли эти файлы, если известно, что размер одного кластера диска — 2048 байт?

Решение. Файл размером 162 Кбайт занимает $162 \cdot 2^{10} : 2048 = 81$ кластер.

Файл размером 1375 байт должен занять $1375 : 2048 = 0,67$ кластера. Но файлы могут занимать только целое число кластеров, поэтому надо округлить 0,67 до ближайшего большего целого числа — 1 кластер.

Вместе оба файла занимают $81 + 1 = 82$ кластера.

Ответ: 82 кластера.

ОБЪЕКТЫ ФАЙЛОВОЙ СИСТЕМЫ WINDOWS

Объектами файловой системы Windows являются *файл* и *папка*. Файловая система обеспечивает возможность доступа к конкретному файлу и позволяет найти свободное место при записи нового файла. Операционная система отвечает за следующие действия, связанные с обслуживанием файловой системы:

- создание файлов и папок, присвоение им имен и их переименование;
- копирование и перемещение файлов между дисками компьютера, между папками одного диска, удаление файлов и папок;
- навигация по файловой структуре с целью доступа к заданному файлу, папке;
- управление атрибутами файлов.

ФАЙЛ

Создание файла включает в себя присвоение ему имени и регистрацию его в файловой системе.

Полное имя файла состоит из двух частей: собственно **имени** и **расширения имени (типа файла)**. Имя от расширения отделяется точкой. Имя файла может включать до 256 любых символов, кроме следующих специальных знаков: \ / : * ? « » < > |, разрешается использовать пробелы и точки. Прописные и строчные буквы не различаются. Рекомендуется давать файлу такое имя, которое отражает суть хранящейся в нем информации.

Расширением имени считаются все символы, идущие после последней точки.

Расширение файла показывает, какого рода информация содержится в данных — рисунок, текст, программа и т. д., и обычно состоит из трех или (реже) четырех букв. Существует огромное множество расширений. Ниже приведены некоторые из них:

exe — «исполняемый» файл, содержащий программу;

cfg — конфигурационный файл, в котором программа указывает параметры своей работы;

dll — динамически подключающаяся библиотека данных, к которой могут обратиться по мере надобности сразу несколько программ;
 sys — системный файл (драйвер устройства);
 lnk — файл ярлыка (указателя на другой файл и проч.);
 hlp — файл справки, а иногда и полное руководство по той или иной программе;
 tmp — временный файл;
 txt, rtf, doc, docx — текстовые файлы;
 htm, html — гипертекстовый документ Интернета;
 xls, xlsx — электронная таблица;
 dat — файл данных;
 wav, mp3 — звуковые файлы;
 mpeg, avi — видеофайлы;
 bmp, jpg, tiff, gif, png — графические изображения;
 arj, zip, rar, 7z — файлы «архивов»;
 ppt, pptx — презентация.

Таким образом, по расширению файла можно легко определить, с помощью какой программы этот файл был создан. Пользователи ОС Windows чаще имеют дело не с расширением файла, а с соответствующим ему графическим значком — **пиктограммой**. Значки имеют разный вид в зависимости от того, к какой именно программе привязан тот или иной тип файла.

Действия со значком (пиктограммой) файла означают действия с самим файлом. Например, удаление значка приводит к удалению файла, копирование — к копированию файла и т. д.

Иногда одним значком могут обозначаться файлы сразу нескольких типов, а расширение файла во всех случаях остается неизменным. По нему ОС автоматически распознает приложение, создавшее этот файл, что позволяет автоматизировать запуск прикладных программ через имя файла.

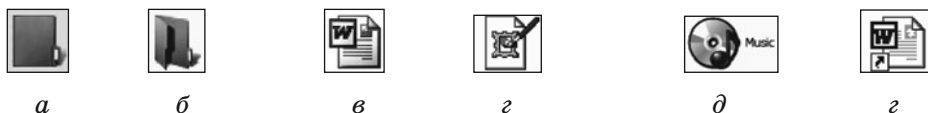


Рис. 2.1. Примеры пиктограмм: а, б — папки; в — текстовый документ; г — графический файл; д — музыкальный файл; е — ярлык текстового документа

ПАПКА

Еще одним важным объектом файловой системы ОС Windows является *папка*. Папка ОС Windows играет ту же роль, что и обычная папка для хранения документов в делопроизводстве: она позволяет упорядочить хранение документов (иногда папку называют *каталогом*). Иными словами, **папка** — это контейнер для программ и файлов.

Файлы объединяют в папки (каталоги) для удобства хранения и поиска. Папки могут быть вложены друг в друга, образуя многоуровневую древовидную структуру.

Корневой каталог (папка) содержит вложенные каталоги (папки) первого уровня, каждый из которых также может содержать вложенные каталоги (папки) второго уровня и т. д. В папках всех уровней могут храниться файлы.

Папка отображается на экране с помощью значка (пиктограммы), имеющего вид канцелярской папки. Помимо файлов папка может содержать другие папки и ярлыки. Размер папки и общий объем ее содержимого можно узнать из свойств папки. Для вывода окна свойств нужно вызвать контекстное меню папки (щелчком правой кнопки мыши) и выбрать команду *Свойства*.

В процессе форматирования диска автоматически создается **корневая папка** — таблица файловой системы диска, содержащая список всех вложенных в нее других папок и всех находящихся на диске файлов. Папка, содержащая другие вложенные в нее папки, по отношению к ним называется **родительской**, а вложенные папки называются **дочерними**.

Правила присвоения имени папке аналогичны именованию файлов, но папкам обычно не задают расширение (ОС никак не реагирует на расширение папки и воспринимает его просто как продолжение имени).

Чтобы указать местонахождение какого-либо файла, пишут его **полное имя**, которое состоит из пути к файлу и имени файла. **Путь** начинается с имени диска, на котором записан файл, затем ставится двоеточие, обратная наклонная черта и перечисляется последовательность всех имен папок (каталогов), которые необходимо открыть, чтобы получить доступ к файлу. Все промежуточные папки разделяются между собой символом «\».

Если корневая папка размещена на логическом диске С, то путь доступа к файлу может быть записан следующим образом:

Обозначение диска	Имя файла
C:\Мои документы\Моя музыка\Образцы\New Storie.wma	
Вложенные папки	Расширение файла

ЯРЛЫК

Важным понятием в среде ОС Windows является *ярлык*. Изображается он обычно значком (пиктограммой) с черной изогнутой стрелкой в левом нижнем углу. Именно такой стрелкой он отличается от значка соответствующего ему объекта.

Ярлык — это небольшой по размеру (1–3 Кбайт) файл с расширением *.lnk*, в котором хранится ссылка (указатель) на какой-либо объект, указывающая его местоположение. Например, можно создать ярлыки для звуковых, графических, архивных файлов, содержащих программы, фильмы и т. п., а также для папок, дисков, принтеров, других компьютеров, интернет-страниц и т. д.

У каждого объекта может быть сколько угодно ярлыков. Объект и его ярлыки обычно располагаются в разных местах. Например, можно разместить файл, запускающий какую-либо программу, в одной папке (чаще всего это одна из папок внутри папки *Program Files*), а ярлык (или несколько ярлыков) к этому файлу — в любом другом месте, например на *Рабочем столе*, в *Главном меню* или в другой папке.

Создание ярлыков позволяет экономить место на жестком диске, избавляя от необходимости создавать множество копий объектов на компьютере.

Ярлыки служат для ускорения открытия папок, запуска программ или документов, доступа к устройствам. Действия, которые можно совершать с ярлыками, аналогичны действиям над файлами (папками). Открыть ярлык — значит открыть (запустить) связанный с этим ярлыком объект. Таким образом, для

пользователя приемы работы с ярлыками ничем не отличаются от приемов работы с их объектами.

Ярлык можно легко создать или удалить, что никак не влияет на связанный с ним объект, — никакие новые объекты или их копии, кроме самого ярлыка, при этом не создаются и не удаляются. Аналогично, копирование или перемещение ярлыка приводит лишь к копированию (перемещению) его самого, а не соответствующего ему объекта.

ДЕЙСТВИЯ С ФАЙЛАМИ И ПАПКАМИ

Открытие файла для ОС означает его подготовку к работе с тем приложением, с которым связан тип этого файла. Подготовка включает в том числе запуск соответствующей программы, в которой данные файла могут быть просмотрены или обработаны. Открытие исполняемого файла означает запуск программы, в нем содержащейся.

Закрытие файла означает разрыв связи между файлом и открывшей его программой.

Изменение (редактирование) файла — это внесение любых изменений в содержание данных файла.

Копирование файла (папки) означает, что в той же папке (но с другим именем) или в другой папке, на том же или другом внешнем носителе создается еще один файл (папка), который является точной копией исходного.

Перемещение файла (папки) означает, что исходный файл (папка) пересылается на другое место (в другую папку или на другой внешний носитель). Новый файл (папка) при этом не возникает, просто исходный файл (папка) меняет свой адрес.

Переименование файла (папки) означает замену старого имени файла на новое. При этом в конкретной папке не может быть двух папок с одинаковым именем или двух файлов с одинаковым полным именем.

Удаление файла или папки является временным. На самом деле они перемещаются при этом в специальную папку — *Корзину*. На уровне жесткого диска ничего не происходит — файлы остаются в тех же кластерах, где были записаны (изменяется только путь доступа к файлам и папкам).

Уничтожение файлов происходит при очистке *Корзины* — при этом файл удаляется из файловой структуры ОС (в таблице размещения файлов он помечается как удаленный), но физически все равно остается там же, где и был. Это сделано для минимизации времени проведения данной операции. Однако при этом появляется возможность записи новых файлов в кластеры, помеченных как «свободные».

ОПЕРАЦИИ С ФАЙЛАМИ, ПАПКАМИ И ЯРЛЫКАМИ В ОС WINDOWS

Чтобы открыть папку, запустить программу или файл, нужно дважды щелкнуть по соответствующему объекту кнопкой мыши.

Если Windows известно (по расширению файла), с какой программой связан именно этот тип файлов, ОС самостоятельно откроет нужный документ в окне программы. Музыкальный файл, например, запустится в Windows Media Player, текстовый документ откроется в *Блокноте*, рисунок — в *Фотоальбоме* и т. д.

Просмотреть содержимое папок, имеющихся на дисках компьютера, можно с помощью системы окон *Мой компьютер* и программы *Проводник*.

Для открытия окна *Мой компьютер* нужно дважды щелкнуть мышью на значке *Мой компьютер* на *Рабочем столе* или в *Главном меню* кнопки *Пуск*.

Правую часть окна *Мой компьютер* занимает рабочее поле, где отображаются значки дисков и других накопителей, установленных на компьютере: жестких дисков, CD (DVD), дисковода гибких дисков, флеш-накопителей и т. д. Диск, на который была установлена операционная система Windows, помечается логотипом Windows. Этот диск называют *системным*.

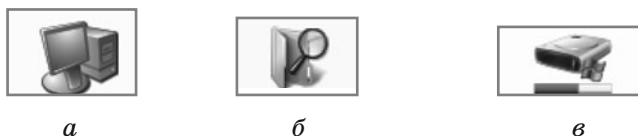


Рис. 2.2. Значки окон: а — Мой компьютер; б — Проводник; в — системный диск

Для вывода информации о том, сколько свободного пространства имеется на каждом жестком диске, нужно в контекстном меню окна *Мой компьютер* выбрать команду *Вид/Таблица*. В основной части окна для каждого диска будут указаны полный объем диска и его свободное пространство. Если дополнительно выбрать команду *Упорядочить значки/Свободно*, список дисков будет отсортирован по порядку возрастания свободного объема каждого диска.

Имя	Тип	Полный объем	Свободно
свободно 0%			
Локальный диск (E:)	Локальный диск	299 ГБ	29,5 ГБ
свободно 10%			
Локальный диск (F:)	Локальный диск	166 ГБ	22,8 ГБ
свободно 40%			
Локальный диск (C:)	Локальный диск	74,5 ГБ	36,0 ГБ
Не определен			
Диск 3,5 (A:)	Диск 3,5		

Рис. 2.3. Отображение объемов дисков в окне *Мой компьютер*

СОЗДАНИЕ ЯРЛЫКОВ

Чтобы создать ярлык нужной программы, например, на *Рабочем столе*, следует через меню *Пуск* найти значок этой программы и перетащить его на *Рабочий стол*, удерживая нажатой правую кнопку мыши. В момент, когда отпускается правая кнопка мыши, открывается контекстное меню, где нужно выбрать опцию *Создать ярлык*.

Для создания ярлыков к любому файлу или папке удобно использовать меню *Файл* программы *Мой компьютер*. Для этого надо установить курсор на названии файла (папки) и выбрать в меню *Файл* команду *Создать ярлык*.

СОЗДАНИЕ ПАПЕК

Чтобы создать папку, например, на *Рабочем столе*, нужно выбрать в контекстном меню пункт *Создать/Папка*. Папка будет создана под именем *Новая папка*, в поле ее названия сразу же будет установлен курсор для переименования (рис. 2.4).

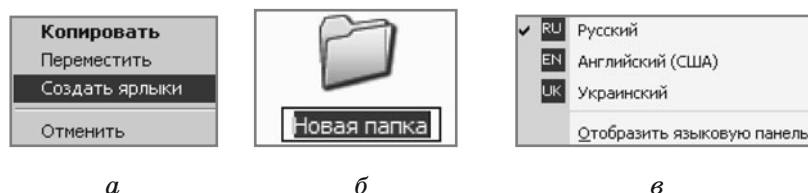


Рис. 2.4. Создание ярлыков и папок: а — контекстное меню перетаскивания файла; б — процесс создания папки

Переименовать папку можно и позже. Для этого в контекстном меню папки нужно выбрать опцию *Переименовать* и ввести в прямоугольнике под папкой ее новое имя. (Чтобы переключить клавиатуру с одного языка на другой, можно использовать установленную комбинацию клавиш — Ctrl+Shift либо Alt+Shift — или щелкнуть мышью по индикатору языка в правой части панели задач и выбрать нужную раскладку.)

Для создания папки в окне *Мой компьютер* можно выбрать команду меню *Файл/Создать/Папку* или контекстного меню *Создать/Папку*. После ее выполнения в конце списка папок и файлов этого окна появится значок с именем *Новая папка*. Имя папки можно сразу изменить.

ВЫДЕЛЕНИЕ ФАЙЛОВ И ПАПЕК

Чтобы выполнять операции с файлами или папками, эти объекты необходимо выделить. Для выделения отдельного файла или одиночной папки нужно один раз кликнуть по ним мышкой. Если необходимо выполнить какие-либо операции с несколькими файлами или папками, следует выделить их все одновременно. Для этого нужно нажать клавишу Ctrl и далее выборочно щелкать по значкам файлов или папок, выделяя их.

Если объекты (файлы и папки) расположены рядом, то можно выделить первый (последний) объект, затем нажать клавишу Shift и указать последний (первый) объект. Еще один способ выделения расположенных рядом объектов — с помощью левой кнопки мыши — протягивание курсора вокруг группы файлов и (или) папок (так называемый метод лассо).

Для одновременного выделения всех без исключения объектов папки можно использовать комбинацию клавиш Ctrl+A.

ПЕРЕИМЕНОВАНИЕ ФАЙЛОВ И ПАПЕК

Переименовать файл или папку можно, выбрав в контекстном меню команду *Переименовать*; затем ввести новое имя и нажать клавишу Enter. Чтобы откатиться от переименования, во время редактирования надо нажать клавишу Esc.

ПЕРЕМЕЩЕНИЕ И КОПИРОВАНИЕ ФАЙЛОВ И ПАПЕК

Для перемещения или копирования объектов их сначала необходимо выделить. Затем выбрать в меню *Правка* команду *Переместить в папку* (или *Копировать в папку*), перейти в требуемую папку и нажать кнопку *Переместить* (соответственно *Копировать*).

Проще, однако, пользоваться следующей последовательностью действий для перемещения, копирования объектов, создания ярлыков с помощью мыши:


- 1) Добиться того, чтобы на экране была видна конечная папка, диск или окно.

- 2) Выбрать объекты для перемещения (копирования).
- 3) Захватив объекты правой кнопкой мыши, переместить их в конечную папку и отпустить кнопку мыши.
- 4) Выбрать в появившемся контекстном меню одну из команд: *Переместить*, *Копировать* или *Создать ярлык*.

Использование правой кнопки мыши намного проще и безопаснее по сравнению с работой левой кнопкой.

УДАЛЕНИЕ ФАЙЛОВ И ПАПЕК

Удалить файлы или папки можно следующими способами:

- в контекстном меню объекта выбрать команду *Удалить*;
- выделить объект и нажать клавишу Delete или нажать на панели инструментов кнопку  *Удалить*;
- захватить объект левой кнопкой мыши и перетянуть его в *Корзину*.

КОРЗИНА

Система Windows сохраняет удаленные файлы в *Корзине*, которая расположена на *Рабочем столе*. Ею можно воспользоваться для восстановления файлов, удаленных по ошибке.

Для просмотра содержимого *Корзины* следует дважды щелкнуть по ее значку. Для восстановления удаленных папок, файлов и ярлыков из *Корзины* нужно выбрать в ней их значки, а затем в контекстном меню или в меню *Файл* выбрать команду *Восстановить*.

Пример 3.

Пользователь работал с папкой *E:\Задания*. Затем он перешел на другой диск в папку *Поездки*, после чего перешел на 2 уровня ниже. Выбрать подходящий вариант полного пути к последней рабочей папке.

- 1) *E:\Поездки\Авиа\Январь*
- 2) *C:\Поездки\Авиа\Январь*
- 3) *V:\Авиа\Январь\Поездки*
- 4) *F:\Поездки\Авиа*

Решение. Поскольку пользователь перешел к другому диску, значит, вариант (1) не подходит. Так как он перешел из папки *Поездки* еще на 2 уровня ниже, не подходят варианты (3) (в нем папка *Поездки* — рабочая) и (4) (в нем произошел переход из папки *Поездки* только на один уровень ниже). Вариант (2) содержит путь к папке, находящейся на 2 уровня ниже, чем папка *Поездки*, и расположенной не на диске E.

Ответ: 2.

Архивирование и разархивирование

Сжатие информации — это процесс преобразования информации файла к виду, при котором уменьшается избыточность в ее представлении и, соответственно, требуется меньший объем памяти для ее хранения.

Сжиматься могут как один, так и несколько файлов, которые в упакованном виде помещаются в так называемый *архивный файл*, или *архив*.

Архив — это специальным образом организованный файл, содержащий в себе один или несколько файлов в сжатом или несжатом виде и служебную информацию об их именах, датах и времени их создания или модификации, размерах и т. п.

Файлы архивов могут иметь формат ZIP, RAR, LZH, ARJ, ARC, TAR и др. в зависимости от программы-архиватора, с помощью которой они были созданы.

Файлы, упакованные в архив, занимают меньше места на диске и быстрее передаются по каналам связи в компьютерных сетях. Кроме того, упаковка группы файлов в один архивный файл значительно упрощает работу с ними: копирование, перенос на другой компьютер и т. п.

Принято различать **архивацию** и **упаковку** (или *компрессию, сжатие*) данных. В первом случае речь идет о слиянии нескольких файлов и даже каталогов в единый файл — архив, во втором — о сокращении объема исходных файлов путем устранения избыточности.

Процесс восстановления файлов из архива (на диск или в оперативную память) точно в таком виде, какой они имели до загрузки в архив, называется **разархивацией (распаковкой)** данных.

Принцип работы программ-архиваторов основан на поиске в файлах избыточной информации и последующем ее кодировании, позволяющем получить минимальный объем хранимых данных. В список задач архиваторов могут входить не только архивация или упаковка файлов, но и сохранение дерева файловой системы, атрибутов и имен файлов, шифровка данных архива, архивация с паролем и т. д.

Степень сжатия зависит от используемой программы, метода сжатия и типа исходного файла. Лучше всего упаковываются файлы графических образов, текстовые файлы и файлы данных, которые могут быть сжаты до 5–40 % от исходного размера. Хуже сжимаются файлы исполняемых программ и загрузочных модулей — до 60–90 %. Почти не сжимаются архивные файлы.

Большие по объему архивные файлы могут быть разбиты на несколько частей. Такие архивы называются **многотомными**, а их составные части — **томами**. Преимущество многотомных архивов в том, что их части можно записать на несколько носителей (диски, флеш-диски и пр.).

Чтобы при распаковке данных не зависеть от наличия соответствующей программы-архиватора, можно создать так называемый **самораспаковывающийся** архивный файл. Это загрузочный (исполняемый) модуль, который самостоятельно разархивирует находящиеся в нем данные при его запуске. Самораспаковывающийся архив называют также *SFX-архивом* (англ. Self-eXtracting). Обычно они создаются в форме EXE-файла.

Наиболее популярными программами-архиваторами на сегодняшний день являются WinZip и WinRAR. «Родным» архивным форматом первого из них является формат ZIP, второго — RAR, несмотря на то что каждый из них поддерживает работу еще с десятками других архивных форматов.

Защита информации от компьютерных вирусов

Компьютерные вирусы — это программы или фрагменты программного кода, которые, попав на компьютер, могут вопреки воле пользователя выполнять различные операции: создавать или удалять объекты, модифицировать файлы, осуществлять действия по собственному распространению по локальным вычислительным сетям или по сети Интернет, создавать всевозможные помехи в работе на компьютере.

Важнейшей функцией компьютерных вирусов является *заражение (инфицирование)* — модификация программных файлов, файлов данных или загрузочных секторов дисков таким образом, что последние сами становятся носителями вирусного кода и в свою очередь могут осуществлять вышеперечисленные операции.

Источниками распространения вирусов чаще всего является Интернет, в том числе электронная почта, внутренние сети интранет (например, частные домашние или офисные сети), съемные носители для обмена информацией между компьютерами.

ПРИЗНАКИ ПРОЯВЛЕНИЯ ВИРУСОВ

При заражении компьютера вирусом очень важно своевременно его обнаружить. Для этого следует знать основные признаки проявления вирусов, первыми из которых являются:

- вывод на экран непредусмотренных сообщений или изображений;
 - подача непредусмотренных звуковых сигналов;
 - самопроизвольный запуск каких-либо программ;
 - самопроизвольные попытки компьютера выйти в Интернет;
 - неожиданное увеличение исходящего интернет-трафика.
- Есть также косвенные признаки наличия вируса в ПК:
- прекращение работы или неправильная работа ранее успешно функционировавших программ;
 - частые зависания и сбои в работе компьютера;
 - медленная работа компьютера при запуске программ;
 - невозможность загрузки операционной системы;
 - исчезновение файлов и каталогов или искажение их содержимого;
 - изменение даты и времени модификации файлов, их размеров;
 - неожиданное значительное увеличение количества файлов на диске;
 - существенное уменьшение размера свободной оперативной памяти;
 - частое обращение к жесткому диску (часто мигает лампочка на системном блоке);
 - браузер «зависает» или ведет себя неожиданным образом (например, окно программы невозможно закрыть) и т. д.

Следует заметить, что вышеперечисленные явления необязательно вызываются присутствием вируса, а могут быть следствием других неполадок. Наличие таких «симптомов» может быть вызвано сбоем в аппаратном или программном обеспечении. Правильная диагностика состояния компьютера нередко затруднена. Однако эти «симптомы» могут быть признаками вируса в компьютере, и при их появлении рекомендуется провести полную проверку компьютера антивирусной программой.

КЛАССИФИКАЦИЯ КОМПЬЮТЕРНЫХ ВИРУСОВ

По типу действий, совершаемых на компьютере пользователя, вредоносные объекты обычно разделяют на две категории: *вредоносные программы* и *потенциально (условно) нежелательные программы*.

Вредоносные программы создаются специально для несанкционированного пользователем уничтожения, блокирования, модификации или копирования информации, нарушения работы компьютеров или компьютерных сетей. К данной категории относятся вирусы и черви, троянские программы и вредоносные утилиты.

Вирусы и черви обладают способностью к несанкционированному саморазмножению, при этом полученные копии также обладают этой способностью.

Черви «переползают» с компьютера на компьютер, используя сети и электронную почту, благодаря чему обладают достаточно высокой скоростью распространения. Помимо сетевых адресов они часто используют данные адресной книги почтовых клиентов. Могут вообще не обращаться к ресурсам компьютера (за исключением оперативной памяти), но, бывает, создают рабочие файлы на дисках системы.

Вирусы заражают другие программы — добавляют в них свой код, чтобы получить управление при запуске зараженных файлов.

Троянские программы не создают свои самовоспроизводящиеся копии, т. е. не являются вирусом в традиционном понимании этого термина (они не заражают другие программы или данные); они не способны самостоятельно проникать на компьютер и распространяются под видом «полезного» программного обеспечения. Однако они уничтожают информацию на дисках, приводят систему к «зависанию», воруют конфиденциальную информацию и т. д. — вред от них может во много раз превышать потери от традиционной вирусной атаки.

Вредоносные утилиты представляют собой инструментарий, созданный для автоматизации создания других вирусов, червей или троянских программ, для взлома, конструкторы вредоносного кода и т. п. Для компьютера, на котором они исполняются, эти утилиты не представляют угрозы. В последнее время наиболее распространенными вредоносными программами стали черви. Далее по распространенности следуют вирусы и троянские программы. Некоторые вредоносные программы совмещают в себе характеристики двух или даже трех из перечисленных выше классов.

Потенциально нежелательные программы разрабатываются и распространяются легально и могут использоваться в повседневной работе, однако обладают функциями, которые могут причинить вред пользователю, но только при выполнении ряда условий. В настоящее время к потенциально (условно) нежелательным программам относят программы классов Adware и Riskware.

Adware — это рекламные программы, предназначенные для показа рекламных сообщений (чаще всего в виде графических баннеров), перенаправления поисковых запросов на рекламные веб-страницы, а также для сбора данных маркетингового характера об активности пользователя. Назначение adware-программ — неявная форма оплаты программ за счет показа пользователю рекламы. Кроме этого, подобные программы, как правило, никак не проявляют своего присутствия в системе — отсутствует значок в системном трее, нет упоминаний об установленных файлах в меню программ. Часто у них нет процедур деинсталляции, используются пограничные с вирусными технологии, позволяющие скрытно внедряться на компьютер пользователя и незаметно осуществлять свою деятельность.

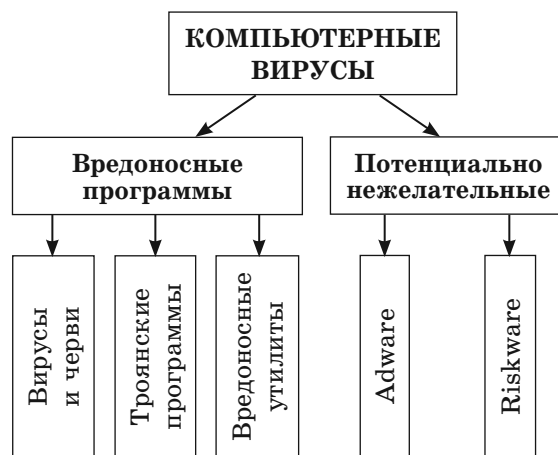


Рис. 2.5. Компьютерные вирусы

Чаще всего такие программы встраиваются в бесплатное и условно-бесплатное ПО (freeware, shareware) или попадают на компьютер пользователя при посещении им «зараженных» веб-страниц. Большинство программ freeware и shareware прекращает показ рекламы после их покупки или регистрации.

К категории *Riskware* относят обычные программы (в том числе свободно продающиеся и широко используемые), которые, тем не менее, способны причинить вред (вызвать уничтожение, блокирование, модификацию или копирование информации, нарушить работу компьютеров или компьютерных сетей).

К категории таких программ относятся программы, имеющие бреши и ошибки, утилиты удаленного администрирования, программы загрузки (скачивания) файлов, автоматического переключения раскладки клавиатуры, мониторы активности компьютерных систем, утилиты для работы с паролями, для остановки процессов или скрытия их работы и проч.

По умолчанию в антивирусных продуктах обнаружение Riskware-программ отключено. При необходимости пользователь сам должен установить опцию поиска подобных программ.

ХАРАКТЕРИСТИКА АНТИВИРУСНЫХ ПРОГРАММ

Антивирусная программа — это программный пакет, предназначенный для эффективной защиты, перехвата и удаления из операционной системы компьютера максимального количества вредоносных (или потенциально вредоносных) программ.

В настоящее время большинство ведущих антивирусных программ сочетает в себе функции постоянной защиты (*антивирусный монитор*) и функции защиты по требованию пользователя (*антивирусный сканер*).

Антивирусный монитор запускается автоматически при старте операционной системы и работает в качестве фонового системного процесса, проверяя на вредоносность действия, совершаемые другими программами. При этом проверяются не только файлы на различных носителях информации, но и оперативная память компьютера. Основная задача монитора — обеспечивать максимальную безопасность при минимальном замедлении работы проверяемых программ. Монитор можно временно отключать — например, когда компьютер отключен от сети и не загружаются никакие файлы с внешних носителей.

Антивирусный сканер запускается самим пользователем и, как правило, заключается в полном или выборочном сканировании присутствующих на жестких и сетевых дисках компьютера файлов, а также в однократной проверке оперативной памяти компьютера. В большинстве случаев антивирусные сканеры гораздо требовательнее к ресурсам компьютера, нежели постоянная антивирусная защита.

2.1.3. Оценка количественных параметров информационных объектов.

Объем памяти, необходимый для хранения объектов

Количество информации можно рассматривать как меру уменьшения неопределенности знания при получении информационных сообщений. Такой подход позволяет количественно измерять информацию.

Впервые *объективный подход* к измерению количества информации был предложен американским инженером Р. Хартли в 1928 г. Позже, в 1948 г., этот подход обобщил создатель общей теории информации К. Шеннон.

Формула Хартли устанавливает связь между количеством возможных равновероятных информационных сообщений N и количеством информации I , которое содержится в этих сообщениях:

$$N = 2^I$$

При *алфавитном подходе* к определению количества информации рассматривают информационное сообщение как последовательность знаков определенной знаковой системы. При этом содержание информации значения не имеет.

По приведенной выше формуле можно рассчитать, какое количество информации I несет каждый из знаков этой системы. Если в алфавите знаковой системы N знаков, то каждый знак несет количество информации:

$$I = \log_2 N$$

Чем больше знаков содержит алфавит, тем больше информации несет один его знак.

Если количество символов алфавита равно N , а количество символов в записи сообщения — M , то информационный объем этого сообщения вычисляется по формуле:

$$I = M \cdot \log_2 N$$

Пример 1

Определить количество информации, которое несет одна буква русского алфавита.

Решение. На практике обычно используются 32 буквы русского алфавита (букву *ё* обычно не применяют), следовательно, количество возможных информационных событий равно 32. Тогда информационный объем одного символа равен:

$$I = \log_2 32 = 5 \text{ (бит)}$$

Ответ: 5 бит.

Если N не является целой степенью 2, то число $\log_2 N$ не является целым числом, и необходимо выполнить округление I в большую сторону. При решении задач в таком случае I можно найти как $\log_2 N'$, где N' — ближайшая к N степень двойки, такая, что $N' > N$.

Пример 2

Определить количество информации для одной буквы английского алфавита.

Решение. В английском алфавите 26 букв. Ближайшая к нему целая степень двойки:

$$N = 26; N' = 32 = 2^5$$

Информационный объем одного символа английского алфавита:

$$I = \log_2 2^5 = 5 \text{ (бит)}$$

Ответ: 5 бит.

Пример 3

Какое минимальное количество двоичных разрядов потребуется для того, чтобы закодировать все десятичные цифры?

Решение. В десятичной системе счисления 10 цифр. Двоичных разрядов 2.

$$M = 10; N = 2.$$

Следовательно, необходимое количество цифр:

$$x = \log_2 10 = 3,3219 \approx 4.$$

(Вариант 2. Чтобы избежать расчета логарифма, можно воспользоваться таким способом: $2^x = 10$; $2^3 = 8$ (мало); $2^4 = 16 > 10$. *Ответ:* $x = 4$.)

Ответ: необходимы 4 двоичные цифры.

Таким образом, мерой объема любого сообщения в компьютере или на каком-либо носителе принято считать общую длину двоичного кода этого сообщения. Такая мера весьма удобна, поскольку она не связана ни с видом сообщения, ни с его смыслом. Текстовое, звуковое или иное сообщение одинаково характеризуется тем количеством двоичных знаков, которым оно представлено в выбранной системе кодировки.

Количество информации в текстовом сообщении

Текстовая информация состоит из букв, цифр, знаков препинания, различных специальных символов. Для кодирования текстовой информации используются различные коды.

Таблица, в которой всем символам компьютерного алфавита поставлены в соответствие порядковые номера, называется **таблицей кодировки**. Существуют различные таблицы кодировок текстовой информации.

Распространенная таблица кодировки **ASCII** (читается «аски», *American Standard Code for Information Interchange* — стандартный американский код для обмена информацией) использует 1 байт для кодов информации. Если код каждого символа занимает 1 байт (8 бит), то с помощью такой кодировки можно закодировать $2^8 = 256$ символов.

Таблица ASCII состоит из двух частей. Первая, *базовая часть*, является международным стандартом и содержит значения кодов от 0 до 127 (для цифр, операций, латинского алфавита, знаков препинания). Вторая, *национальная часть*, содержит коды от 128 до 255 для символов национального алфавита, т. е. в национальных кодировках одному и тому же коду соответствуют различные символы.

32	!	33	"	34	#	35	\$	36	%	37	&	'	39	(40)	41	*	42	+	43		
,	44	-	45	.	46	/	47	0	48	1	49	2	50	3	51	4	52	5	53	6	54	7	55
8	56	9	57	:	58	;	59	<	60	=	61	>	62	?	63	@	64	A	65	B	66	C	67
D	68	E	69	F	70	G	71	H	72	I	73	J	74	K	75	L	76	M	77	N	78	O	79
P	80	Q	81	R	82	S	83	T	84	U	85	V	86	W	87	X	88	Y	89	Z	90	[91
\	92]	93	^	94	_	95	`	96	a	97	b	98	c	99	d	100	e	101	f	102	g	103
h	104	i	105	j	106	k	107	l	108	m	109	n	110	o	111	p	112	q	113	r	114	s	115
t	116	u	117	v	118	w	119	x	120	y	121	z	122	{	123		124	}	125	~	126	▯	127

Рис. 2.6. Фрагмент таблицы кодировки ASCII (базовая часть — символы с кодами 32–127)

В настоящее время существует несколько различных кодировок второй части таблицы для кириллицы — KOI8-P, KOI8-U, Windows, MS-DOS, Macintosh, ISO. Наиболее распространенной является таблица кодировки Windows-1251.

Из-за разнообразия таблиц кодировки могут возникать проблемы при переносе русского текста между компьютерами или различными программами.

Ђ	Ѓ	„	ѓ	„	…	†	‡	€	‰	Љ	<
128	129	130	131	132	133	134	135	136	137	138	139
Њ	Ќ	Ќ	Ќ	Ћ	„	„	„	„	•	–	—
140	141	142	143	144	145	146	147	148	149	150	151
□	™	Љ	>	њ	ќ	ћ	џ		Ў	ў	Ј
152	153	154	155	156	157	158	159	160	161	162	163
х	ѓ	„	§	€	©	€	«	¬		®	ї
164	165	166	167	168	169	170	171	172	173	174	175
о	±	І	і	ѓ	™	¶	•	ё	№	€	»
176	177	178	179	180	181	182	183	184	185	186	187
ј	Ѕ	ѕ	ї	А	Б	В	Г	Д	Е	Ж	З
188	189	190	191	192	193	194	195	196	197	198	199
И	Й	К	Л	М	Н	О	П	Р	С	Т	У
200	201	202	203	204	205	206	207	208	209	210	211
Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
212	213	214	215	216	217	218	219	220	221	222	223
а	б	в	г	д	е	ж	з	и	й	к	л
224	225	226	227	228	229	230	231	232	233	234	235
м	н	о	п	р	с	т	у	ф	ч	ц	ч
236	237	238	239	240	241	242	243	244	245	246	247
ш	щ	ъ	ы	ь	э	ю	я				
248	249	250	251	252	253	254	255				

Рис. 2.7. Таблица кодировки Windows CP-1251
(национальная часть — символы с кодами 128–255)

Поскольку объем в 1 байт явно мал для кодирования разнообразных и многочисленных символов мировых алфавитов, была разработана система кодирования **Unicode**. В ней для кодирования символа отводится 2 байта (16 бит). Это означает, что система позволяет закодировать $2^{16} = 65\,536$ символов. Полная спецификация стандарта Unicode включает в себя все существующие, вымершие и искусственно созданные алфавиты мира, а также множество математических, музыкальных, химических и прочих символов.

А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К
1040	1041	1042	1043	1044	1045	1025	1046	1047	1048	1049	1050
Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц
1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062
Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	а	б	в
1063	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073	1074
г	д	е	ё	ж	з	и	й	к	л	м	н
1075	1076	1077	1105	1078	1079	1080	1081	1082	1083	1084	1085
о	п	р	с	т	у	ф	х	ц	ч	ш	щ
1086	1087	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097
ъ	ы	ь	э	ю	я						
1098	1099	1100	1101	1102	1103						

Рис. 2.8. Фрагмент таблицы кодировки UNICODE Cyrillic
(русскоязычные символы)

► Пример 4

Оценить информационный объем (количество информации) следующего сообщения: *Бит — единица количества информации.*

Решение. В сообщении 36 символов (включая точку, тире и пробелы). Если сообщение будет закодировано с помощью таблицы кодов ASCII, то для кодирования потребуется 36 байт, т. е. информационный объем сообщения — 288 бит. Если же будет применяться система кодирования Unicode, то объем сообщения окажется вдвое больше и будет составлять 72 байта, или 576 бит.

Количество графической информации

Растровое графическое изображение состоит из отдельных точек — **пикселей**, образующих строки и столбцы.

Основные свойства пикселя — его расположение и цвет. Значения этих свойств кодируются и сохраняются в видеопамяти компьютера.

Качество изображения зависит от *пространственного разрешения* и *глубины цвета*.

Разрешение — величина, определяющая количество точек (пикселей) на единицу площади.

Глубина цвета — объем памяти (в битах), используемой для хранения и представления цвета при кодировании одного пикселя растровой графики или видеоизображения.

Для графических изображений могут использоваться различные **палитры** — наборы цветов. Количество цветов N в палитре и количество информации I , необходимое для кодирования цвета каждой точки, связаны соотношением:

$$N = 2^I$$

Например, для черно-белого изображения палитра состоит из двух цветов. Можно вычислить, какое количество информации необходимо, чтобы закодировать цвет каждой точки (пикселя):

$$2 = 2^I \Rightarrow I_{\text{пикселя}} = 1 \text{ бит}$$

Чтобы определить информационный объем видеоизображения, необходимо умножить количество информации одного пикселя на количество пикселей в изображении:

$$I = I_{\text{пикселя}} \cdot X \cdot Y,$$

где X — количество точек изображения по горизонтали, Y — количество точек изображения по вертикали.

Существует несколько цветовых моделей для количественного описания цвета. В основе модели **RGB** (сокращение от *англ. Red, Green, Blue*) лежат три основных цвета: красный, зеленый и синий. Все другие цвета создаются с помощью смешения их оттенков. Например, при смешивании красного и зеленого цветов получим желтый, красного и синего — пурпурный, зеленого и синего — бирюзовый. Если смешать все три основных цвета максимальной яркости, получим белый цвет.

Если один цвет имеет 4 оттенка, то общее количество цветов в модели RGB будет составлять $4 \cdot 4 \cdot 4 = 64$. При 256 оттенках для каждого цвета общее количество возможных цветов будет равно $256 \cdot 256 \cdot 256 = 16\,777\,216 \approx 16,7$ млн.

В современных компьютерах для представления цвета обычно используются от 2-х до 4-х байт. Два байта (16 бит) позволяют различать 2^{16} , то есть

65 536 цветов и оттенков. Такой режим представления изображений называется *High Color*. Четыре байта (32 бита) обеспечивают цветовую гамму в 2^{32} , то есть 4 294 967 296 цветов и оттенков (приблизительно 4,3 миллиарда). Такой режим называется *True Color*.

В графических редакторах применяются и другие цветовые модели. Например, модель **СМУК** — она основана на цветах, получающихся при отражении белого света от предмета: бирюзовом (*англ. Cyan*), пурпурном (*англ. Magenta*), желтом (*англ. Yellow*). Эта модель применяется в полиграфии, где чаще всего употребляется черный цвет (ключевой, *англ. Key*).

Пример 5

Черно-белое растровое графическое изображение имеет размер 20×12 точек. Каков информационный объем изображения?

Решение. В черно-белой палитре два цвета, следовательно, глубина цвета 1 бит:

$$I_{\text{пикселя}} = \log_2 N = \log_2 2 = 1$$

Информационный объем изображения I найдем по формуле:

$$I = I_{\text{пикселя}} \cdot X \cdot Y = 1 \cdot 20 \cdot 12 = 240 \text{ (бит)}$$

Ответ: 240 бит.

Пример 6

Определить объем изображения, если известно, что оно создано в режиме True Color и занимает весь экран в 1280×768 пикселей.

Решение. Найдем объем изображения как произведение количества пикселей в нем на количество байтов в кодировке цвета:

$$1280 \cdot 768 \cdot 4 \text{ байт} = 3\,932\,160 \text{ байт} = 3840 \text{ Кб} = 3,75 \text{ Мб.}$$

Ответ: объем изображения 3,75 Мб.

Пример 7

Каково максимально возможное количество цветов для изображения на экране монитора размером 128×128 точек, если для него выделено 16 Кб оперативной памяти?

Решение. Всего точек в изображении: $128 \cdot 128 = 2^7 \cdot 2^7 = 2^{14}$.

Объем памяти: 16 Кб = $2^4 \cdot 2^{10}$ байт = $2^4 \cdot 2^{10} \cdot 2^3$ бит = 2^{17} бит.

Глубина цвета: $2^{17} / 2^{14} = 2^3 = 8$ (бит).

Количество цветов: $2^8 = 256$.

Ответ: 256 цветов.

Измерение объемов звуковой информации

Звук является непрерывным сигналом. Для использования звука в компьютере его преобразуют в цифровой сигнал. Это преобразование называется **дискретизацией**: для кодирования звука производят его измерение с определенной частотой (несколько раз в секунду).

Частота дискретизации и **точность** представления измеренных значений определяют качество представления звука в компьютере. Чем выше частота дискретизации и чем больше количество разных значений, которыми можно характеризовать сигнал, тем выше качество отображения звука.

В современных компьютерах обычно применяется частота дискретизации в 22 кГц или 44,1 кГц (1 кГц — это тысяча измерений за 1 секунду), а для

представления значения сигнала выделяются 2 байта (16 бит), что позволяет различать 2^{16} , то есть 65 536 значений.

2.1.4. Оценка количественных параметров информационных процессов.

Скорость передачи и обработки объектов; стоимость информационных продуктов, услуг связи

Скорость передачи информации (информационных сообщений) — количество информации, переданное в единицу времени (т. е. объем двоичного кода, переданного за 1 секунду).

Скорость передачи информации обычно измеряется в битах за секунду (*бод*). Кроме того, используются кратные единицы Кбит/с и Мбит/с. Однако иногда в качестве единицы измерения применяют байт в секунду (байт/с) и кратные ему единицы Кбайт/с и Мбайт/с (табл. 2.1).

Таблица 2.1

Отношения между единицами измерения скорости передачи информации

Название	Обозначение		Величина
бит за секунду	бит/с	bps	1 бит/с
килобит за секунду	Кбит/с	Kbps	$2^{10} = 1024$ бит/с
мегабит за секунду	Мбит/с	Mbps	2^{20} бит/с
гигабит за секунду	Гбит/с	Gbps	2^{30} бит/с
терабит за секунду	Тбит/с	Tbps	2^{40} бит/с
байт за секунду	байт/с	Bps	1 байт/с = 8 бит/с
килобайт за секунду	Кбайт/с	KBps	$2^{10} = 1024$ байт/с
мегабайт за секунду	Мбайт/с	MBps	2^{20} байт/с
гигабайт за секунду	Гбайт/с	GBps	2^{30} байт/с
терабайт за секунду	Тбайт/с	TBps	2^{40} байт/с

Пример 8

Данные объемом 2 Кбайта были переданы через некоторое соединение за 128 миллисекунд. Какое время (в миллисекундах) может занять передача через то же соединение данных объемом 512 байт?

Решение. При решении необходимо учитывать соотношения между различными единицами измерения. Лучше записывать все числа в виде степеней двойки:

$$128 \text{ мс} = 2^7 \text{ мс}$$

$$2 \text{ Кбайта} = 2 \cdot 2^{10} \text{ байт} = 2^{11} \text{ байт}$$

$$512 \text{ байт} = 2^9 \text{ байт}$$

Определим время передачи одного байта через соединение:

$$2^7 : 2^{11} = 2^{-4}$$

Время второй передачи данных может составить:

$$2^{-4} \cdot 2^9 = 2^5 = 32 \text{ (мс)}.$$

Ответ: 32 мс.

2.2. ЗАПИСЬ СРЕДСТВАМИ ИКТ ИНФОРМАЦИИ ОБ ОБЪЕКТАХ И О ПРОЦЕССАХ ОКРУЖАЮЩЕГО МИРА

2.2.1. Запись изображений и звука с использованием различных устройств

Графическая и звуковая информация может быть представлена в *аналоговой* или *дискретной* форме. При *аналоговом* представлении физическая величина принимает *бесконечное* множество значений, непрерывно изменяющихся во времени. При *дискретном* представлении физическая величина принимает *конечное* множество значений, и значения ее изменяются скачкообразно.

Для компьютерной обработки звука и изображений они должны быть представлены в цифровой, дискретной форме. Преобразование непрерывных изображений и звука в набор дискретных закодированных значений называется **дискретизацией**.

Запись звука

Звук — это звуковая волна с непрерывно меняющимися *амплитудой* и *частотой*. **Частота** сигнала определяет тон звука: чем она больше, тем тон звука выше. **Амплитуда** определяет громкость звука: чем она больше, тем звук громче.

Для **оцифровки** (дискретизации) звука он разбивается на составляющие, каждой из которых присваивается числовой код. Дискретизацию звукового сигнала осуществляет *звуковая карта* компьютера (или *видеоадаптер*), а именно — ее аналого-цифровой преобразователь (АЦП).

Звуковая карта представляет собой небольшую плату с набором микросхем со специальными разъемами для подключения микрофона, динамиков, клавиатуры и других устройств. Звуковые карты предоставляют широкий спектр возможностей работы со звуком, например, запись звука с микрофона или конструирование сложных полифонических мелодий.

Запись звука происходит через *микрофон*, который создает непрерывный электрический сигнал, а воспроизведение — через *динамики* (*акустические колонки*) или *наушники*, которые звучат также под действием непрерывного электрического сигнала. В результате происходит преобразование аналоговой формы представления звука в дискретную и обратное преобразование из дискретной формы в аналоговую. Первый процесс называется **аналого-цифровым преобразованием** (АЦП), второй — **цифро-аналоговым преобразованием** (ЦАП). Звуковая карта совмещает функции АЦП и ЦАП.

Преобразователь АЦП через определенные интервалы времени (с определенной частотой) измеряет уровень звукового сигнала на входе и сохраняет эту числовую величину на диске. Последовательность этих чисел и составляет звуковой файл.

Чем чаще измеряется уровень звукового сигнала, тем точнее цифровой сигнал воспроизводит форму аналогового. Этот параметр называется **частотой дискретизации**, или **частотой сэмплирования**. Она может принимать значения от 8 до 48 кГц.

Напряжение на входе звуковой карты измеряется с некоторой точностью, зависящей от разрядности сохраняемого числа. Этот параметр называется **глубиной**, или **разрядностью звука** (битрейтом, *англ.* bit rate). Современные 16-битные звуковые карты обеспечивают 16-битную глубину кодирования звука, что дает возможность кодирования $65\,536 (= 2^{16})$ различных уровней громкости.

На аудиодисках звуковая информация представлена с частотой дискретизации 44,1 кГц (вдвое выше того, что может слышать человеческое ухо) и глубиной звука 16 бит. Современные профессиональные звуковые карты могут записывать звук с частотой 96 кГц и 4-байтной глубиной звука (и даже выше), что обеспечивает высокое качество кодирования.

После оцифровки звука используют специальные программы редактирования звуковых файлов (редакторы звукозаписей) для монтажа музыки, ее реставрации, добавления спецэффектов и т. п.

Оцифрованный звук представляется в звуковых редакторах в наглядной форме, поэтому операции копирования, перемещения и удаления частей звуковой дорожки легко осуществить с помощью мыши. Кроме того, можно накладывать звуковые дорожки друг на друга (*микшировать* звуки) и применять различные акустические эффекты (эхо, воспроизведение в обратном направлении и др.).

Звуковые редакторы позволяют также изменять качество цифрового звука и объем звукового файла, изменяя частоту дискретизации и глубину кодирования. Оцифрованный звук можно сохранять в звуковых файлах в универсальном формате без сжатия WAV или в формате со сжатием MP3.

Непосредственная запись звуковой волны приводит к большим размерам файлов. Поэтому было разработано множество форматов звуковых файлов, позволяющих сохранять звук более компактно. Файлы со сжатием называются *lossy* (с потерей качества — например, MP3, OGG, WMA), файлы без потери качества — *loseless* (например, WAV, FLAC). В *lossy*-форматах применяется не только сжатие, но и специальное кодирование звука, выбрасывается часть информации, недоступная человеческому слуху, которую нельзя будет восстановить. В некоторых *loseless*-форматах также происходит сжатие, но обратимое.

Наиболее часто используемые форматы звуковых файлов:

- *WAVE (.wav)* — широко распространенный формат для хранения звуковых файлов. Файлы в этом формате имеют большой размер, зависящий от частоты дискретизации, глубины звука, его продолжительности, моно- или стереозвуча.
- *MPEG-3 (.mp3)* — наиболее популярный формат звуковых файлов. При кодировании из мелодии удаляются звуки, не воспринимаемые человеческим ухом (обычно человек воспринимает диапазон от 20 Гц до 20 кГц).
- *MIDI (.mid)* — эти файлы содержат не сам звук, а команды для его воспроизведения (звук синтезируется). Если звуковая карта не содержит синтезатора, то такой звук воспроизводиться не будет.
- *Real Audio (.ra, .ram)* — разработан для воспроизведения звука в Интернете в режиме реального времени. Имеет не очень высокое качество записи, но небольшой размер, который достигается методами сжатия.
- *MOD (.mod)* — музыкальный формат, сохраняющий образцы оцифрованного звука. Эти образцы можно использовать как шаблоны для собственных звуковых произведений. Файлы в этом формате содержат набор образцов звука, ноты и информацию о длительности. Каждая нота воспроизводится с помо-

щью одного из имеющихся в начале файла звуковых шаблонов. Такой файл, в отличие от MIDI-файла, полностью задает звук, что позволяет воспроизводить его на любой компьютерной платформе.

► Пример 1

Звук воспроизводится в течение 5 секунд при частоте 44,1 кГц и глубине звука 8 бит. Определить его размер (в Мб).

Решение. $44\,100 \text{ Гц} \cdot 5 \text{ с} \cdot 8 \text{ бит} = 1\,764\,000 \text{ бит} = 220\,500 \text{ байт} = 215 \text{ Кб} = 0,2 \text{ Мб}$.

Ответ: 0,2 Мб.

Запись изображений

К компьютеру могут быть подключены **цифровые фотокамеры, видеокамеры и ТВ-тюнеры**, что позволяет получать фотоснимки и видеоизображение непосредственно в цифровом (компьютерном) формате.

Для получения высококачественных фотографий применяются цифровые фотоаппараты. Для сохранения фотографий они используют модули флеш-памяти или жесткие диски маленького размера. Запись изображений на жесткий диск компьютера может осуществляться путем подключения камеры к компьютеру.

Если установить в компьютер специальную плату (ТВ-тюнер) и подключить к ее входу телевизионную антенну, то появляется возможность просматривать телевизионные передачи непосредственно на компьютере.

Для трансляции видео по компьютерным сетям используются **веб-камеры**.

Существуют программы, предназначенные для просмотра и организации изображений (так называемые *графические просмотрщики*). Они осуществляют просмотр, упорядочение и публикацию цифровых фотографий, конвертирование графических файлов из одного формата в другой и др. Наиболее развитые программы этого класса содержат ряд инструментов для обработки изображений, в том числе и пакетной. Примерами таких программ являются ACDSee, XnView, IrfanView, Picasa.

2.2.2. Запись текстовой информации с использованием различных устройств

Текстовая информация — информация, выраженная с помощью естественных и формальных языков в письменной или печатной форме.

Для ручного ввода текстовой информации чаще всего используются **клавиатура** и **мышь**. Для голосового ввода — **микрофон** вместе с соответствующими современными программами распознавания голоса.

Сканирование

Для оптического ввода в компьютер и преобразования в электронную форму текстовые документы и изображения (фотографии, рисунки, слайды) необходимо оцифровать. Технологический процесс оцифровки, в результате которого создается графический образ бумажного документа, называется **сканированием**. Устройства, способные к оцифровке изображений, — планшетный, листовой, слайдовый и барабанный сканеры, цифровые камеры, платы ввода видеоданных и др.

Графический файл, полученный в результате сканирования, хранит растровое изображение исходного документа (состоящее из точек).

Распознавание текста

После обработки документа сканером получается графическое изображение документа (*графический образ*), т. е. набор разноцветных точек. Сканированный текст нельзя редактировать, выбрать из него фрагмент и т. п. Для того, чтобы редактирование стало возможным, следует перевести сканированный текст из графического в текстовый формат. Программы, способные выполнять эту операцию, называются **программами распознавания текста**, или *OCR* (англ. Optical Character Recognition — оптическое распознавание символов).

Проблема распознавания текста является весьма сложной. Современные алгоритмы распознавания текста не ориентируются ни на конкретный шрифт, ни на конкретный алфавит. Большинство программ способно распознавать текст на нескольких языках, представленный в различных формах (например, таблицах, нескольких колонках) и с различным качеством печати.


Лидером среди программ распознавания, которые поддерживают работу с русским языком, является FineReader российской компании АBBYY. Программа производит распознавание текста более чем со 180 языков, открывает файлы многих графических форматов (TIFF, JPG, PFD, PNG, DjVu и др.). Встроена возможность распознавания изображений с цифровых фотоаппаратов.

Еще одна система — OCR CuneiForm — бесплатная программа сканирования и распознавания текста российской компании Cognitive Technologies. Она обеспечивает распознавание текста с сохранением исходного вида документов на более чем 20 языках, в том числе русском, а также распознавание смешанного русско-английского текста.

Текстовые документы предпочтительнее сканировать из системы распознавания, а не с помощью запуска «родной» программы сканера, поскольку та все равно будет вызвана для процесса сканирования, а после его завершения документ предстоит распознавать.

Рассмотрим процесс распознавания документов на примере программы АBBYY FineReader.

При установке программы на компьютер команды ее запуска добавляются во многие контекстные меню. Можно открыть изображение документа прямо из окна *Проводника* или *Моего компьютера*, выбрав из контекстного меню команду *Открыть с помощью АBBYY FineReader*.

Также при установке программа интегрируется в приложения MS Office. В таких программах, как MS Word, MS Excel, появится панель *АBBYY FineReader*. Для распознавания документов непосредственно из MS Word или MS Excel нужно нажать кнопку  этой панели, проверить в диалоговом окне установленные опции и нажать кнопку *Старт*. Будет запущена программа FineReader, а распознанный ею текст будет передан обратно в MS Word или MS Excel.

В стартовом окне FineReader (рис. 2.9) содержится список наиболее распространенных *сценариев обработки документов* (последовательностей операций). Например, сценарий *Сканировать в PDF* содержит набор операций: «отсканировать — распознать — сохранить в PDF». Наиболее часто используется сценарий *Сканировать в Microsoft Word* — он предлагает отсканировать документ, распознать его содержимое и передать в текстовый редактор Word. Можно выполнить требуемые действия и без выбора сценария — запуская самостоятельно нужные этапы.



Рис. 2.9. Стартовое окно программы ABBY FineReader

После запуска сценария на экране появляется панель его выполнения, список шагов сценария, а также подсказки и предупреждения (рис. 2.10).

Слева на боковой панели (1) видны значки (образы) страниц распознаваемого документа и ход распознавания. Выбранная в этой панели страница отображается в основной части окна слева (2). На этом изображении программа автоматически выделяет области с текстом, картинками, таблицами. К каждой из них будут применены различные методы распознавания.

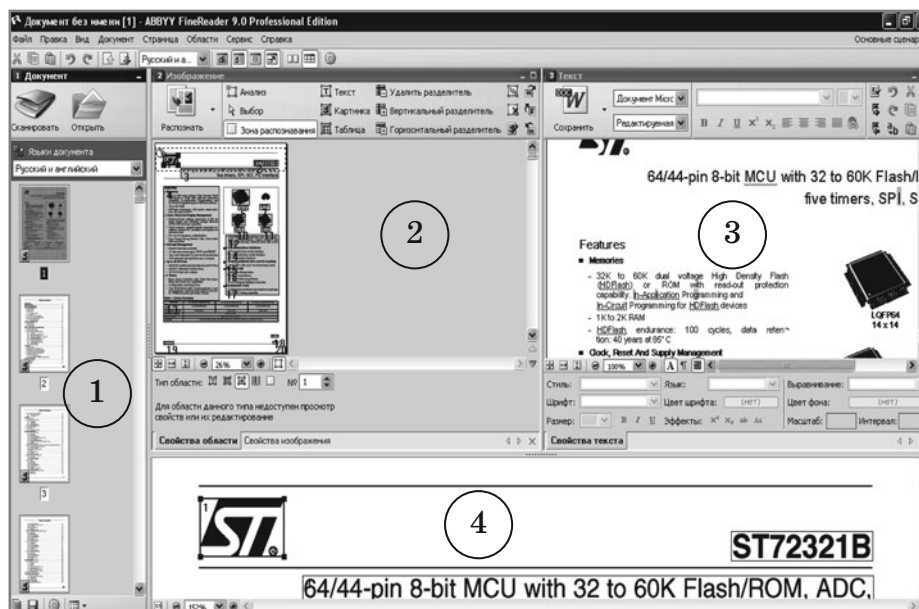


Рис. 2.10. Основное окно программы ABBY FineReader: 1 — панель страниц; 2 — панель Изображение; 3 — панель Текст; 4 — панель Крупный план

При желании пользователь может выбрать язык документа в левой панели. Для многоязычных документов можно указать несколько языков. Но это делать вовсе не обязательно — программа может определить язык автоматически.

Для запуска распознавания выбранной страницы следует нажать кнопку *Распознать* вверху окна *Изображение* (2) или выбрать команду меню *Документ / Распознать документ*. Если требуется распознать сразу все страницы документа — в выпадающем списке кнопки *Распознать* имеется команда *Распознать документ*.

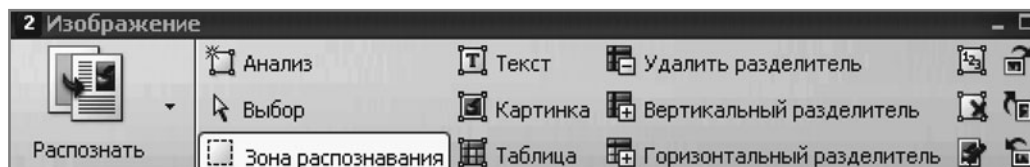


Рис. 2.11. Панель инструментов окна *Изображение*

Результат распознавания будет отображен в основной части программы справа — в панели *Текст* (3). В ней можно отредактировать документ, исправить замеченные ошибки и нераспознанные символы. При установке курсора в панели *Текст* на какое-либо слово — в панели *Крупный план* (4) в нижней части экрана показывается увеличенное изображение слова.

Неточно распознанные символы (в правильности распознавания которых FineReader не уверен) будут выделены цветом. Такие слова можно проверить с помощью встроенной проверки орфографии *Проверка* (вызывается с помощью основного меню *Сервис/Проверка*). При этом показывается ошибочное слово, его изображение в документе и варианты замены.

Результаты распознавания можно сохранить в файл, передать в указанное приложение, скопировать в буфер обмена или отправить по электронной почте. Сохранить можно все страницы документа или только выбранные. Для сохранения документа служит кнопка *Сохранить* вверху окна *Текст* (3).

2.2.3. Запись музыки с использованием различных устройств

Для создания цифровой музыки можно записывать звук с **микрофона** или **линейного входа** звуковой карты, используя компьютер как компактную студию звукозаписи. Еще один способ — записывать музыку непосредственно в музыкальных программах (*секвенсорах*), переправляя в компьютер ноты с **синтезатора** или **MIDI-клавиатуры**.

2.2.4. Запись таблиц результатов измерений и опросов с использованием различных устройств

Компьютерные системы для сбора, проверки, интеграции и анализа информации, относящейся к земной поверхности, называются **геоинформационными системами** (ГИС).

Использование компьютера и других электронных устройств, например дигитайзера или сканера, позволяет проводить подготовку исходных данных к записи, или кодированию точек, линий и областей для их дальнейшего использования. Кроме того, источниками могут быть готовые цифровые карты, цифровые модели рельефа и др.

Специализированные картографические сканеры большого формата используют растровый подход. Карта прикрепляется к вращающемуся барабану, а чувствительный датчик прибора перемещается под прямым углом к направлению вращения. Таким образом, сканируется вся карта, линия за линией и в результате создается подробное растровое изображение всей карты. Возможности анализа данных, заложенные в ГИС, позволяют создавать картографические модели реальных объектов.

Измерять и вводить в компьютер информацию об окружающем мире позволяют самые разнообразные датчики — расстояния, освещенности, температуры, силы, влажности и др. Зачастую они используются в различных виртуальных компьютерных лабораториях.

2.3. СОЗДАНИЕ И ОБРАБОТКА ИНФОРМАЦИОННЫХ ОБЪЕКТОВ

2.3.1. Создание текста посредством квалифицированного клавиатурного письма с использованием базовых средств текстовых редакторов.

Работа с фрагментами текста. Страница. Абзацы, ссылки, заголовки, оглавления. Проверка правописания, словари. Включение в текст списков, таблиц, изображений, диаграмм, формул

Для организации работы с текстами существует множество программ, например WordPerfect, OpenOffice.org Writer, AbiWord. Здесь и далее приведено описание работы с документами в окне текстового процессора MS Word, однако большинство правил верны и при работе в любом другом текстовом редакторе.

Правила набора текста и составления документов

Профессиональный набор и оформление документов предполагают соблюдение определенных правил работы.

- Ввод текста производят с клавиатуры. Позиция ввода подсвечивается **курсором ввода** — мигающей вертикальной чертой.
- Текст следует набирать последовательно, отделяя слова и предложения **только одним пробелом**. Если в тексте нужны некие специальные промежутки, нельзя ставить несколько пробелов подряд — для этого существуют табуляция, различные виды выравнивания текста, отступы (выступы) и поля документа.
- В конце каждого абзаца (и только в этом случае!) следует нажимать **клавишу Enter** — при этом в текст будет вставлен специальный (невидимый) символ,

и курсор будет переведен в следующий абзац. Внутри текущего абзаца курсор будет перемещаться на следующую строку автоматически. При крайней необходимости начать новую строку не завершая абзаца используется комбинация клавиш Shift+Enter.

- **Отступ первой строки** абзаца («красная строка») не выполняется с помощью пробелов или нажатия клавиши Tab — его следует установить, например, с помощью линейки.
- **Центрирование заголовка** выполняют только с помощью выравнивания абзаца по центру. При изменении текста или размера шрифта заголовков будет по-прежнему оставаться посередине строки. Для этой цели не применяются пробелы перед текстом заголовка.
- Для набора **текста с новой страницы** предназначена комбинация клавиш Ctrl+Enter.
- **Знаки препинания** в документе расставляются по следующим правилам:
 - а) запятая, точка, двоеточие, точка с запятой, вопросительный и восклицательный знаки набираются вслед за словом без пробела;
 - б) кавычки и скобки также набираются без пробелов, вплотную к тексту, который они обрамляют;
 - в) тире обрамляется пробелами с обеих сторон, дефис ставится без пробелов.Правильная расстановка знаков препинания позволит, например, избежать такой ситуации, когда точка или запятая располагается не в конце строки, а автоматически переносится на следующую.
- В некоторых ситуациях требуется, чтобы два слова находились обязательно рядом на одной строке (например, инициалы человека рядом с его фамилией, единицы измерения рядом с числом и т. п.). В таком случае между ними следует поставить **«неразрывный» пробел** — он устанавливается с помощью комбинации клавиш Ctrl+Shift+пробел.
- Для набора **разреженного текста** используют разреженный шрифт. Не следует вставлять пробелы между буквами — это приводит к некорректному расположению текста, невозможности проверить правописание, организовать поиск текста. По этим же причинам нельзя расставлять **переносы в словах** с помощью знака «дефис». В Word имеется возможность автоматической расстановки переносов.
- Набранный текст следует **сохранять** как можно чаще. Для этого необходимо использовать возможности панели быстрого доступа, команд кнопки Office, комбинации клавиш — те средства, которые наиболее удобны в данный момент. Для важных текстов следует установить опцию автосохранения.
- Перед окончанием работы с текстом следует воспользоваться автоматической **проверкой орфографии и грамматики**. Даже правильно оформленный, но неграмотный документ создает негативное впечатление.

Пример 1

Выбрать вариант текста, где правильно расставлены пробелы и оформлены знаки препинания.

- 1) Слова принадлежат веку , а мысли—векам .
- 2) Слова принадлежат веку, а мысли— векам.
- 3) Слова принадлежат веку, а мысли — векам.
- 4) Слова принадлежат веку, а мысли — векам.


Решение. В варианте (1) перед запятой и точкой не должно быть пробелов (они должны быть «прижаты» к предыдущим словам). В этом же варианте отсутствуют необходимые пробелы вокруг тире. В варианте (2) есть пробел только после тире, а перед ним отсутствует. Вариант (3) содержит более одного пробела подряд между словами. В варианте (4) знаки расставлены корректно.

Ответ: вариант (4).

Запуск текстового редактора

Запуск редактора MS Word осуществляется стандартными способами Windows — например, с помощью меню кнопки *Пуск/Программы*. Запуск приводит к автоматическому созданию нового текстового документа под рабочим названием *Документ1*.

Открытие любого имеющегося файла документа Word также приводит к запуску программы.

Для работы с файлами в MS Word служит кнопка  *Office*, расположенная в левом верхнем углу окна программы. Для сохранения документа в файле служат команды *Сохранить* и *Сохранить как* в меню этой кнопки. Команда *Сохранить как* запрашивает пользователя, где и под каким названием сохранять документ. Для открытия существующего документа в меню кнопки *Office* служит команда *Открыть*.

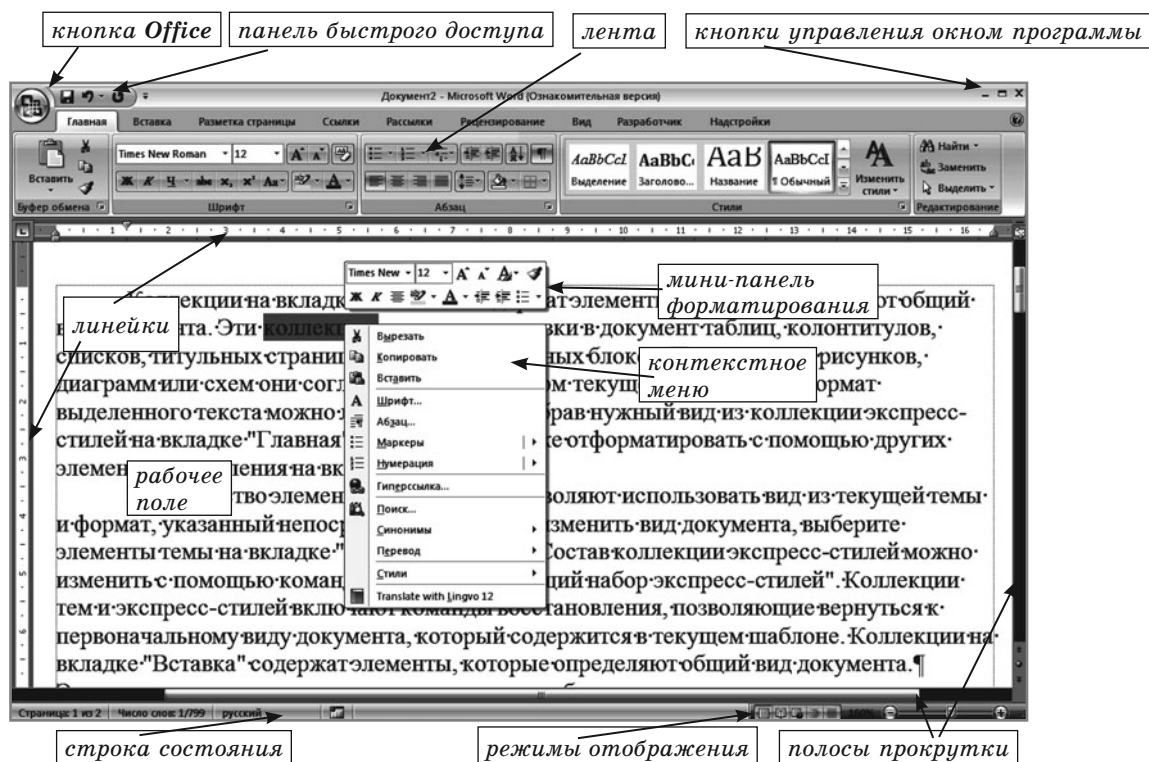



Рис. 2.12. Общий вид окна приложения MS Word

Файлы, с которыми работали недавно, можно открыть быстрее, выбрав их из списка последних открывавшихся документов в правой части меню кнопки *Office*.

Для закрытия окна программы (в т. ч. всех открытых в ней документов) можно использовать кнопку *Выход из Word* в окне меню *Office* либо кнопку  *Закрыть* в правом верхнем углу окна программы. При этом Word проверит, были ли сохранены произведенные в документах изменения. Если какой-либо документ не был сохранен, будет выдан запрос на его сохранение.

Интерфейс MS Word

ЛЕНТА

Вдоль верхнего края окна расположена *лента*, которая содержит несколько вкладок с кнопками. С помощью ленты можно быстро находить необходимые команды. Команды на вкладках упорядочены в логические группы. Названия вкладок отображаются на ленте сверху.

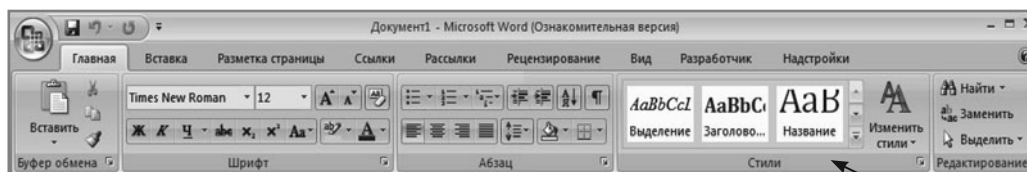


Рис. 2.13. Внешний вид ленты (вкладка Главная)

Каждая из вкладок ленты содержит группы команд определенной тематики. Так, кнопки вкладки *Главная* предназначены для основных действий по вводу и редактированию текста (включая работу с буфером обмена, форматирование, поиск и отбор данных).

Вкладка *Вставка* позволяет добавить в документ различные объекты — страницы, символы, таблицы, рисунки, клипы, диаграммы, надписи, колонтитулы, формулы, гиперссылки и др.

На вкладке *Разметка страницы* собраны команды настройки параметров страниц документа.

ЛИНЕЙКА

Для удобства позиционирования текста и объектов в документе слева и сверху рабочей области окна выводятся соответственно вертикальная и горизонтальная линейки. Они позволяют не только оценить расположение элементов текста, но и произвести установку различных отступов, размеров полей и элементов таблиц, ширины колонок, позиций табуляции и пр.

РЕЖИМЫ ПРОСМОТРА

Документ Word может быть представлен на экране в различном виде. Для выбора режима просмотра в строке состояния (слева от ползунка масштабирования) имеется ряд кнопок. Такие же кнопки присутствуют в группе *Режимы просмотра документа* на вкладке ленты *Вид*.



Чаще всего работа ведется в режиме  *Разметка страницы*. В этом режиме положение текста, таблиц, рисунков и других элементов отображается на экране так, как они будут напечатаны. Режим разметки используют и для создания документа, и для окончательного его оформления. Особенно он удобен для работы со сложными элементами — колонтитулами, полями, многоколоночным текстом, графическими объектами. Именно этот режим установлен в Word по умолчанию.



Рис. 2.14. Кнопки режимов просмотра документа: в строке состояния (а) и на вкладке ленты Вид (б)

Режим чтения  был разработан для удобства чтения документов с экрана компьютера.

Пример 2

Создать документ-приглашение аналогично представленному на рис. 2.15.

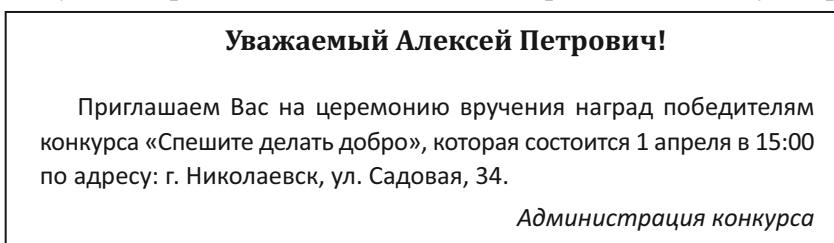


Рис. 2.15. Документ-приглашение

Запустим текстовый редактор. В открывшемся окне появится чистый лист в режиме разметки страницы. Можно начинать набор текста. Он будет появляться в позиции мигающего текстового курсора.

После набора первого предложения (обращения) следует нажать клавишу Enter — курсор переместится в следующий абзац. В следующий раз клавиша Enter должна быть нажата по окончании набора второго абзаца текста. В результате на экране будет получен документ, аналогичный представленному на рисунке 2.16.

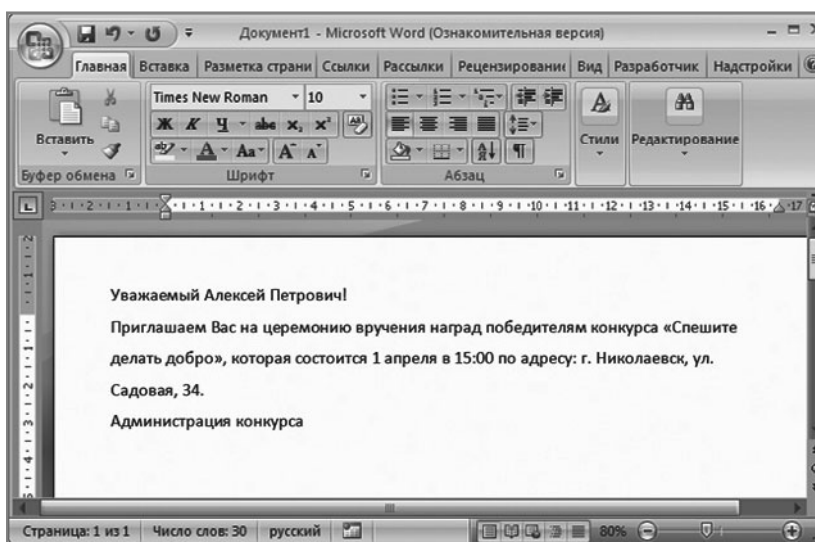



Рис. 2.16. Первоначальный вид приглашения

Для сохранения документа в файле нужно нажать кнопку  *Сохранить* — откроется диалоговое окно, в котором предстоит выбрать папку для сохранения и ввести название файла, допустим, *Приглашение на церемонию*.

Чтобы текст приобрел такой вид, как в условии, предстоит его отформатировать (оформить) и, возможно, внести в него некоторые изменения (см. пример 3).

Работа с фрагментами текста

ПЕРЕМЕЩЕНИЕ ПО ДОКУМЕНТУ

Чтобы поставить курсор в определенное место документа, надо подвести в эту позицию указатель мыши и один раз щелкнуть ее левой кнопкой. Указатель при наведении на текст имеет вид латинской буквы I.

Передвижение по документу с помощью мыши осуществляется с использованием полос прокрутки. Для перемещения надо потянуть мышью бегунок или нажать стрелки на полосах прокрутки. Для построчного просмотра документа следует нажимать одинарные стрелки на полосе прокрутки, для постраничного просмотра — двойные.

Для перемещения по документу в Word активно используются разные комбинации клавиш (табл. 2.2).

Таблица 2.2

Некоторые комбинации клавиш для перемещения

Клавиши	Действие
→ ← ↓ ↑	на один символ вправо/влево, на одну строку вниз/вверх
Home	в начало текущей строки
End	в конец текущей строки
Page Up	на высоту экрана вверх
Page Down	на высоту экрана вниз
Ctrl + →	к следующему слову
Ctrl + ←	к предыдущему слову
Ctrl + Home	в начало документа к самому первому символу
Ctrl + End	к концу документа после самого последнего символа

ВЫДЕЛЕНИЕ ФРАГМЕНТОВ ТЕКСТА

Для выполнения многих действий с текстом — шрифтового форматирования, копирования и перемещения, удаления, преобразования в таблицу и др. — следует предварительно выделить необходимый фрагмент. Выделение можно осуществить с помощью мыши, клавиш или их комбинаций. Выделенный фрагмент будет слегка затенен серо-голубым фоном.

Для выделения используют следующие способы.

- *Выделить любой фрагмент*: провести по нему мышью с нажатой левой кнопкой.
- *Выделить слово*: дважды щелкнуть по нему мышью.
- *Выделить строку*: щелкнуть мышью в поле слева от строки — в так называемой *зоне выделения* (курсор при этом должен иметь вид стрелки, направленной вправо вверх). Если выделяемых строк несколько, протянуть мышшь вдоль них по зоне выделения.

- *Выделить абзац*: щелкнуть мышью в поле слева от абзаца. Другой способ: трижды щелкнуть по самому абзацу.
- *Выделить весь документ*: нажать комбинацию клавиш Ctrl+A.

Для выделения также используют клавишу Shift вместе с соответствующими комбинациями клавиш перемещения. Например, комбинация Ctrl+Home перемещает курсор к началу строки, а комбинация Shift+Ctrl+Home выделяет фрагмент от текущей позиции до начала строки.


Чтобы *снять выделение* фрагмента, надо один раз щелкнуть мышью вне выделенного блока — например, в правом поле документа.


УДАЛЕНИЕ ТЕКСТА

Для *удаления символа* справа от текстового курсора надо нажать клавишу Delete, для удаления символа слева от курсора — клавишу Backspace.

Для *удаления фрагмента* текста надо сначала выделить его, затем нажать клавишу Delete.

ОТМЕНА И ВОЗВРАТ ДЕЙСТВИЙ

Для отмены последнего выполненного действия служит кнопка  *Отменить* на панели быстрого доступа. Выпадающий список у этой кнопки позволяет отменить сразу несколько последних операций (для этого нужно выбрать в списке самое раннее отменяемое действие).

Для возврата отмененных действий предназначена соседняя кнопка панели быстрого доступа  *Вернуть*.


Эти операции могут быть выполнены с помощью клавиатуры. Для отмены действия служат комбинации клавиш Ctrl+Z и Alt+Backspace, для возврата — Ctrl+Y.

Пример 3

Усовершенствуем документ, созданный в предыдущем примере. Предусмотрим, чтобы не разрывались между собой некоторые его фрагменты (чтобы при любых изменениях весь фрагмент целиком размещался на одной, а не на разных строках).

С этой целью между числом «1» и словом «апреля», словом «в» и числом «15», словами «г.» и «Николаевск», «ул.» и «Садовая» надо вставить специальный символ (*неразрывный пробел*), предотвращающий разрывы строк в этих позициях.

Для этого установим курсор с помощью мыши или клавиатуры в позицию перед буквой «а» в слове «апреля» с помощью клавиши Backspace удалим пробел слева от курсора и вставим неразрывный пробел, нажав комбинацию клавиш Ctrl+Shift+Enter. Аналогично нужно поступить и во всех остальных случаях.

Сохраним документ с помощью кнопки  на панели быстрого доступа (в этот раз никакого диалогового окна на экран выдано не будет — обновленный документ будет сохранен под тем же названием в той же папке, что и раньше).

КОПИРОВАНИЕ И ПЕРЕМЕЩЕНИЕ ДАННЫХ. БУФЕР ОБМЕНА

Перемещать и копировать можно только выделенные фрагменты документа.

В пределах видимой части документа проще всего использовать перетаскивание выделенного фрагмента мышью. Для этого фрагмент нужно захватить

указателем мыши (который должен иметь вид стрелки, повернутой влево вверх) и перетащить в требуемую позицию (она помечается вертикальной пунктирной линией).

Для копирования при перетаскивании фрагмента следует нажать и удерживать клавишу Ctrl.

Если требуется перетащить фрагмент за пределы видимой части документа, надо подтянуть его к верхнему или нижнему краю листа и, не отпуская кнопку мыши, задержать там — документ начнет автоматически прокручиваться. В нужный момент следует вернуть указатель в поле документа и отпустить фрагмент в требуемой позиции.

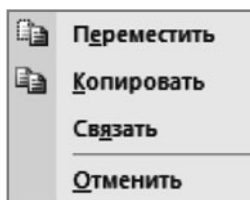

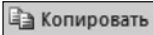



Рис. 2.17. Контекстное меню

Для перетягивания фрагмента удобно использовать правую кнопку мыши — после ее отпускания на экран выводится контекстное меню, где надо выбрать необходимое действие (например, *Переместить* или *Копировать*). Это меню позволяет и отказаться от операции (команда *Отменить*).

Другой подход — использовать для копирования и перемещения текста *буфер обмена* — специальную область памяти компьютера для хранения промежуточной информации. Буфер обмена является общим для всех приложений MS Office, включая MS Word.

Для копирования и перемещения фрагмента сначала следует отправить его в буфер обмена, а затем извлечь в нужную позицию. При этом отправить фрагмент можно двумя способами: удалив из исходного текста («вырезать» в буфер обмена) либо оставив в документе («скопировать» в буфер).

Вырезание в буфер обмена можно выполнить с помощью кнопки  вкладки ленты *Главная* (группа *Буфер обмена*), копирование — соседней кнопкой . Кроме того, можно выбрать эти команды из контекстного меню выделенного фрагмента (вызвав его правой кнопкой мыши). Удобно использовать комбинацию клавиш: для вырезания — Ctrl+X или Shift+Del, для копирования — Ctrl+C или Ctrl+Ins.

Информация из буфера обмена может быть вставлена кнопкой  вкладки ленты *Главная* (группа *Буфер обмена*) либо комбинациями клавиш Ctrl+V или Shift+Ins.

После вставки фрагмента буфер обмена MS Office не очищается. Следовательно, можно неоднократно вставлять тот же самый фрагмент (в том числе и в документы других приложений).

ПОИСК И ЗАМЕНА ТЕКСТА

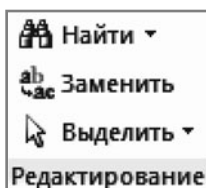


Рис. 2.18. Команды группы Редактирование

Часто в документе требуется найти или заменить какие-либо фрагменты текста. Для этого служат кнопки группы *Редактирование* на вкладке ленты *Главная*.

Чтобы найти текст, надо нажать кнопку *Найти*, в открывшемся окне ввести в поле *Найти* искомый текст и нажать кнопку *Найти далее*. Для поиска следующих вхождений этого же текста надо много-

кратно нажимать кнопку *Найти далее*. Найденные фрагменты текста будут по очереди выделяться в документе фоновой заливкой. Кнопка *Найти в* приведет к одновременному выделению всех вхождений такого фрагмента в тексте.

Если поиск требуется только в какой-то части текста, ее необходимо предварительно выделить.

Word позволяет организовать автоматическую замену найденного фрагмента текста другим. Для этого в том же окне поиска имеется вкладка *Заменить*. С помощью кнопки *Заменить* вкладки ленты *Главная* можно вызвать это окно, раскрытое сразу на вкладке замены. В появившемся диалоговом окне надо заполнить строку *Найти* (т. е. указать, что надо заменить) и строку *Заменить на* (т. е. указать, чем надо заменять).

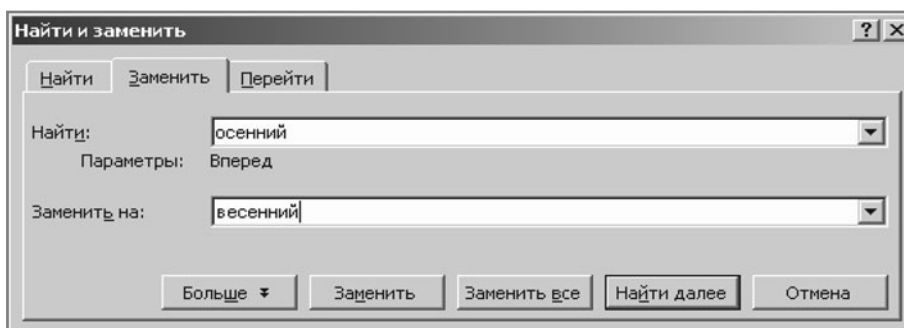


Рис. 2.19. Окно замены текста

Форматирование текста

ШРИФТОВОЕ ОФОРМЛЕНИЕ ТЕКСТА

Форматирование — это изменение внешнего вида документа или отдельных его частей. Можно форматировать символы, абзацы, таблицы, страницы и др.

Внешний вид символов текста определяет **шрифт**. Шрифтовое оформление текста можно задать как до его набора, так и по окончании. Если фрагмент уже набран, перед шрифтовым оформлением его необходимо предварительно выделить.

Наиболее часто используемые опции шрифтового (и не только) оформления собраны на *мини-панели форматирования*. Она всплывает автоматически при выделении текста, поначалу полупрозрачная, но при наведении на нее курсора становится яркой.



Рис. 2.20. Инструменты мини-панели форматирования

Шрифтовое оформление задается с помощью кнопок группы *Шрифт* на вкладке ленты *Главная*.

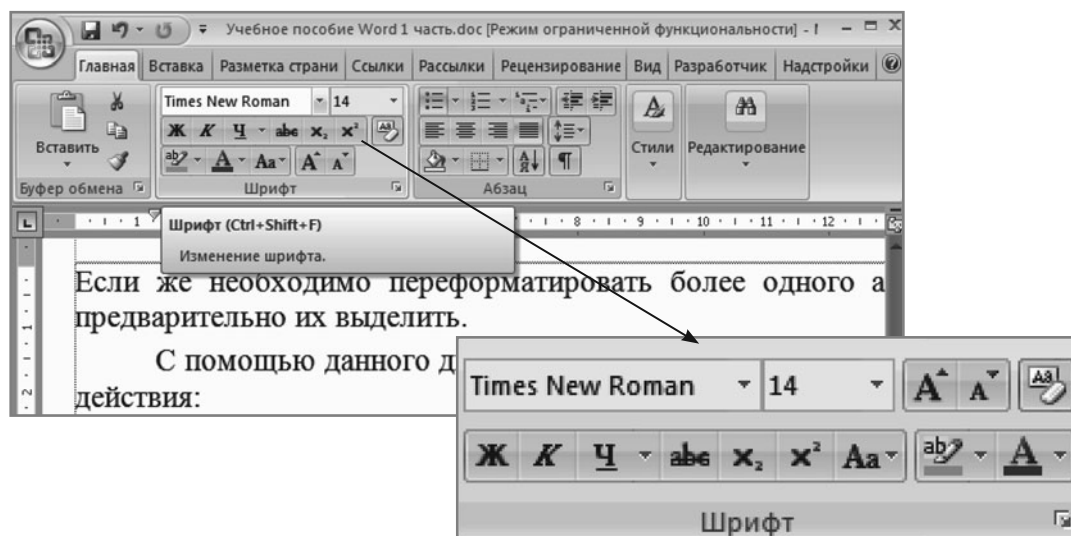







Рис. 2.21. Кнопки шрифтового оформления на вкладке ленты Главная (в узком и широком окне)


Полностью все опции шрифтового форматирования собраны в диалоговом окне *Шрифт*, которое вызывается с помощью значка  *Шрифт* этой же группы. Это же окно можно вызвать командой *Шрифт* контекстного меню выделенного фрагмента.


Шрифтовое оформление включает в себя нижеперечисленные составляющие.

1. **Гарнитуру** (*тип*, или *семейство*) шрифта, например Times New Roman, Arial, System или Calibri. Гарнитура определяет внешний вид шрифта для всех его размеров и вариантов начертания. Например, все данное предложение набрано **только шрифтами ГАРНИТУРЫ Arial**.

2. Способ **начертания** шрифта определяет разновидности написания символов одного шрифта. Обычно к ним относят **курсив** , **полужирный**  и **подчеркнутый**  варианты шрифта. (На ленте соответствующие кнопки действуют как переключатели — при повторном их выборе происходит отмена данного начертания.) Свойства шрифта «полужирный» или «светлый» называют **насыщенностью**.

3. **Размер (кегель)** шрифта , причем нужный размер можно как выбирать из выпадающего списка, так и вводить с клавиатуры. Кегль измеряется в пунктах («пт») — типографской единице измерения, равной приблизительно 0,35 мм. Например, данное предложение набрано кеглем от 8 пт до 18 пт.

4. Способ **подчеркивания** текста . Может быть **одинарным**, **двойным**, **точечным**, **пунктирным** и др.

5. **Цвет** символов текста . По умолчанию устанавливается цвет *авто* — темный на светлом фоне.

6. Иные специальные **эффекты** шрифтового оформления, например зачеркивание, теневое или контурное оформление, установка всех прописных букв, надстрочных или подстрочных индексов и др.

Во второй вкладке диалогового окна шрифта — *Интервал* — можно задать разреживание и уплотнение текста с различными коэффициентами. Так мож-

но организовать, например, разреженный шрифт для заголовков. К примеру, данное предложение набрано шрифтом, разреженным на 1,5 пт. А это предложение уплотнено на 0,7 пт.


В диалоговом окне *Шрифт* образец будущего представления фрагмента текста отражается в нижней части окна.

Кнопки форматирования на ленте имеют *функцию предпросмотра*: если навести на них курсор — в тексте будет показан результат такого форматирования выделенного фрагмента.

Для очистки шрифтового формата служит кнопка  *Очистить формат*.

ФОРМАТИРОВАНИЕ АБЗАЦЕВ

Абзацем в Word называется фрагмент текста между нажатиями клавиши Enter. При ее нажатии в текст вставляется специальный (непечатаемый) символ ¶, который называется *знаком конца абзаца*.

Непечатаемые символы обычно не отображаются на экране. Чтобы увидеть этот и другие непечатаемые символы, надо включить режим их отображения с помощью кнопки  *Отобразить все знаки*. В таком режиме видно, как устроен текст, и зачастую именно благодаря этому режиму можно понять, почему в расположении текста произошли какие-либо изменения. Другие непечатаемые символы: обычный пробел — · (точка), неразрывный пробел — ○, конец страницы — длинная точечная линия.

Абзац может содержать любое количество строк (в том числе и не содержать текста вообще).

Если необходимо форматировать несколько абзацев, следует предварительно их выделить.

Большая часть параметров абзацного форматирования сосредоточена в группе кнопок *Абзац* вкладки ленты *Главная*. Среди них — тип выравнивания текста абзаца, межстрочное расстояние, отступы и выступы и др. Кроме того, группа *Абзац* имеется также на вкладке *Разметка страницы*.

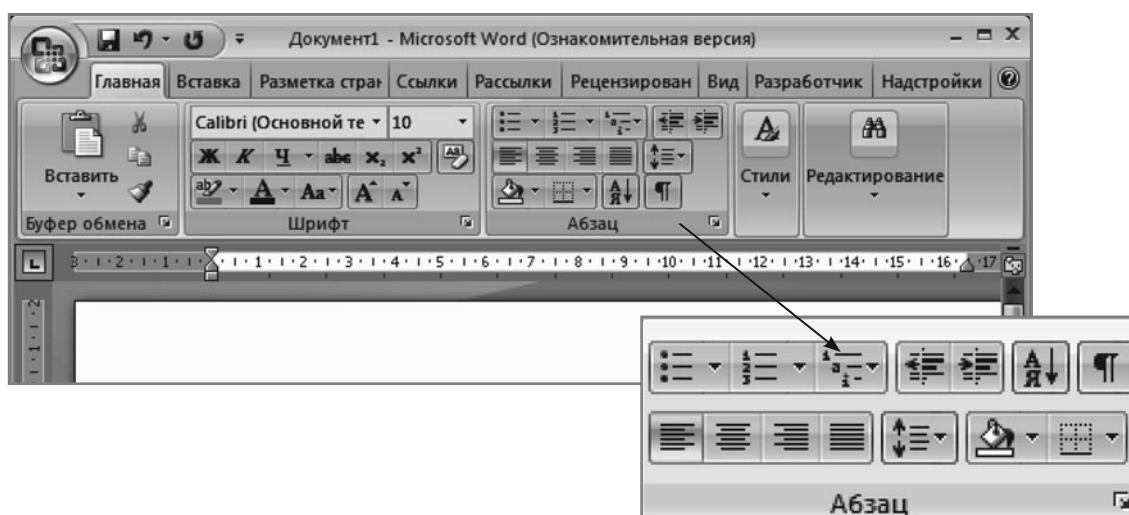







Рис. 2.22. Группа кнопок *Абзац* на вкладке ленты *Главная* (в узком и широком окне)


Весь набор опций абзацного форматирования представлен в диалоговом окне *Абзац*. Его можно вызвать с помощью значка  в этих же группах кнопок ленты или из контекстного меню текста (командой *Абзац*).

Существует четыре варианта выравнивания текста в абзаце: по левому краю (ему соответствует кнопка ленты ) , по правому краю () , по центру () и по ширине () .

При выравнивании по левому краю текст абзаца будет прилегать к левому краю полосы документа, а правый будет неровным, рваным (как в данном абзаце).

Аналогично — для абзацев текста, выровненных по правому краю. Их строки будут иметь выровненный правый край и неровный левый.

Выравнивание абзаца по центру чаще всего применяют для заголовков, названий, подписей.

Для центрирования абзаца на мини-панели инструментов имеется кнопка .

Выравнивание по ширине означает, что оба края абзаца — левый и правый — будут равными за счет вставки дополнительных промежутков (не пробелов!) между словами текста.

Установка отступа абзаца изменяет положение текста относительно левого и/или правого полей страницы. Имеется возможность установить отступ всех строк, только первой строки и всех остальных строк. Для этого можно воспользоваться кнопками *Слева* и *Справа* на вкладке ленты *Разметка страницы*.

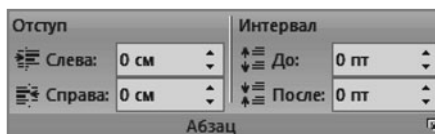


Рис. 2.23. Группа кнопок *Абзац* на вкладке ленты *Разметка страницы*

Для установки отступов удобно использовать горизонтальную линейку. Необходимо всего лишь перетащить нужный бегунок линейки в требуемую позицию.

На рисунке 2.24 приведен пример установки отступа первой строки 1,27 см (там находится левый верхний маркер линейки), что соответствует понятию «красной строки» абзаца. Левый и правый края абзаца совпадают с левым и правым полями документа (там находятся нижние маркеры линейки — левый и правый).

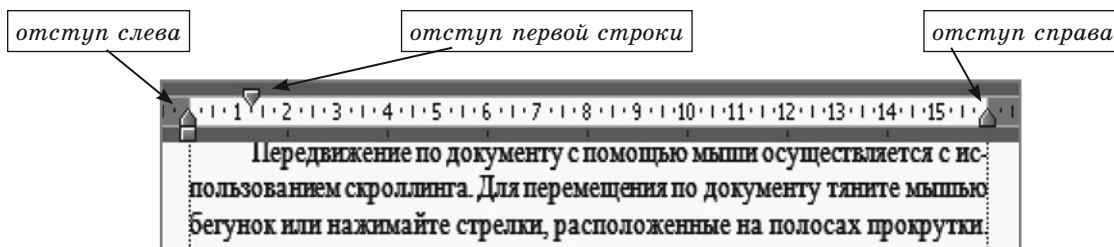





Рис. 2.24. Установка абзацных отступов с помощью линейки



На ленте в группе *Абзац* вкладки *Главная* имеются кнопки  *Увеличить отступ* и  *Уменьшить отступ*, позволяющие изменить левый отступ. Такие же кнопки имеются на мини-панели инструментов.

Расстояние между строками абзаца по вертикали называется **межстрочным интервалом** и устанавливается в диалоговом окне *Абзац* либо с помощью кнопки  *Междустрочный интервал* вкладки *Главная*. Чаще всего используются одинарный (1,0) и полуторный (1,5) интервалы. Коэффициенты означают, что расстояние между строками равно высоте одной или полутора строк текста.



Можно устанавливать интервалы между самими абзацами. Для этого используются кнопки *До* и *После* на вкладке ленты *Разметка страницы*.




Пример 4

Продолжим работу над документом, созданным в примерах 2 и 3. Отформатируем приглашение в соответствии с первоначальным образцом.

Оформим первое предложение текста. Выделим его щелчком мыши слева от него (в зоне выделения) и выберем во всплывшей мини-панели форматирования из выпадающего списка шрифт *Cambria*. Сразу же, не снимая выделения, на этой же панели установим полужирное начертание с помощью кнопки . Тут же нажмем кнопку , чтобы добиться центрирования текста документа.

Для установки интервалов перейдем на вкладку ленты *Разметка страницы* и в группе кнопок *Абзац* укажем интервал перед абзацем 12 пт, а после него — 18 пт.

Выделим второй абзац текста (например, двойным щелчком мыши в зоне выделения слева от него). Установим для этого текста шрифт *Calibri*. В группе кнопок ленты *Абзац* нажмем кнопку , чтобы выровнять текст по ширине страницы. Для выбора полуторного межстрочного интервала воспользуемся там же кнопкой .

Наконец, выделив последнее предложение, выберем курсивное начертание (с помощью кнопки  на мини-панели или ленте). Зададим выравнивание по правому краю (кнопкой  на вкладке *Главная*). Укажем интервал перед абзацем 18 пт. Сохраним файл с помощью кнопки  на панели быстрого доступа.

Пример 5

Указать отличия в шрифтовом и абзацном оформлении двух фрагментов документа.

1) *Периферийные устройства* — устройства для ввода или вывода информации. К ним относят принтеры, клавиатуры, мыши, сканеры и т. д. Подсоединение их к компьютеру производится через специальные разъемы — порты ввода/вывода.

2) *Периферийные устройства* — устройства для ввода или вывода информации. К ним относят принтеры, клавиатуры, мыши, сканеры и т. д. Подсоединение их к компьютеру производится через специальные разъемы — порты ввода/вывода.

- 1) гарнитура шрифта;
- 2) начертание шрифта (прямое, курсивное);
- 3) насыщенность шрифта (светлый, полужирный);
- 4) выравнивание текста.


Решение. Первый фрагмент текста оформлен рубленным шрифтом (без засечек), имеет выравнивание по ширине, прямое начертание. Во втором фрагменте


использованы шрифт с засечками, левостороннее выравнивание, курсивное начертание. Насыщенность текста в обоих текстах светлая. Таким образом отличия в оформлении текстов — варианты 1), 2), 4).

Ответ: 1), 2), 4).


ОБРАМЛЕНИЕ И ЗАЛИВКА

Word предоставляет возможность установить **обрамление** текста (границы). Обрамлять можно один или несколько абзацев, с одной или с нескольких сторон, границы могут быть разной толщины, цвета и находиться на разном расстоянии.

Самый простой способ установки границ — с помощью кнопки  группы *Абзац* вкладки ленты *Главная*. Ее выпадающий список содержит перечень возможных вариантов оформления. Тот вариант, который был выбран последним, будет отображаться на самой кнопке и в ее подписи. Чтобы повторно выбрать этот же вариант заливки, можно нажимать только кнопку, без выбора строки списка.

Чтобы удалить границы, надо выбрать в выпадающем списке кнопки ленты  команду *Нет границы*.

Можно изменить цвет фона для выбранного абзаца (абзацев). При фоновой заливке изменяется фон всего абзаца — от левого до правого поля. Обычно такая заливка используется для визуального выделения фрагментов документа.

Для установки цвета фона можно использовать кнопку  *Заливка* в группе кнопок ленты *Абзац* вкладки *Главная*.

Организация списков

РАЗНОВИДНОСТИ СПИСКОВ

При работе с текстом часто возникает необходимость пометить абзацы — иногда специальными значками-маркерами, иногда с помощью цифр или комбинированным способом. Для решения этой задачи Word предусматривает три вида списков — *маркированные*, *нумерованные* и *многоуровневые*. Примеры разных видов списков приведены на рис. 2.25.


Устройства ввода информации: <ul style="list-style-type: none"> ♦ клавиатура; ♦ сканеры; ♦ дигитайзеры (графические планшеты); ♦ манипуляторы (джойстик, мышь, трекбол, световое перо); ♦ сенсорные экраны. 	
Виды антивирусных программ: <ol style="list-style-type: none"> 1. детекторы; 2. доктеры (или фаги); 3. ревизоры; 4. фильтры; 5. вакцины (или иммунизаторы). 	Экономический рост бывает: <ol style="list-style-type: none"> 1. Экстенсивный <ol style="list-style-type: none"> 1.1. простой 1.2. структурный <ul style="list-style-type: none"> * качественный * количественный 2. Интенсивный <ol style="list-style-type: none"> 2.1. частичный 2.2. полный 2.3. глобальный


Рис. 2.25. Примеры организации списков: маркированный, нумерованный, многоуровневый

Оформление списков в Word производится с помощью первых трех кнопок группы *Абзац* на вкладке ленты *Главная* — каждая из них отвечает за один из видов списков.


Можно сначала набрать текст, затем оформить его в виде списка; можно поступить наоборот. Часто первый вариант оказывается эффективнее.

МАРКИРОВАННЫЕ СПИСКИ


Для создания **маркированного списка** надо выделить абзацы текста (если они уже набраны) и нажать кнопку  *Маркеры*. Начало каждого абзаца будет помечено маркером, вид которого был выбран ранее. Если требуется иной маркер, следует выбрать подходящий значок в выпадающем списке этой кнопки. Если в списке нет подходящего символа, можно вызвать команду *Определить новый маркер*.


Кнопка  *Маркеры* имеется также на мини-панели форматирования и в контекстном меню выделенного текста.

После маркировки абзацев нажатие клавиши *Enter* в конце любого из них оформит новый начатый абзац тем же самым маркером.

Для снятия маркировки выделенных абзацев следует повторно нажать кнопку  *Маркеры* (она работает как переключатель) или выбрать в ее выпадающем списке вариант *нет*.

НУМЕРОВАННЫЕ СПИСКИ

Нумерованный список позволяет структурировать текст. Для установки нумерации следует выделить абзацы текста (если они уже набраны) и воспользоваться кнопкой  *Нумерация*. Абзацы будут пронумерованы тем способом, который был использован в предыдущий раз. Для выбора другого способа у кнопки имеется выпадающий список вариантов нумерации.

Команда  *Нумерация* имеется также в контекстном меню выделенного текста. Если необходимого варианта нумерации в списке нет, надо вызвать команду *Определить новый формат номера* и задать свой.


Чтобы нумерацию в списке начать со значения, отличного от 1, в выпадающем списке кнопки  *Нумерация* следует воспользоваться командой *Задать начальное значение*. В диалоговом окне можно выбрать между вариантами *Продолжить нумерацию* или



Рис. 2.26. Выбор маркированного списка

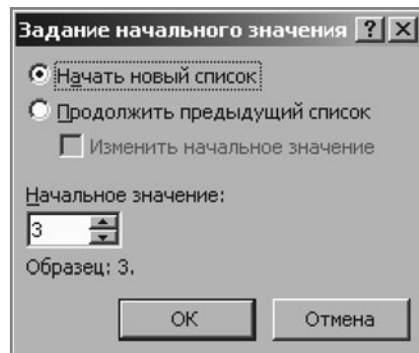


Рис. 2.27. Определение начала/продолжения нумерации списка

Начать заново; а также указать, с какого значения следует начинать нумеровать абзацы текста.

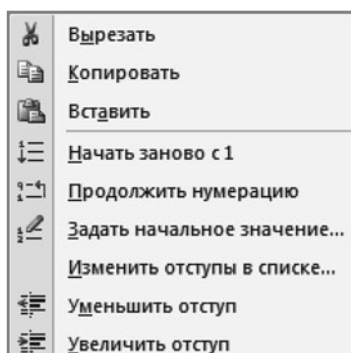




Рис. 2.28. Команды контекстного меню нумерованного абзаца

Команды *Начать заново с 1*, *Продолжить нумерацию*, *Задать начальное значение* присутствуют в контекстном меню нумерованных абзацев.

Если абзацы оформлены нумерованным списком, то нажатие клавиши Enter в конце любого из них оформит новый абзац тем же образом. При этом будет сохранена корректность нумерации — если новый абзац добавляется в середине имеющегося списка, нумерация всех последующих будет сдвинута на 1. Если нумерация более не требуется, надо удалить вставленный номер клавишей Backspace.


Для снятия нумерации выделенных абзацев следует повторно нажать кнопку  *Нумерация* (она работает как переключатель) или выбрать в ее выпадающем списке вариант *нет*.

МНОГОУРОВНЕВЫЕ СПИСКИ

Оба рассмотренных выше вида списков можно комбинировать — получая разновидности *комбинированного списка*. Для этого предназначена кнопка  *Многоуровневый список* вкладки ленты *Главная*.

Можно выбрать один из предусмотренных вариантов в списке этой кнопки или разработать свой вид списка с помощью команды *Определить новый многоуровневый список*.

В результате применения многоуровневого списка к уже набранному тексту все абзацы будут отформатированы как 1-й уровень. Чтобы перевести абзац на 2-й уровень, надо после его номера нажать клавишу Tab. Для обратного перехода — на уровень выше — служит комбинация клавиш Shift+Tab.

Для удаления многоуровневого списка следует нажать кнопку ленты  *Многоуровневый список* и выбрать вариант *нет*.

Пример 6

Создадим документ, содержащий маркированный список.

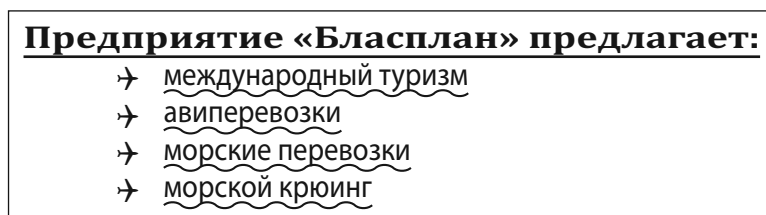



Рис. 2.29. Пример маркированного списка

Решение. Начнем работу с набора текста — каждая строка в этом примере является отдельным абзацем, т. е. ее набор завершается нажатием клавиши Enter.

Выделив последние 4 строки мышью, выберем во всплывшей мини-панели выпадающий список кнопки  *Маркеры*, а из него — команду *Определить новый маркер*. В открывшемся окне нажмем кнопку *Символ*, найдем в таблице шрифта Wingdings требуемый значок и нажмем *ОК*.

Осталось установить шрифтовое оформление текста и сохранить его.

Параметры страницы

Для настройки внешнего вида страниц служат кнопки *Поля*, *Ориентация* и *Размер* на вкладке ленты *Разметка страницы* (в группе *Параметры страницы*).

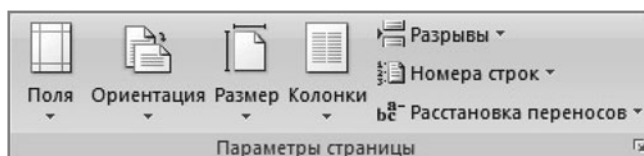



Рис. 2.30. Группа кнопок *Параметры страницы* вкладки *Разметка страницы*

В выпадающих списках этих кнопок перечислены наиболее часто используемые варианты настроек. Так, *размер бумаги* чаще всего используют А4 (210×297 мм) или его половинку — А5 (148×210 мм). *Ориентация бумаги* может быть *книжной* (длинная сторона листа расположена вертикально) или *альбомной* (длинная сторона листа расположена горизонтально).

Номера страниц можно вставить с помощью кнопки  *Номер страницы* в группе *Колонтитулы* на вкладке *Вставка*.

Пользователю предлагается несколько вариантов нумерации — вверху и внизу страницы, на боковых полях документа. Там же находится и команда удаления всей нумерации страниц.

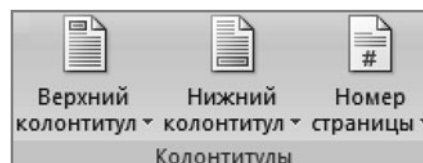


Рис. 2.31. Группа кнопок *Колонтитулы* вкладки *Вставка*

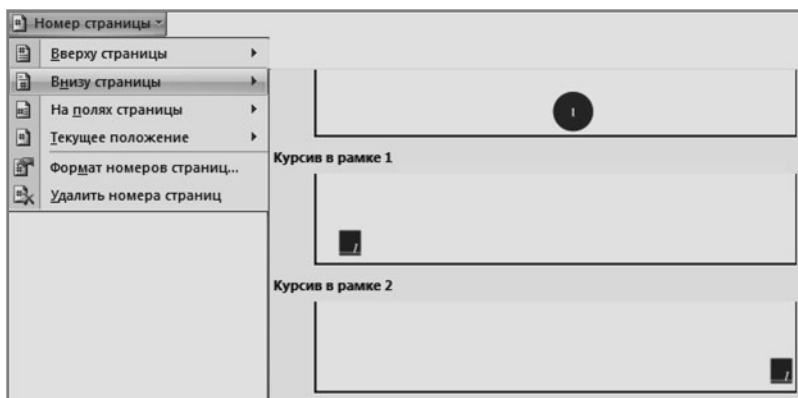


Рис. 2.32. Варианты вставки нумерации страниц

Стили и темы в оформлении документов Word

При работе с документом часто приходится оформлять одинаково однородные фрагменты — заголовки, названия рисунков и таблиц, основной текст документа и т. п. Облегчить единообразное оформление документов помогают *стили*. Они позволяют применить совокупность атрибутов форматирования за одно действие. С помощью стилей автоматически создается оглавление документа.

Стиль — набор параметров форматирования, который применяется к символам текста, абзацам, таблицам и спискам.

Стиль символа задает форматирование фрагмента текста внутри абзаца: гарнитуру шрифта, его размер, цвет, начертание и др.

Стиль абзаца определяет внешний вид абзаца: выравнивание текста, отступы, межстрочный интервал, оформление.

Стиль таблицы задает вид границ, заливку, выравнивание текста и шрифты в таблице.

Стиль списка применяет одинаковое выравнивание, шрифты, знаки нумерации или маркеры к спискам.

Документ всегда оформлен с использованием стилей, даже если они специально не применялись. По умолчанию к тексту документа применяется стиль *Обычный*. Для применения иного стиля можно использовать кнопки из группы *Стили* на вкладке *Главная*.

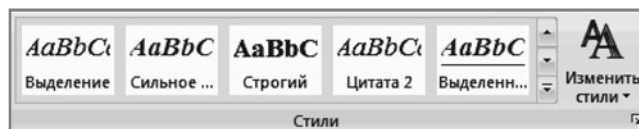



Рис. 2.33. Группа кнопок ленты Стили

Для создания нового стиля можно воспользоваться кнопкой  *Создать стиль* в области задач *Стили*. В открывшемся окне нужно задать имя стиля и весь набор параметров форматирования.

Автоматическое оглавление документа

Word помогает автоматизировать многие операции по оформлению текста. Так, кнопки вкладки *Ссылки* позволяют автоматически сформировать оглавление документа и предметный указатель, вставить в текст сноски и перекрестные ссылки, создать списки литературы и иллюстраций.

Группа кнопок *Оглавление* предназначена для сборки оглавления документа и его обновления в случае внесения изменений в текст. Для этого сначала надо оформить встроенными стилями *Заголовок 1* — *Заголовок 9* те заголовки, которые требуется включить в оглавление. *Заголовок 1* применяют к названиям самого верхнего уровня (главам), *Заголовок 2* — к вложенным подзаголовкам (параграфам, разделам). Следует использовать столько уровней заголовков, сколько должно содержать будущее оглавление.

Затем надо установить курсор в позицию будущего оглавления (обычно в начало или конец документа) и выбрать на вкладке *Ссылки* в списке кнопки *Оглавление* один из видов автоматического (автособираемого) оглавления. Если

ни один из них не подходит — внешний вид оглавления можно изменить, выбрав у кнопки *Оглавление* команду *Оглавление*.

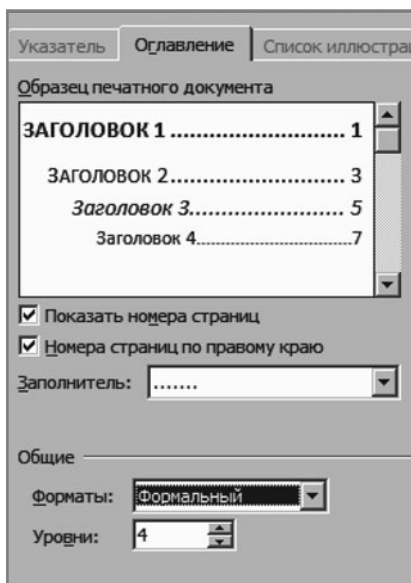


Рис. 2.34. Настройка внешнего вида оглавления

Word найдет все заголовки, оформленные указанными стилями, отсортирует их по уровню заголовка, добавит соответствующие номера страниц и отобразит оглавление в документе.

Оглавление будет представлять из себя единое поле. В заголовке этого поля можно выбрать кнопку замены вида оглавления и команды его удаления (*Удалить оглавление*) или обновления (*Обновить таблицу*).

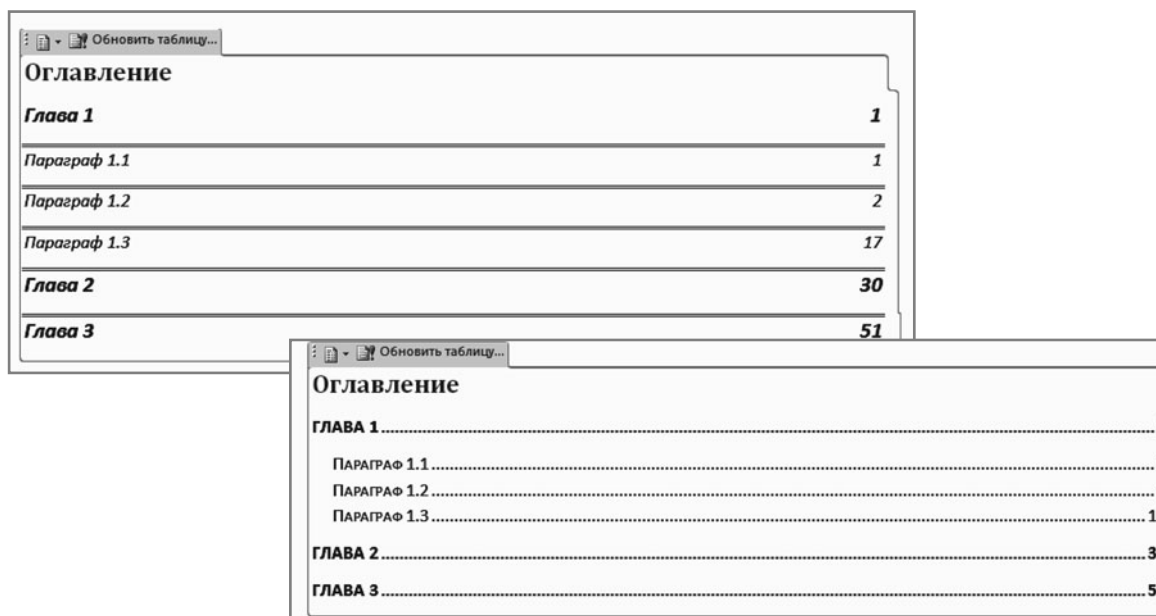


Рис. 2.35. Оглавление (два варианта внешнего вида)

Ссылки

СНОСКИ

Сноски (*обычные* и *концевые*) используют для различных уточняющих сведений. **Обычные сноски** печатаются внизу каждой страницы документа. **Концевые сноски** размещаются в конце документа. Один документ может содержать как обычные, так концевые сноски.

Сноска состоит из двух связанных частей: знака сноски и текста сноски. *Знак сноски* вставляется в позицию курсора. *Текст сноски* (в конце страницы или документа) повторяет этот знак и содержит разъяснение, уточнение, ссылку. Можно задать автоматическую нумерацию сносок (при перемещении, копировании или удалении знака сноски Word перенумерует оставшиеся сноски).

Для организации сносок служат кнопки группы *Сноски* на вкладке *Ссылки* (при этом *Вставить сноску* добавляет обычную сноску). После ввода текста в области сносок следует вернуть курсор в основной документ, щелкнув в нем мышью.

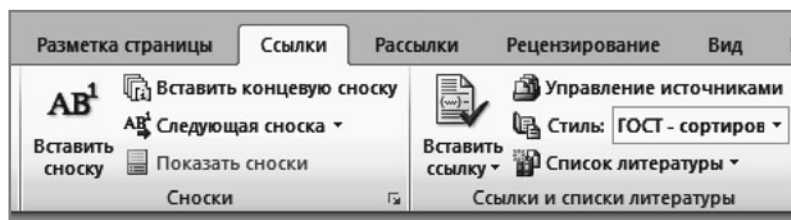


Рис. 2.36. Кнопки работы со сносками и списком источников

Для удаления сноски следует выделить ее знак в тексте и нажать клавишу Del.

СПИСОК ИСТОЧНИКОВ

Очень удобным средством Word является автоматизированное формирование **списка источников литературы** к научной работе, реферату, статье. Если в документе потребуется сделать повторную ссылку на источник информации или произойдут изменения в списке источников — программа отследит нумерацию, автоматически внесет все изменения в список и т. д. Для формирования списка источников литературы предназначены кнопки вкладки *Ссылки* (группа *Ссылки и списки литературы*).

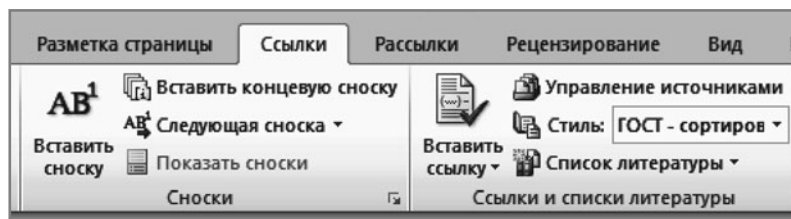


Рис. 2.37. Кнопки вкладки Ссылки

Для добавления первой ссылки на источник служит кнопка *Вставить ссылку/Добавить новый источник*. В окне следует заполнить сведения о нем (краткие или подробные).

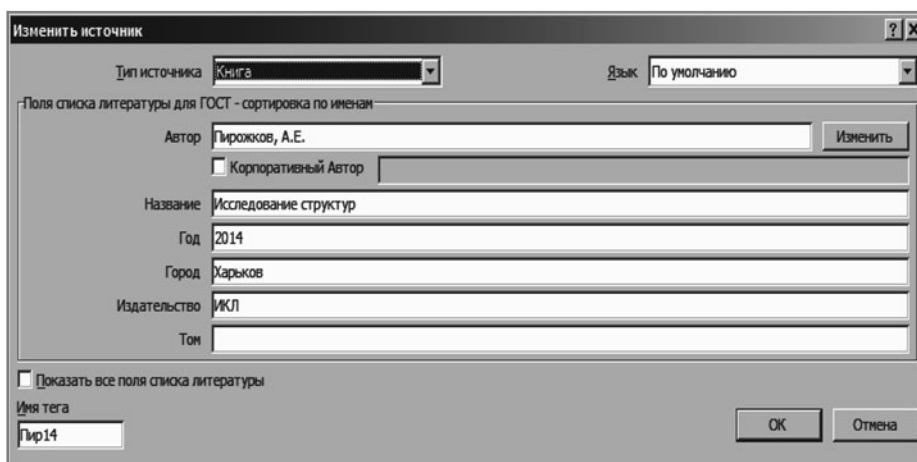


Рис. 2.38. Окно краткого описания источника

Ссылка будет вставлена в текст в таком виде, который определен с помощью кнопки *Стиль* (рис. 2.39). Если был выбран стиль *ISO цифровая ссылка* — будет вставлен порядковый номер (рис. 2.40а), если один из стилей *ГОСТ* — фамилия автора и год издания (рис. 2.40б). И в том и в другом случае ссылке можно будет отредактировать командами ее контекстного меню.

Для повторной ссылки на тот же источник достаточно будет выбрать его из списка кнопки *Вставить ссылку* (рис. 2.41).

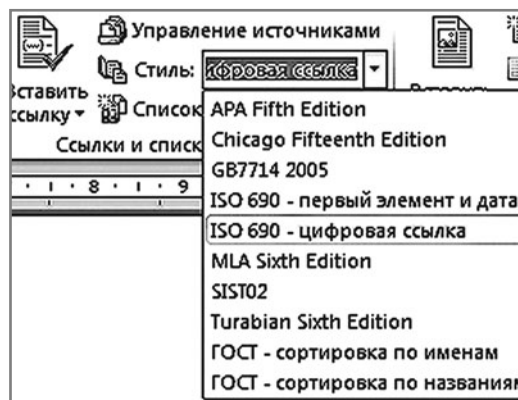
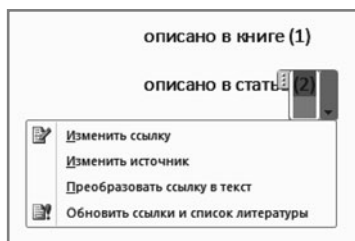
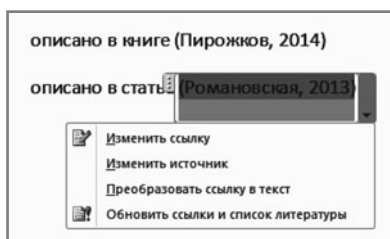


Рис. 2.39. Выбор стиля ссылки на источник



а



б

Рис. 2.40. Два варианта ссылок на источник литературы: а — нумерация, б — название

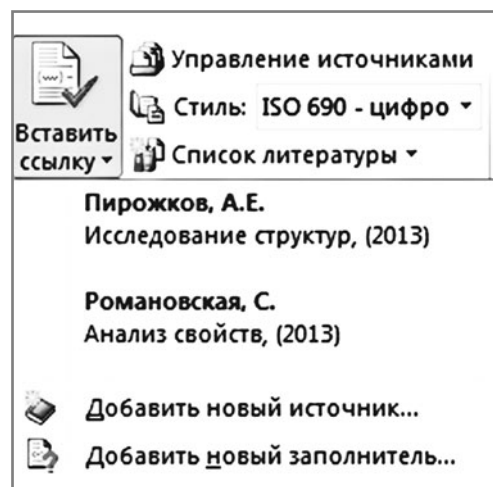


Рис. 2.41. Повторная вставка ссылки на источник

В конце документа можно собрать список всех источников, использованных в документе, с помощью кнопки *Список литературы* (рис. 2.42).

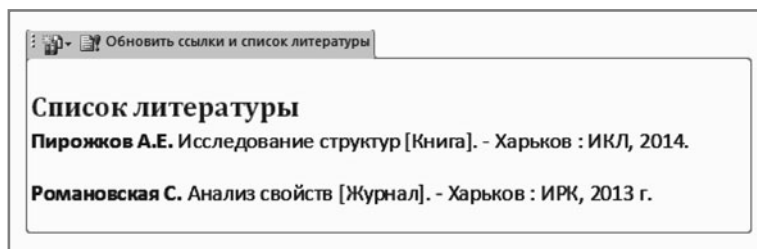


Рис. 2.42. Собранный автоматически список литературы

Организация табличных данных

Таблицы используются для упорядочения данных, создания макетов страниц со столбцами текста и рисунков; их можно применять для создания бланков документов.

Для работы с таблицами Word имеет на ленте две контекстно-зависимые вкладки — *Конструктор* и *Макет*, объединенные в группу *Работа с таблицами*. Они становятся доступными, как только курсор попадает в таблицу. Вкладка *Конструктор* позволяет быстро форматировать таблицу, а вкладка *Макет* используется для редактирования таблиц.

СОЗДАНИЕ ТАБЛИЦ

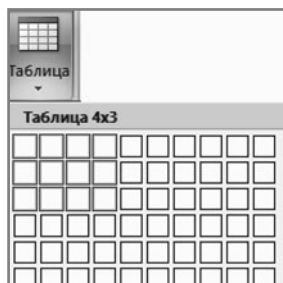


Рис. 2.43. Создание простой таблицы

Для того чтобы быстро создать простую таблицу, достаточно нажать кнопку *Таблица* на вкладке *Вставка* и обвести в появившейся сетке ячеек нужное количество строк и столбцов. На рис. 2.43 отображен процесс создания таблицы, содержащей 3 строки и 4 столбца.

Можно создать таблицу, используя команду *Вставить таблицу*. При этом в диалоговом окне надо указать число столбцов и строк в создаваемой таблице.

ВЫДЕЛЕНИЕ ЭЛЕМЕНТОВ ТАБЛИЦЫ

Для многих операций с таблицей потребуется выделить ее полностью или частично. Способы выделения с помощью мыши приведены в табл. 2.3.

Таблица 2.3

Выделение элементов таблицы с помощью мыши

Цель	Действие
Выделить ячейку	Щелкнуть левую границу ячейки
Выделить строку	Щелкнуть левую границу строки

Цель	Действие
Выделить столбец	Щелкнуть верхнюю часть столбца таблицы
Выделить несколько ячеек, строк или столбцов	Протащить мышью по требуемым ячейкам (строкам, столбцам) с нажатой левой кнопкой
Выделить всю таблицу	Подвести мышью к левому верхнему углу таблицы и щелкнуть по всплывшему значку с четырехнаправленной стрелкой

Те же действия можно выполнить с помощью кнопки *Выделить* на вкладке *Макет* в группе *Таблица* — в ее выпадающем списке есть команды *Выделить ячейку*, *Выделить строку*, *Выделить столбец*, *Выделить таблицу*.

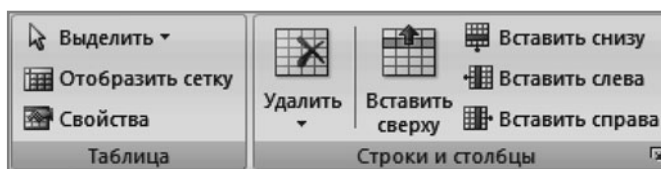


Рис. 2.44. Группы кнопок *Таблица* и *Строки и столбцы* вкладки *Макет*

ДОБАВЛЕНИЕ И УДАЛЕНИЕ ЭЛЕМЕНТОВ ТАБЛИЦЫ

На вкладке *Макет* в группе *Строки и столбцы* сосредоточены кнопки, которые позволяют добавить строки выше и ниже текущей и столбцы — левее и правее текущего. Команды кнопки *Удалить* удаляют выделенные элементы таблицы (ячейки, строки, столбцы).

Можно использовать для этих операций контекстное меню — в нем также имеются соответствующие команды удаления и вставки.

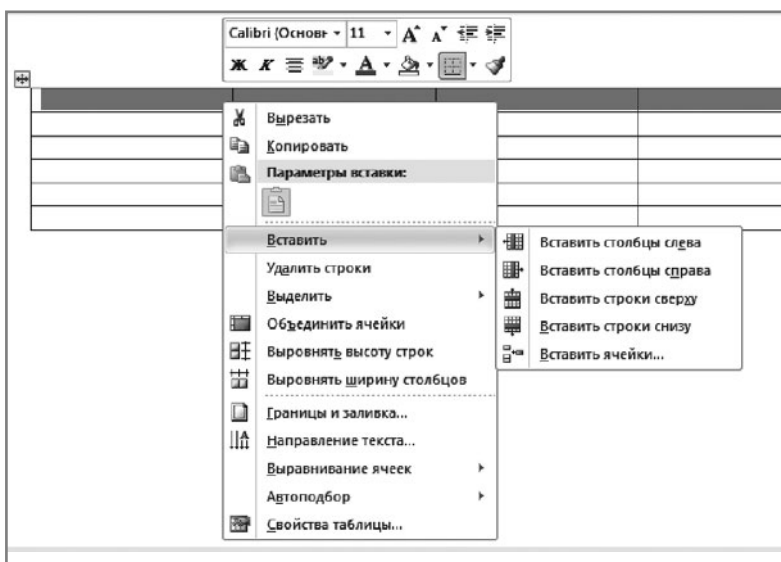


Рис. 2.45. Команды контекстного меню таблицы

Чаще всего для добавления новой строки в таблицу устанавливают курсор справа от той строки, за которой последует вставка, и нажимают клавишу Enter. Это самый простой способ вставки строк.

Самым простым способом удаления элементов таблицы является их выделение и нажатие клавиши Backspace. Однако необходимо иметь в виду, что клавиша Delete удаляет не строки, столбцы, ячейки, а их содержимое (очищает их).

ИЗМЕНЕНИЕ РАЗМЕРОВ ЭЛЕМЕНТОВ ТАБЛИЦЫ

Ширину любого столбца, высоту любой строки, как и отдельных ячеек таблицы, легко изменить перетаскиванием мышью их границ. Для этого надо выделить нужные элементы таблицы, подвести курсор к границе (при этом курсор приобретет форму двойной черты с двунаправленной стрелкой) и перетащить появившуюся пунктирную линию на нужную позицию. Если предварительно никакой элемент таблицы выделен не был, изменения коснутся текущей строки или столбца.

Для изменения размеров очень удобно использовать линейку. Как только курсор устанавливается в таблицу, на горизонтальной и вертикальной линейках появляются метки, соответствующие границам столбцов и строк. Их можно перемещать вдоль линейки, тем самым изменяя высоту строк и ширину столбцов. Если никакой элемент таблицы выделен не был, изменения коснутся всех ее строк/столбцов, иначе — только выделенного блока.

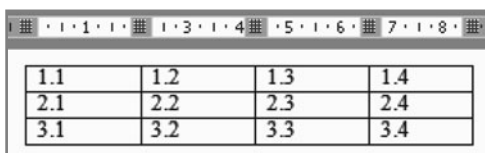


Рис. 2.46. Метки границ таблицы на линейке

ОБЪЕДИНЕНИЕ И РАЗБИЕНИЕ ЭЛЕМЕНТОВ ТАБЛИЦЫ

Часто несколько ячеек таблицы требуется объединить в одну — например, в шапку таблицы.

ОКЕАН	ПЛОЩАДЬ			Средняя глубина, м
	поверхности, млн км ²	островов, млн км ²	общая, млн км ²	
Тихий	178,7	3,9		3957
Атлантический	91,7	1,0		3602
Индийский	76,2	0,8		3736
Северный Ледовитый	14,7	3,8		1131

Рис. 2.47. Пример таблицы с объединенными ячейками

Самый простой способ объединения — выделить требуемые ячейки, вызвать контекстное меню и выбрать из него соответствующую команду *Объединить ячейки*. Эта же команда имеется в группе *Объединить* контекстной вкладки *Макет*.

Вызов операции разбиения ячейки осуществляется аналогично, однако при этом требуется указать, как именно следует ее разбить — на сколько строк и/или столбцов.

ВЕРТИКАЛЬНАЯ ОРИЕНТАЦИЯ ТЕКСТА

Способы оформления текста внутри таблиц такие же, как и при работе со свободным текстом: можно использовать шрифтовое и абзацное форматирование, создавать маркированные и нумерованные списки и т. д. Кроме того, в ячейках таблицы можно выравнивать текст не только по горизонтали, но и по вертикали.

Для этого служат девять кнопок группы *Выравнивание* вкладки *Макет*. Соответствующие команды имеются и в контекстном меню таблицы — в подменю *Выравнивание ячеек*.

Кнопка *Направление текста* работает как переключатель, вращая текст на 90° влево или вправо.

Пример 7

Создать таблицу, изображенную на рис. 2.47.

Решение. Для создания таблицы воспользуемся кнопкой *Таблица* на вкладке ленты *Вставка* и выберем команду *Вставить таблицу*. В появившемся окне укажем число столбцов и строк. Переключатель *Автоподбор ширины столбцов* установим в положение *по ширине окна*, чтобы таблица была растянута на всю ширину страницы.

В результате будет вставлена сетка из указанного числа рядов и колонок.

Далее предстоит объединить две первые ячейки (по отдельности) в первом и последнем столбцах таблицы. Для этого выделим первые 2 ячейки первого столбца и в контекстном меню выберем команду *Объединить ячейки*. Аналогично поступим с первыми двумя ячейками последнего столбца.

Можно приступить к набору данных таблицы. Если потребуется ввести еще строки, например добавить 7-ю строку «Мировой океан», для этого достаточно установить курсор за последней, 6-й строкой и нажать клавишу Enter.

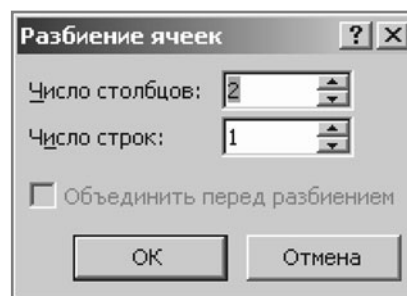


Рис. 2.48. Задание параметров разбиения ячеек

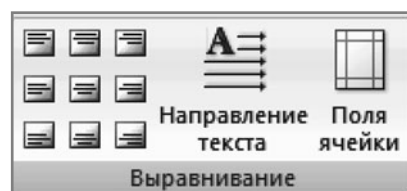


Рис. 2.49. Группа кнопок Выравнивание вкладки Макет

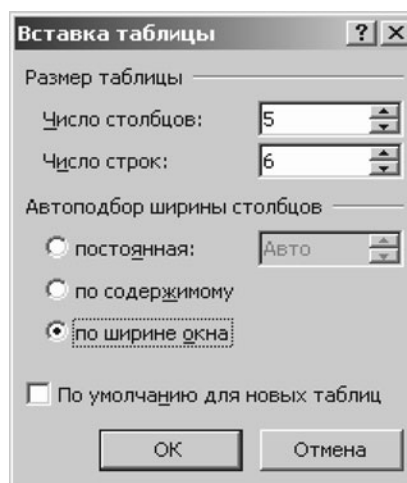


Рис. 2.50. Окно создания таблицы

Для поворота слова «Океан» в первой ячейке воспользуемся кнопкой ленты *Направление текста*.

По окончании ввода данных следует их отформатировать обычным образом.

Вставка в текст различных объектов

ВСТАВКА СПЕЦИАЛЬНЫХ СИМВОЛОВ

При вводе текста часто приходится использовать символы, которых нет на клавиатуре. Это могут быть математические знаки, буквы иных алфавитов, служебные символы и др.

Для вставки таких символов надо щелкнуть на вкладке *Вставка* в группе *Символы* по кнопке *Символ*. В появившемся списке следует выбрать нужный символ. Он будет вставлен в текущую позицию.

Если в списке нет требуемого знака, надо выбрать команду *Другие символы* — откроется диалоговое окно *Символ*, где отображается полный перечень возможных символьных наборов различных шрифтов.

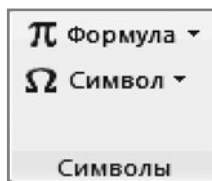


Рис. 2.51. Группа кнопок Символы вкладки Вставка

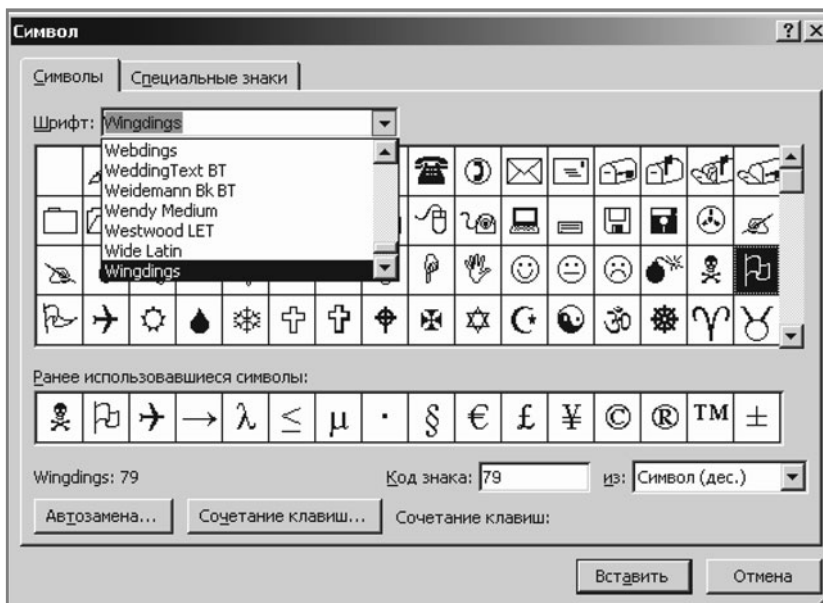


Рис. 2.52. Окно вставки специальных символов (таблица шрифта Wingdings)

В диалоговом окне нужно выбрать шрифт — его таблица будет отображена в центральной части окна. Существуют специализированные шрифты, являющиеся наборами математических символов, музыкальных знаков и т. п. Примером наиболее часто используемых коллекций символов-рисунков служат шрифты Symbol, Webdings, Wingdings, Wingdings2 и Wingdings3.

Кнопка *Вставить* добавляет символ в позицию курсора и не закрывает окно. Можно продолжить выбор и вставку новых символов.

Вторая вкладка — *Специальные знаки* — этого окна позволяет вставлять в текст различные служебные символы: знак конца абзаца, разрывы строк, разделов, колонок, неразрывные пробелы, дефисы и т. п.

ВСТАВКА ФОРМУЛ

В Word встроен формульный редактор, позволяющий набирать математические формулы.

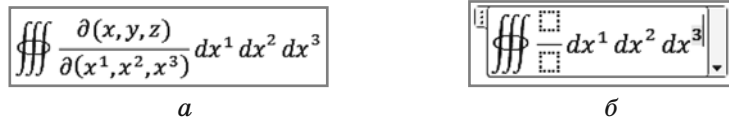


Рис. 2.53. Формула (а) и формульный блок в процессе его заполнения (б)

Для вставки формулы используется команда *Формула*, кнопка которой находится в группе *Символ* вкладки ленты *Вставка*. При ее нажатии в документ помещается формульный блок и открывается контекстно-зависимая вкладка *Работа с формулами/Конструктор*, на которой сосредоточены все команды набора формул.

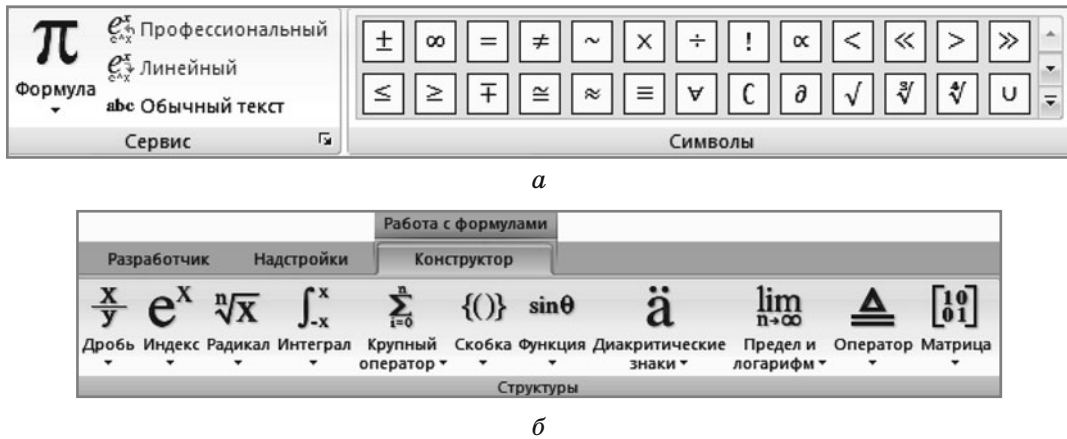


Рис. 2.54. Группы кнопок для работы с формулами: Сервис и Символы (а); Структуры (б)

Для вставки в формульный блок фрагментов формулы следует выбирать соответствующие шаблоны в выпадающих списках кнопок группы *Структуры*. Выбранный шаблон будет вставлен в позицию курсора. На рис 2.55 показаны различные шаблоны дробей в выпадающем списке кнопки *Дроби*.

Шаблоны можно вкладывать один в другой для построения многоступенчатых формул. Так, можно использовать шаблон интеграла, вложив в его подынтегральное выражение, например, шаблон дроби.

Для написания простых формул необязательно использовать вставку объекта, намного легче это сделать с помощью вставки специальных символов: $\alpha \pm \beta \geq \gamma$.



Рис. 2.55. Шаблоны дробей

ВСТАВКА ГРАФИЧЕСКИХ ОБЪЕКТОВ

К **графическим объектам** Word можно отнести рисунки из графических файлов и коллекции клипов MS Office, организационные и обычные диаграммы, надписи и фигурный текст WordArt. Кроме того, средствами Word можно создавать несложные рисунки самостоятельно. Для вставки всех этих объектов используют группы кнопок *Иллюстрации* и *Текст* вкладки ленты *Вставка*.

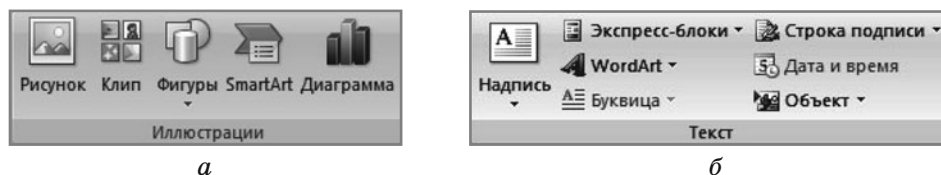


Рис. 2.56. Группы кнопок Иллюстрации (а) и Текст (б) вкладки Вставка

Работа с этими графическими объектами производится с использованием возможностей *контекстных вкладок*, которые появляются при выделении объекта. Для выделения объекта достаточно щелкнуть по нему мышью — вокруг него появятся *ограничивающие маркеры*.



Рис. 2.57. Область задач Клип



Рис. 2.58. Набор встроенных фигур Word

Так, для выделенного рисунка (в том числе рисунка клипа) на ленте добавится вкладка *Работа с рисунками*, для фигуры — *Средства рисования*, для диаграммы — *Работа с диаграммами*, для фигуры SmartArt — *Работа с рисунками SmartArt*.

1. При вставке **рисунков** (кнопка *Рисунок*) на экран выводится окно открытия файлов. Оно позволяет выбрать рисунки из файлов графических форматов JPG, BMP, GIF, TIFF, PNG и др.

2. Вставка **клипов** (кнопка *Клип*) добавляет в текст мультимедийные файлы из коллекций, содержащих рисунки, звуки, анимацию, фильмы. Вместе с MS Office поставляется коллекция готовых клипов при наличии подключения к Интернет можно вставить дополнительные клипы.

3. В поставку Word входит также набор готовых **фигур** нескольких категорий: линии, основные фигуры, фигурные стрелки, блок-схемы, звезды, ленты и выноски. Их можно выбрать в меню кнопки *Фигуры*.

4. **Объекты SmartArt** занимают промежуточное положение между фигурами и диаграммами. Они позволяют представить различную информацию в виде удоб-

ных графических блоков. В окне их выбора для каждого объекта приведен пример, описание в правой панели и всплывающая подсказка. Каждый из блоков объекта впоследствии можно изменять, вписывать в них тексты, форматировать.

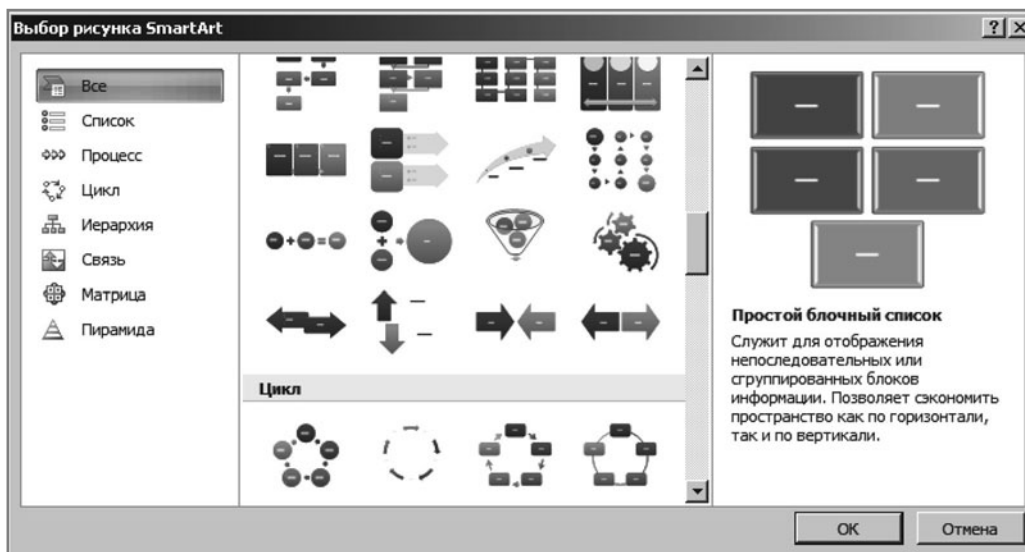


Рис. 2.59. Окно выбора рисунков SmartArt

5. При попытке вставить в документ Word диаграмму (кнопка *Диаграмма*) автоматически запускаются электронные таблицы MS Excel, откуда заимствованы виды диаграмм.

6. Для создания **фигурного текста** предназначена кнопка *WordArt* в группе *Текст* на вкладке ленты *Вставка*. Фигурный текст является графическим объектом, а не текстом. Сначала в выпадающем списке нужно выбрать подходящий стиль из коллекции WordArt, затем ввести текст в появившемся окне, указав параметры его шрифтового оформления. К фигурному тексту можно добавлять тень, наклонять, вращать, растягивать его (с помощью кнопок контекстной вкладки *Работа с объектами WordArt / Формат*).



Рис. 2.60. Пример фигурного текста



Рис. 2.61. Варианты стилей фигурного текста

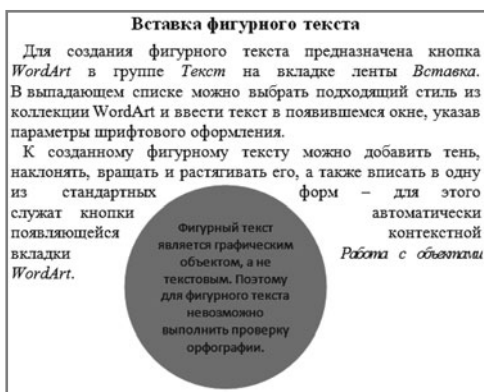


Рис. 2.62. Образец фигурной надписи

7. Кнопка *Надпись* содержит список заготовок для создания надписей различного типа. **Надпись** — это объект векторной графики, в который вставлен текст. Надписи используют при оформлении документов, например, чтобы вынести цитату в центр страницы, разместить дополнительные материалы «на полях» и т. д.

Для удаления любого графического объекта достаточно выделить его и нажать клавишу Delete.

Сервисные функции

ПРОВЕРКА ПРАВОПИСАНИЯ

Проверку правописания в Word можно производить как в процессе работы над документом (программа будет «на лету» выделять ошибки для их немедленного исправления), так и по окончании составления документа по запросу пользователя. Второй способ предпочтителен для больших документов, перегруженных вставленными объектами. Выбор варианта устанавливается в окне настройки параметров Word (вызывается кнопкой *Office/Параметры Word*) — в разделе *Правописание*.

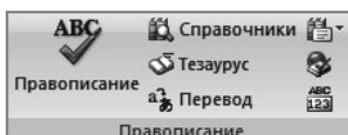




Рис. 2.63. Группа кнопок Правописание вкладки Рецензирование

При постоянном контроле ошибки будут выделяться при наборе текста: красной волнистой линией будут подчеркнуты слова с орфографическими ошибками, зеленой волнистой линией — предложения, нарушающие правила грамматики.

Если была выбрана проверка по запросу пользователя, то запустить ее можно кнопкой  *Правописание*. Она находится в группе *Правописание* на вкладке ленты *Рецензирование*. Проще использовать клавишу F7. Запустится процесс проверки, и при обнаружении ошибок отобразится окно проверки правописания.

Чтобы проверка осуществлялась корректно, надо указать язык, на котором составлен документ. Обычно Word определяет язык автоматически, для установки языка вручную служит кнопка  *Выбрать язык* в группе кнопок *Правописание*. В открывшемся окне можно выбрать язык и установить флажок *Определять язык автоматически*.

В случае обнаружения ошибок Word предлагает помощь в их исправлении. Проще всего для исправления ошибок использовать контекстное меню, выбрав в нем нужный вариант.

Для проверки орфографии Word использует свои встроенные словари. Однако любой язык развивается, поэтому пользователю предоставляется возможность пополнять словарный запас программы. Для этого предназначена команда *Добавить в словарь*.

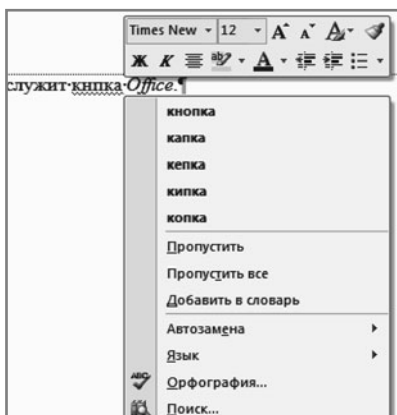


Рис. 2.64. Варианты замены слова в контекстном меню

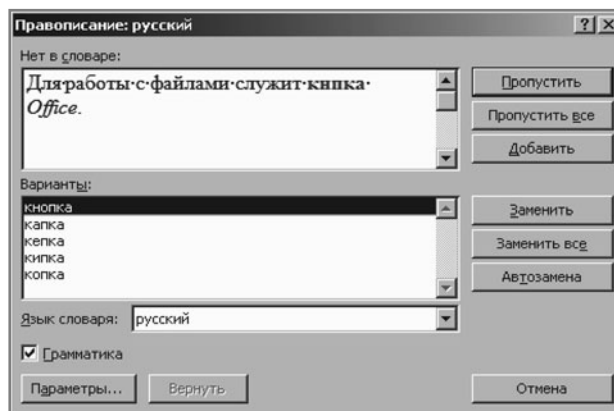


Рис. 2.65. Окно проверки правописания

РАССТАНОВКА ПЕРЕНОСОВ

Для автоматической расстановки переносов служит команда *Расстановка переносов/Авто* вкладки ленты *Разметка страницы* (группа *Параметры страницы*).

Нельзя вставлять переносы вручную с помощью знака «-» (дефис) — после того как в текст будут внесены изменения, слово может сместиться в середину строки, и в нем останется знак переноса. Для вставки переносов вручную служит «мягкий перенос», который проще всего вставить с помощью комбинации клавиш **Ctrl+-**.

2.3.2. Базы данных. Поиск данных в готовой базе.

Создание записей в базе данных

База данных (БД) — совокупность данных, организованная по определенным правилам и предназначенная для длительного хранения во внешней памяти компьютера, постоянного обновления и использования.

БД представляет собой информационную модель определенной предметной области.

Базы данных обеспечивают хранение информации, а также удобный и быстрый доступ к данным. Информация в БД должна быть непротиворечивой, избыточной и целостной.

Для создания БД и манипулирования их данными используют **системы управления базой данных (СУБД)**. Они представляют собой совокупность языковых и программных средств, предназначенных для создания, ведения и использования БД. *Языковые средства* — языки описания и манипулирования данными, языки запросов к БД. *Программные средства* — прикладные программы для обработки данных, содержащихся в БД.

Таким образом, именно СУБД позволяет создавать базы данных, обновлять и дополнять информацию, обеспечивать гибкий доступ к данным. СУБД создает

на экране компьютера определенную среду для работы пользователя (*интерфейс БД*) и имеет определенные режимы работы и систему команд.

Основные функции СУБД:

- создание структуры БД;
- ввод данных в БД и их хранение;
- изменение (редактирование) структуры БД и данных;
- поиск данных в БД;
- сортировка и группировка данных;
- защита БД;
- проверка целостности данных БД.

Примеры современных СУБД: Oracle, Firebird, Interbase, MS SQL Server, MySQL, MS Access.

Классификация баз данных

В зависимости от **характера данных** различают *фактографические* и *документальные* базы данных.

Фактографические БД содержат информацию, относящуюся к определенной предметной области и представленные в строго определенном формате. Например, фактографическими являются базы данных кадрового состава учреждения, транспортных систем, научных данных, сведений о населении, о природных ресурсах и т. д.

Документальные БД служат для хранения и работы с документами. Часто это документы на естественном языке — монографии, публикации в периодике, тексты законодательных актов и т. д. Зачастую документальные БД содержат видео- и звуковые документы.

Современные БД все чаще сочетают в себе признаки документальных и фактографических баз.

По **способу хранения данных** различают *централизованные* и *распределенные* БД.

В **централизованных БД** все данные хранятся на одном компьютере (локальном или сетевом). В **распределенных БД** части базы сохраняются на множестве компьютеров, объединенных между собой в сеть.

В зависимости от вида **организации данных** различают следующие основные модели представления данных в базе:

- иерархическую;
- сетевую;
- реляционную.

Наиболее распространенной из этих моделей является реляционная. Сегодня появляются новые виды организации БД, как правило, имеющие в своей основе реляционный подход.

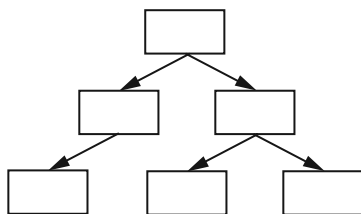


Рис. 2.66. Схема иерархической модели

В **иерархической модели** данные представляют в виде древовидной (*иерархической*) структуры. В ней существует строгая подчиненность элементов, один из которых главный, остальные подчиненные. Такая организация данных удобна для работы с информацией,

упорядоченной по уровням, однако для данных со сложными логическими связями она оказывается слишком громоздкой. Графическое представление модели — дерево.

В **сетевой модели** данные организуются в виде произвольного графа, т. е. связи между элементами произвольны. Она более гибкая, в ней нет явно выраженного главного элемента, существует возможность установления горизонтальных связей. Недостатком ее является высокая сложность реализации.

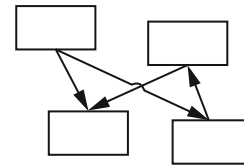


Рис. 2.67. Схема сетевой модели

Кроме того, значительным недостатком иерархической и сетевой моделей является то, что структура данных задается на этапе разработки БД и не может быть изменена позже, при организации доступа к данным.

Реляционная модель представляет собой совокупность таблиц, которые также называют *отношениями* (отсюда и название: *англ.* relation — отношение). Эта модель была предложена в 70-х годах XX века сотрудником фирмы IBM Эдгаром Коддом. Она основана на теории множеств и математической логике. Доказано, что любую структуру данных можно свести к табличной форме. Достоинствами реляционной модели данных являются простота, гибкость структуры, удобство компьютерной реализации, наличие теоретического описания. Большинство современных БД для персональных компьютеров являются реляционными.

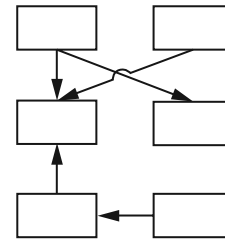


Рис. 2.68. Схема реляционной модели

В реляционной БД эффективно реализуются следующие операции:

- сортировка данных (например, учащихся по алфавиту, сеансов по времени);
- выборка данных по группам (например, все маршруты по направлениям, все учащиеся по классам);
- поиск записей (например, поиск книг по их автору, товаров по наименованию) и т. д.

Таблицы реляционной БД

Каждая таблица состоит из строк и столбцов и предназначена для хранения данных об однотипных объектах. Каждая строка содержит информацию об одном конкретном объекте БД (например, книге, сотруднике, товаре), а каждый столбец — конкретную характеристику этого объекта (например, фамилию, название, цену). Строки такой таблицы называются **записями**, столбцы — **полями**. Каждое поле должно иметь *имя*, уникальное в пределах таблицы.

В разных записях значения отдельных полей могут повторяться. Но в таблице должно иметься такое поле или набор полей, значение в котором не повторяется ни для одной записи — оно является *уникальным* для каждой записи. Такое поле называют **ключевым полем (ключом)** таблицы. По значению ключа

чегого поля можно однозначно получить всю запись таблицы с этим ключевым значением.

Простой ключ состоит из одного поля, а *составной* — из нескольких полей. В таблице может быть определен только один ключ, который называют *первичным ключом*. Например, таким полем может служить код изделия, номер автомобиля или читательского билета.

Поле
↓

Запись →	Ключ	Поле 1	Поле 2	Поле 2
	Запись 1	Значение	Значение	Значение
	Запись 2	Значение	Значение	Значение
	Запись 3	Значение	Значение	Значение

Рис. 2.69. Структура таблицы базы данных

Любая таблица в реляционной БД обладает следующими свойствами:

- каждый элемент таблицы — один элемент данных;
- все поля (столбцы) в таблице являются однородными, т. е. данные в одном столбце имеют один *тип* (число, текст, дата и т. д.);
- каждое поле имеет уникальное имя;
- одинаковые записи (строки) в таблице отсутствуют;
- порядок следования записей в таблице может быть произвольным.

Каждое поле таблицы имеет определенный тип. **Тип** — это множество значений, которые может принимать поле, и множество операций, которые можно выполнять над этими значениями. Существуют четыре основных типа для полей БД: *символьный*, *числовой*, *логический* и *дата*.

Поля **символьного** типа предназначены для последовательностей символов (тексты, коды и т. д.). Поля **даты** содержат даты вида «день/месяц/год». **Числовые** поля могут сохранять только числа. **Логические** поля сохраняют данные, которые могут принимать только одно из двух значений: «да» или «нет» («истина» или «ложь»).

Пример 1

В таблице сведений об автомобилях требуется сохранять сведения о моделях автомобилей, их мощности, цвете кузова и времени выпуска. Каждый автомобиль характеризуется своим кодом. Определить структуру таблицы.

Решение. Перечисленные данные будут составлять поля таблицы. Таким образом, в таблице могут быть определены поля (столбцы) *Код автомобиля*, *Модель*, *Мощность двигателя*, *Цвет*, *Год выпуска*.

Поле *Код автомобиля* — ключевое: оно однозначно определяет любую запись таблицы.

Типы всех полей, кроме *Год выпуска*, — символьные; *Год выпуска* — числовое. Поле *Код автомобиля* содержит текстовые символы, поэтому имеет символьный тип. Поле *Год выпуска* содержит не дату, а число, поэтому имеет тип числовой.

Записи таблицы содержат данные о конкретных автомобилях.

Поле
↓

Запись →

Код автомобиля	Модель	Мощность двигателя	Цвет	Год выпуска
V1288	Volkswagen Polo	86/5000	серый	2011
V390	Volkswagen Golf	170/5700	зеленый	2010
ТС50	Toyota Corolla	124/6000	красный	2013
ТСА188	Toyota Camry	268/6200	серый	2013
R5995	Renault Laguna	150/6000	синий	2009
SI58	Subaru Impreza	100/5500	серебристый	2012

Пример 2

Определить поля и записи в представленной таблице клиентов. Указать, какое поле может быть в этой таблице ключевым.

Номер клиента	Фамилия	Имя	Город	Телефон
110	Блинов	Иван	Харьков	454-16-78
111	Веснин	Игорь	Курск	482-82-36
112	Воронов	Андрей	Киров	242-55-89
113	Воронов	Павел	Иркутск	167-71-32
114	Кралева	Мария	Минск	963-51-13
115	Нечаева	Дарья	Харьков	760-65-19
116	Федоренко	Павел	Воронеж	820-92-21

Решение. Каждый столбец таблицы содержит некоторую характеристику объекта (клиента). Эти столбцы являются *полями* таблицы: *Номер клиента, Фамилия, Имя, Город, Телефон.*

Среди всех полей только *Номер клиента* однозначно определяет конкретного клиента. Поля *Фамилия* или *Имя* для этой цели не подходят, т. к. они содержат повторяющиеся фамилии и имена. Даже если бы предложенная таблица не держала однофамильцев, то такой вариант мог возникнуть позже, при дополнении базы данных. Поэтому поле *Фамилия* не может однозначно характеризовать клиента. При условии что каждому клиенту присвоен свой, уникальный номер, поле *Номер клиента* может быть ключевым полем данной таблицы.

Каждая строка таблицы содержит сведения об одном объекте (клиенте), поэтому строки таблицы — *записи*.

Связи между таблицами

Как правило, реляционная БД состоит из набора взаимосвязанных таблиц, некоторые из них являются *главными*, остальные — *подчиненными*. Организация связи (отношений) между таблицами называется **связыванием** или **соединением таблиц**.

Связи между таблицами устанавливаются при создании БД. Поля, которые используются для связывания таблиц, называются *полями связи*. Поле связи подчиненной таблицы называется **внешним ключом**.

Существуют следующие виды связей между таблицами:

- отношение «один-к-одному»;
- отношения «один-ко-многим» и «много-к-одному»;
- отношение «много-ко-многим».

Отношение «один-к-одному» означает, что одной записи в главной таблице соответствует не более одной записи в подчиненной таблице. Полями связи в таком случае являются ключевые поля обеих таблиц.

Отношение «один-к-одному» обычно используют, чтобы разбить таблицу с большим числом полей на несколько таблиц. В этом случае в первой таблице остаются поля с наиболее важной и часто используемой информацией, а остальные поля переносятся в другую таблицу. Например, в базе данных библиотеки в первой таблице — поля с названием книги, фамилией ее автора, годом издания, а во второй — поля с датой приобретения книги, местом ее хранения и т. д.

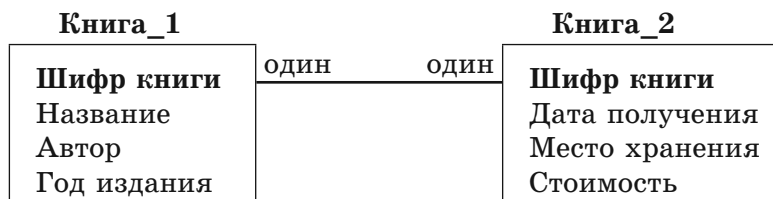


Рис. 2.70. Отношение «один-к-одному»

Отношение «один-ко-многим» означает, что одной записи главной таблицы может соответствовать несколько записей подчиненной таблицы, а каждой записи подчиненной — только одна запись главной таблицы. Это наиболее часто встречающийся вид отношений.

Например, в главной таблице хранятся данные о книгах, а в подчиненной — данные о выдаче книг читателям. Одну книгу могут брать в библиотеке для чтения многократно, поэтому одной записи в главной таблице может соответствовать несколько записей в подчиненной таблице.

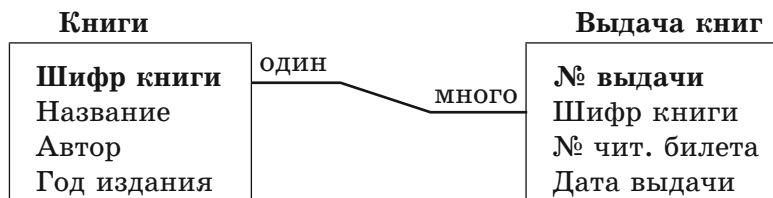


Рис. 2.71. Отношение «один-ко-многим»

Отношение «много-к-одному» отличается от отношения «один-ко-многим» только направлением. Если на отношение «один-ко-многим» посмотреть со стороны подчиненной таблицы, а не главной, то оно превращается в отношение «много-к-одному».

Отношение «много-ко-многим» означает, что каждой записи одной таблицы может соответствовать несколько записей другой таблицы, и в то же время каждой записи второй таблицы — несколько записей первой.

Подобное отношение реализуется, например, в базе данных библиотеки между таблицами книг и читателей. Любую книгу могут взять для чтения (по очереди) много читателей, и любой читатель может взять несколько книг.

Для отношения «много-ко-многим» сложно организовывать связи между таблицами и взаимодействие между их записями. Многие СУБД, в том числе MS Access, не поддерживают организацию подобного отношения. В таких случаях его реализуют через отношение «один-ко-многим». Например, отношением «один-ко-многим» связаны таблица книг и таблица выдачи книг читателям, и таким же отношением связаны таблица читателей и таблица выдачи книг читателям.

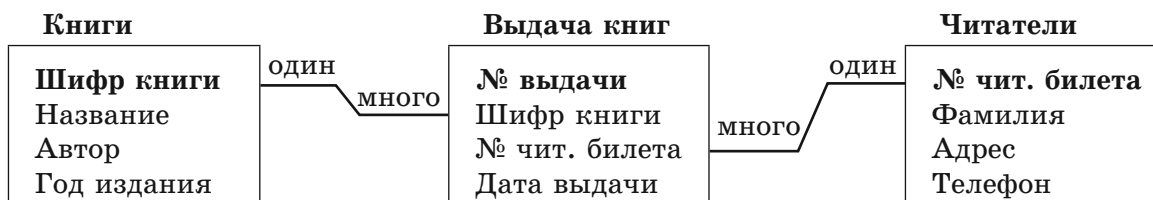


Рис. 2.72. Замена отношения «много-ко-многим» на «один-ко-многим»

Добавление, удаление, поиск, сортировка записей таблицы

Информация БД постоянно обновляется. Способы добавления и удаления записей в ее таблицах зависят от возможностей, предоставляемых СУБД.

Наиболее просто это осуществляется в тех СУБД, которые имеют графический интерфейс. Например, для добавления записи в таблицу MS Access достаточно установить курсор в последнюю строку таблицы и ввести данные в поля. Эта последняя строка всегда пустая и помечена слева символом *. Чтобы быстро попасть в нее, можно воспользоваться кнопкой *Новая (пустая) запись* в строке состояния таблицы.

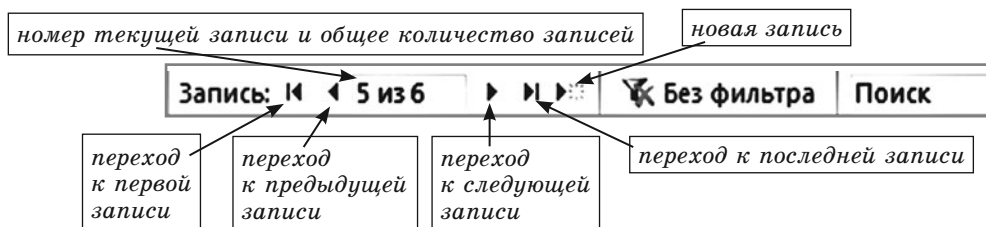


Рис. 2.73. Строка состояния таблицы с кнопками добавления и перехода

Для перехода от одного поля к другому используют клавишу Enter. Переходить от одной записи к другой можно с помощью клавиши Enter, мыши или кнопок переходов в строке состояния. Для удаления записи достаточно выделить ее (щелкнуть мышью слева от нее, в зоне выделения) и нажать клавишу Delete.

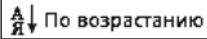

Как правило, для ввода данных в таблицы создают специальные объекты базы данных — *формы*. Они содержат поля одной или нескольких таблиц, а также вспомогательные элементы, обеспечивающие удобство ввода и редактирования данных.

Для **поиска данных** в таблице существуют разные способы. Например, можно воспользоваться автоматическим поиском по содержимому таблицы и переходить от одного найденного значения к другому. Иной способ — упорядочить (отсортировать) записи таблицы определенным образом, что облегчит поиск нужной информации. Наконец, можно использовать механизм отбора данных — фильтрацию и запросы.

В MS Access для **поиска значения** достаточно ввести фрагмент искомого значения в поле *Поиск* строки состояния таблицы. Уже в процессе ввода курсор переместится в соответствующую запись. Для перехода к следующему результату надо нажать клавишу Enter.

Сортировка данных в таблице — это упорядочение ее записей. Для сортировки надо указать поля сортировки и порядок (по возрастанию или по убыванию значений). Записи в таблице сортируются целиком.

Например, при сортировке таблицы абитуриентов по дате поступления записи с ранней датой будут целиком перемещены вверх по отношению к записям с поздней датой.

В MS Access для **сортировки** служат команды  и  в заголовке каждого поля таблицы и на вкладке ленты *Главная*.

Сортировка полей типа «дата» учитывает календарную последовательность.

Порядок сортировки полей символьного типа (строк) определяется таблицей кодировки символов. Строки сравниваются посимвольно слева направо до первого несовпадающего символа. Символ, имеющий меньший код, определит меньшее значение всего поля. Например, значение строки «март» больше значения «май», т. к. код символа «р» больше кода символа «й». Короткое значение считается меньше длинного, если короткое полностью совпадает с началом длинного. Так, значение «компьютер» меньше значения «компьютерный».

Если строки имеют одинаковую длину и содержат одну и ту же последовательность символов, то они равны.

Пример 3

Сравнить значения полей «Полевой» и «поле».

Решение. При сравнении первых же символов обнаруживаем, что код символа «П» меньше кода «п». Следовательно, вся строка «Полевой» меньше, чем «поле» (несмотря на то что слово «поле» короче, строки полностью не совпадают).

Пример 4

Указать порядок записей в таблице после сортировки ее по полю *Класс*.

№ по списку	Фамилия	Имя	Класс
1	Чепраков	Николай	7а
2	Кротов	Аркадий	5б
3	Крылов	Павел	9а
4	Незнамов	Андрей	9б
5	Чернова	Мария	10а
6	Шепелева	Татьяна	5а

Решение. Учитывая, что в поле *Класс* значения содержат буквы, — это поле символьного типа. Тогда при сравнении его значений сначала будут располагаться те, у которых меньше код первого символа. Первым будет значение «10а» (код его первого символа «1» — наименьший из всех). За ним следует значение «5а», затем «5б» (первые символы «5» у них одинаковы, у вторых код «а» меньше кода «б»). Следующий первый символ — «7» — у значения «7а». И наконец, последними будут значения «9а» и «9б».

В соответствии с этим порядком будут перемещены записи таблицы в результирующем наборе: сначала 5-я, затем 6-я, 2-я, 1-я, 3-я, 4-я.

Такой результат сортировки символьных значений (10-й класс оказался раньше 5-ого) может оказаться неожиданным для пользователей. Поэтому его учитывают при разработке БД (например, создают два поля для названия класса — отдельно для номера и буквы).

Ответ: Порядок записей — 5, 6, 2, 1, 3, 4.

Отбор записей таблицы

Зачастую требуется отобрать часть записей таблицы, которые удовлетворяют некоторому условию. Это условие называют **критерием отбора**.

Отбор записей в самой таблице путем задания условия отбора называется **фильтрацией** таблицы. Фильтрация позволяет отобрать записи из одной таблицы для простых условий. Если требуется выбрать данные сразу из нескольких таблиц или условие является сложным, создают специальные объекты базы данных — **запросы**. Они представляют собой средства получения данных из БД в соответствии с требованиями пользователя.

И в том и в другом случае условие отбора должно быть записано в виде логического выражения. **Логическое выражение** — выражение, содержащее логические и арифметические операции, а также операции сравнения. Результатом логического выражения является логическая величина («истина» либо «ложь»). К логическим выражениям применяются все правила вычисления результатов логических операций.

Простейшая форма логического выражения в условии отбора — указание имени какого-либо поля логического типа. Поскольку такие поля имеют значения «истина» или «ложь», то результат выражения совпадет со значением поля.

Простые логические выражения могут содержать операции сравнения (их еще называют отношениями): больше ($>$), меньше ($<$), больше или равно ($>=$), меньше или равно ($<=$), равно ($=$), не равно (\neq или $<>$). Сравнение данных производится по тем же правилам, что и при сортировке.

Операции сравнения накладывают требования на значения поля. Например, выражение *Количество* $<=$ 100 требует найти все записи таблицы, в которых значение поля *Количество* не превышает 100.

Результатом отбора являются те записи таблицы, для полей которых логическое выражение принимает значение «истина».

Пример 5

Для приведенной таблицы определить результат отбора в соответствии с заданными логическими выражениями. Поля *Отправление* и *Назначение* — символьные, *Доставка* — логическое, остальные — числовые.

- А) Отправление = "Женева"
 Б) Доставка
 В) Назначение >= "С"

№ поездки	Отправление	Назначение	Расстояние	Доставка
12	Женева	Амстердам	1038	да
168	Брюссель	Берлин	789	нет
308	Мадрид	Бухарест	3503	нет
3	Афины	Амстердам	3070	да
558	Хельсинки	Лиссабон	4055	нет
23	Осло	Амстердам	1397	да
16	Киев	Амстердам	2070	нет
460	Женева	Стамбул	2399	нет
108	Женева	Барселона	808	нет

Решение.

А) В результате будут отобраны записи, содержащие в поле *Отправление* значение "Женева". Это строки с номерами поездок 12, 460, 108.

Б) В этом логическом выражении участвует только название логического поля *Доставка*. Результатом отбора будут все записи, у которых это поле содержит значение «истина» («да»), т. е. записи о поездках с номерами 12, 3, 23.

В) Требуется найти значения в поле *Назначение*, которые больше значения "С". Это должны быть строки, которые начинаются с символа «С», но длиннее; или строки, которые начинаются с символа, код которого больше кода «С» (т. е. следующие по алфавиту). Из всех строк в поле *Назначение* подходит только значение «Стамбул» — все остальные начинаются с символов, предшествующих "С" в алфавите. Результатом будет запись о поездке с номером 460.

Ответ: результаты отбора — записи о поездках с номерами: А) 12, 460, 108; Б) 12, 3, 23; В) 460.

Составные логические выражения конструируются из простых при помощи логических операций И, ИЛИ, НЕ. Эти операции удовлетворяют правилам алгебры логики. Порядок выполнения операций определен в алгебре логики (сначала НЕ, затем И, затем ИЛИ). Для изменения этого порядка используются скобки.

Результат применения операции И (логическое умножение) равен «истина» только тогда, когда обе части выражения истинны.

Результат применения операции ИЛИ (логическое сложение) равен «истина» только тогда, когда хотя бы одна из частей выражения истинна.

Результат применения операции НЕ (логическое отрицание) к какому-либо логическому выражению — «истина» тогда, когда исходное выражение ложно, и «ложь» — когда оно истинно.

Пример 6

Для таблицы, приведенной в примере 5, определить результат отбора в соответствии с заданными логическими выражениями.

- А) (Отправление = "Женева") И Расстояние > 1000
- Б) (Отправление = "Женева") И Доставка
- В) (Отправление = "Женева" ИЛИ Отправление = "Киев") И (Назначение = "Амстердам" ИЛИ Назначение = "Берлин")

Решение.

А) Сначала отбираются строки, удовлетворяющие первой части выражения (записи о поездках из Женева — с номерами 12, 460, 108). Затем из них отбираются строки, которые удовлетворяют второй части выражения (записи о поездках с расстоянием свыше 1000 — с номерами 12 и 460).

Б) Сначала отбираются записи таблицы, удовлетворяющие первой части условия (о поездках с номерами 12, 460, 108). Затем из них отбираются те, в которых поле *Доставка* имеет значение «истина». Из отобранных строк только запись о поездке с номером 12 имеет значение поля *Доставка* «да» («истина»).

В) В соответствии со скобками сначала отбираются записи, удовлетворяющие первой части логического выражения. Затем — второй его части, после чего из них отбираются записи, удовлетворяющие обоим частям выражения (попавшие и в один, и в другой результаты отбора).

Первой части условия удовлетворяют записи с номерами поездок 12, 16, 460, 108. Второй части условия — записи с номерами поездок 12, 168, 3, 23, 16. Записи, попавшие в оба результата отбора, имеют номера 12, 16.

Ответ: результаты отбора — записи о поездках с номерами: А) 12, 460; Б) 12; В) 12, 16.

Пример 7

Для таблицы, приведенной в примере 5, записать условие отбора записей о поездках:

- А) в пункт назначения Стамбул на расстояние не свыше 1000;
- Б) из Женева или Брюсселя на расстояние от 1000 до 2000.

Решение.

А) Условие о пункте назначения представляет логическое выражение Назначение = "Стамбул".

Условие об ограничении расстояния: Расстояние \leq 1000.

По условию оба эти ограничения должны выполняться одновременно, поэтому для объединения следует использовать операцию логического умножения:

(Назначение = "Стамбул") И (Расстояние \leq 1000).

Если применить это условие отбора к исходным данным, то в результате не будет получено ни одной строки, так как в таблице нет ни одной записи, удовлетворяющей условию. Тем не менее, это не означает, что условие записано неверно, а лишь то, что в данный момент времени таких поездок не имеется.

Б) для отбора поездок из Женева или Брюсселя должно быть записано условие Отправление = "Женева" ИЛИ Отправление = "Брюссель".

Для ограничения расстояния: Расстояние \geq 1000 И Расстояние \leq 2000.

Затем оба этих условия надо объединить логической операцией И, так как они должны быть выполнены одновременно:

(Отправление = "Женева" ИЛИ Отправление = "Брюссель") И Расстояние \geq 1000 И Расстояние \leq 2000.

Вторую часть выражения можно заключить или не заключать в скобки, поскольку обе операции И имеют одинаковый приоритет выполнения.

2.3.3. Рисунки и фотографии. Ввод изображений с помощью инструментов графического редактора, сканера, графического планшета, использование готовых графических объектов. Геометрические и стиливые преобразования. Использование примитивов и шаблонов

Получение графических изображений

К графическим изображениям относятся рисунки, чертежи, графики, диаграммы, иллюстрации, в том числе и трехмерные. Они могут быть получены в компьютере одним из двух способов: введены с одного из специализированных устройств или созданы самостоятельно.

Для ввода в компьютер графической информации используются различные устройства, например *фотоаппараты, графические планшеты, сканеры, микроскопы* и т. п.

Графические планшеты могут использоваться как для рисования, так и для ввода рукописного текста. С помощью специальной ручки можно чертить, рисовать схемы, добавлять заметки и подписи к электронным документам.

К основным техническим характеристикам графических планшетов относятся *разрешающая способность* и способность реагировать на *силу нажатия* пера. Разрешающая способность измеряется в *lpi* (lines per inch — количество линий на дюйм): количество точек на экране монитора, на которые сместится курсор при перемещении пера по планшету на 1 дюйм. В современных планшетах разрешающая способность достигает 2048 lpi, а количество воспринимаемых градаций нажатий на перо составляет 1024.

Для оцифровки изображений (оптического ввода в компьютер и преобразования в электронную форму) применяются **сканеры**. В основе работы всех сканеров лежит один и тот же принцип. Сканируемое изображение освещается белым светом (в черно-белых сканерах) или тремя основными цветами (красным, зеленым и синим). Отраженный свет воспринимается светочувствительными элементами, которые последовательно считывают изображение и преобразуют его в компьютерный формат. Минимальный элемент изображения интерпретируется сканером как цветная (или серая) точка. Таким образом, в результате сканирования документа создается графический файл, в котором хранится растровое изображение исходного документа (состоящее из точек).

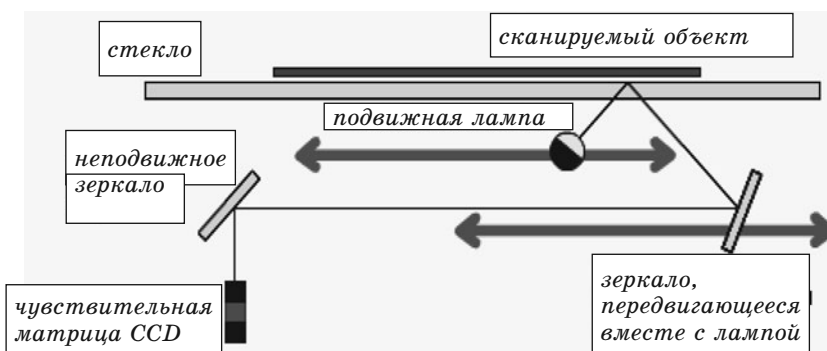


Рис. 2.74. Принцип действия планшетного сканера

Количество точек определяется как размером изображения, так и разрешением сканера. **Разрешение** — количество отснятых точек на дюйм* изображения, которые сканер получает с объекта в процессе работы. В современных сканерах уровень разрешения составляет не менее 600 dpi (dots per inch — точек на дюйм).

Кроме разрешения, сканирование характеризуется **глубиной цвета** — количеством оттенков, которые устройство способно распознать. Однобитное сканирование воспроизводит все тона исходного изображения как черный или белый (2 уровня). 8-битное сканирование вводит $2^8 = 256$ различных уровней серого. Глубина цвета в 24 бита соответствует $16\,777\,216 (= 2^{24})$ цветовых оттенков. С ростом разрядности увеличивается и количество деталей изображения, которые может вводить сканирующее устройство.

Пример 1

Рассмотрим организацию сканирования с использованием планшетного сканера. Обычно процесс включает следующие этапы.

1. Включить сканер и положить на стекло под крышкой сканируемый документ.

2. Запустить программное обеспечение, поставляемое вместе с устройством. Для сканирования могут быть использованы программы других производителей, в т. ч. такие, которые осуществляют последовательно сканирование и распознавание, однако они все равно будут использовать ПО производителя.

3. Указать тип сканируемого документа. Например, для фотосканера могут быть варианты: *фото, позитив, цветной или черно-белый негатив, рисунок, журнал, газета, текст/штриховой рисунок*.

4. Задать тип изображения:

- *цветное* — для сканирования высококачественных изображений в 24-битном цвете;
- *оттенки серого* — для сканирования высококачественных изображений в 8-битных оттенках серого;
- *черно-белое* — для сканирования штриховых рисунков или черно-белых изображений в черновом качестве.

5. Задать приложение или устройство вывода информации, для которого сканируется изображение (параметр *Назначение*). При выборе значений *Экран* или *Принтер* автоматически устанавливается разрешение сканирования — соответственно 96 и 300 dpi. Если требуется, можно изменить это значение. Для скани-

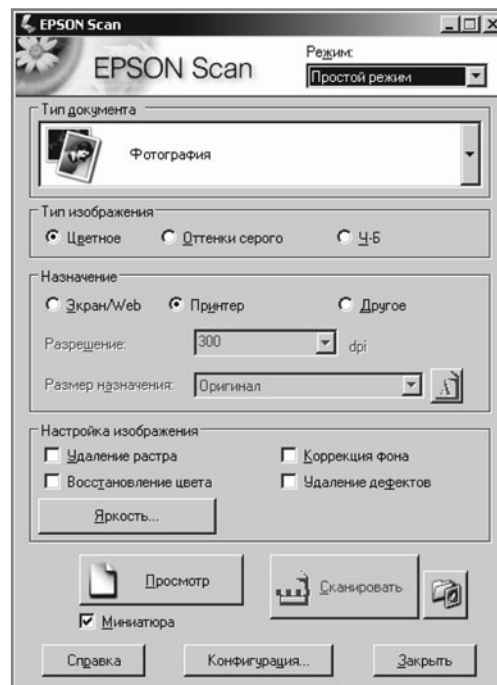


Рис. 2.75. Окно настройки сканирования (простой режим) для сканера Epson

* Один дюйм равен 2,54 см.

рования распечатанных на принтере черно-белых документов лучше устанавливать режим *оттенки серого* и разрешение 300 dpi.

6. Запустить предварительное сканирование (кнопка *Просмотр*). В окне предварительного просмотра отразится первоначальный результат. Если результат просмотра будет неудовлетворительным, следует изменить настройки и повторно нажать кнопку *Просмотр*. Если требуется — задать поворот просматриваемого изображения. Можно обвести мышью фрагмент, который будет отсканирован.

7. Если все операции выполнены успешно, задать в основном окне сканирования папку назначения и имя создаваемого файла, после чего нажать кнопку *Сканировать*.

Создание графических изображений

ГРАФИЧЕСКИЕ РЕДАКТОРЫ

Создание изображений возможно с помощью инструментов различных графических редакторов. **Графический редактор** — это программа, предназначенная для создания и изменения графического изображения на экране компьютера, а также его сохранения в виде графического файла.

Графический редактор, обладающий дополнительными интеллектуальными средствами, называют *графическим процессором*. Такие программы позволяют обрабатывать изображения с помощью разнообразных графических эффектов, преобразовывать формат, палитру, масштаб, работать с многослойными изображениями, получать изображения со сканера и другой цифровой техники и т. д. Любой графический редактор включает в себя текстовый редактор и позволяет набирать тексты.

Графические редакторы с большим диапазоном функций называют *графическими пакетами*. В зависимости от *методов построения и области применения* различают группы таким программ:

- пакеты *иллюстративной графики* предназначены в основном для создания и редактирования рисунков. К ним, например, относят Adobe Photoshop и CorelDRAW — пакеты для создания двумерных изображений, 3ds Max — для создания трехмерных изображений, Adobe Flash для создания анимационных фильмов;
- средства *научной графики* используются для решения научных и производственных задач. Современная научная компьютерная графика дает возможность проводить вычислительные эксперименты с наглядным представлением их результатов;
- средства *деловой графики* предназначены для наглядного представления деловой активности. Обычно такие программные средства включаются в состав электронных таблиц и мощных офисных пакетов;
- пакеты *инженерной и конструкторской графики* используются в основном для создания различных графиков, диаграмм, чертежей, например, AutoCAD, ArchiCAD, Grapher, GraphPlus;
- программы *компьютерной анимации* создают движущиеся изображения на экране монитора. Эти программы выполняют расчеты движений, опирающиеся на их математическое описание, и изображают все промежуточные состояния между начальным и конечным положением движущихся объектов.

В зависимости от *методов обработки и сохранения изображений* различают *растровые* и *векторные* графические редакторы. Многие графические программы могут обрабатывать только векторные или только растровые изображения, но существуют программы, в которых можно обрабатывать оба типа файлов.

Растровое графическое изображение представляет собой совокупность данных о цвете отдельных элементов изображения — **пикселей**, образующих строки и столбцы (матрицу точек). Такие изображения имеют высокое качество, но требуют больших объемов памяти из-за необходимости сохранять информацию о каждой точке изображения. Кроме того, растровые изображения плохо поддаются масштабированию (уменьшению и увеличению размера) и другим преобразованиям.

Растровые графические редакторы, работающие с растровыми изображениями, широко применяются в работе художников-иллюстраторов, при подготовке изображений к печати типографским способом или на фотобумаге, публикации в Интернете. Многие современные растровые редакторы содержат и векторные инструменты редактирования в качестве вспомогательных. К растровым редакторам относятся Adobe Photoshop, Corel Paint Shop Pro, Microsoft Paint, Microsoft Office Picture Manager, Paint.NET и др.

Векторные изображения состоят из *геометрических примитивов*, таких как точки, линии (прямые, кривые, дуги окружностей) и многоугольники. Эти изображения сохраняются в памяти компьютера в виде набора параметров математических формул, описывающих примитивы, толщину и цвет линий, цвет заливки сплошных областей и др. Такие изображения легко масштабируются, при этом характеристики изображения пересчитываются по формулам для нового размера. Векторные изображения занимают небольшой объем памяти, однако имеют не такую богатую палитру, как растровые.

Векторные графические редакторы, позволяющие создавать и редактировать векторные изображения, широко используются для разработки и создания печатной продукции: научных иллюстраций, брошюр, буклетов, визиток, этикеток. Наиболее популярны такие векторные редакторы, как CorelDRAW, Adobe Illustrator, Adobe Flash, Macromedia FreeHand, OpenOffice.org Draw.

Программы для работы с **трехмерной графикой (3D-редакторы)** и **анимацией** могут использовать как векторные, так и растровые изображения. Они оперируют объектами в трехмерном пространстве (которые представляют собой набор поверхностей или частиц), но результаты обычно имеют вид плоской картинки, проекции. Трехмерная компьютерная графика широко используется в кино, компьютерных играх, в архитектурной визуализации, в печатной продукции, а также в науке и промышленности. Стандартом 3D-графики в кино и на телевидении стал графический редактор Maya, в других областях широко используется графический программный пакет 3ds Max.

ХАРАКТЕРИСТИКИ ИЗОБРАЖЕНИЙ

Растровые изображения характеризуются пространственным разрешением и глубиной цвета. **Пространственное разрешение** — величина, определяющая количество точек (пикселей) по горизонтали и вертикали. **Глубина цвета** — объем памяти (в битах) для представления цвета одного пикселя растрового изображения. Чем больше пространственное разрешение и глубина цвета, тем качественнее изображение и тем больший объем оно занимает в памяти компьютера.

Для количественного описания цвета используются **цветовые модели**. Они задают параметры формирования цвета, например, смешиванием нескольких цветовых оттенков или указанием тона, контраста и яркости и т. д.

В широко распространенной модели **RGB** (сокращение от *англ.* Red, Green, Blue) каждый цвет создается с помощью смешения оттенков трех основных цветов: красного, зеленого и синего. Выбор этих цветов в качестве основных обусловлен особенностями человеческого восприятия цвета. Например, для получения желтого цвета необходимо смешать красный и зеленый, для фиолетового — синий и красный и т. д.

Чтобы указать цвет пикселя в модели RGB, необходимо задать *яркость (интенсивность)* каждой из трех составляющих в указанном порядке. Значения интенсивности варьируются от 0 (минимальная яркость) до 255 (максимальная). При максимальной интенсивности красного (255) и минимальной — зеленого и синего (по 0) получим красный цвет, который записывается как тройка чисел (255, 0, 0). Аналогично, например, зеленый цвет представлен максимальной интенсивностью зеленого и минимальной — всех остальных (0, 255, 0). Представления, например, (0, 200, 0) или (0, 127, 0) тоже задают оттенки зеленого цвета, но более темные (меньшей интенсивности).

Если все основные цвета имеют максимальную интенсивность, получим белый цвет (255, 255, 255), при минимальной интенсивности — черный (0, 0, 0). Варианты серого цвета получаются, если все основные цветовые составляющие равны между собой по интенсивности.

Таблица 2.4

Некоторые варианты получения цвета в модели RGB

Цвет	Красный (R)	Зеленый (G)	Синий (B)
Черный	0	0	0
Синий	0	0	255
Зеленый	0	255	0
Бирюзовый (сине-зеленый)	0	255	255
Красный	255	0	0
Пурпурный	255	0	255
Желтый	255	255	0
Белый	255	255	255

Модель RGB используется для изображений, предназначенных для просмотра на экране монитора. В полиграфии обычно применяют модель **СМΥК** (Cyan, Magenta, Yellow, Key color).

В СМΥК используются четыре цвета, первые три в аббревиатуре названы по первой букве — бирюзовый, пурпурный, желтый. Эти цвета описывают отраженный от белой бумаги свет трех основных цветов модели RGB. При нулевом значении всех параметров модели получим белый цвет. При смешении всех трех цветов в максимальном значении должен получиться черный. Однако на практике при использовании реальных типографских красок зачастую вместо черного получается неопределенный темный цвет, поэтому (а также в целях


экономии цветного тонера) черный цвет добавлен в модель CMY в качестве четвертого базового под названием ключевого (*англ.* Key — ключевой).

В программах компьютерной графики часто используется модель HSB (или HSV). Считается, что эта модель ближе к человеческому восприятию цветов, чем модели RGB и CMYK. Координатами цвета в модели HSB являются цветовой тон (*англ.* Hue — оттенок), насыщенность (*англ.* Saturation) и яркость (*англ.* Brightness).


Оттенок H определяет цвет в спектре, который разделен на 360 оттенков — по цветовому кругу от 0 до 360 (оба значения представляют красный оттенок). Например, значение 60 соответствует желтому, 120 — зеленому, 240 — синему оттенкам. *Насыщенность S* определяет часть чистого цвета: чем S больше, тем «чище» цвет, поэтому этот параметр иногда называют *чистотой* цвета. Чем ближе этот параметр к нулю, тем ближе цвет к нейтральному серому. *Яркость V* иногда называют значением цвета, поэтому модель часто называют HSV (*англ.* Value — значение). При минимальной яркости любой цвет становится черным. Насыщенность и яркость изменяются в пределах от 0 до 100.

ГЕОМЕТРИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ ГРАФИЧЕСКИХ ОБЪЕКТОВ

ПЕРЕМЕЩЕНИЕ

Для перемещения объекта следует навести на него курсор мыши (в области границы объекта) так, чтобы курсор принял форму четырехнаправленной стрелки , и, нажав кнопку мыши, перетащить в требуемую позицию.


ИЗМЕНЕНИЕ РАЗМЕРА

Для изменения размеров графических объектов используются маркеры, ограничивающие их по углам и серединам сторон. Как правило, срединные маркеры позволяют изменить только один размер — горизонтальный либо вертикальный, а угловые — сразу оба. При подведении мыши к маркеру форма курсора меняется на двунаправленную стрелку  — в этот момент можно нажать левую кнопку мыши и, не отпуская ее, переместить маркер в нужном направлении.

ОБРЕЗКА

Некоторые графические объекты допускают обрезку. Для этого служит кнопка *Обрезка*. При ее нажатии по углам и серединам сторон рисунка появляются засечки, которые можно захватить мышью и перемещать, обрезая рисунок. По окончании обрезки надо щелкнуть мышью в стороне от рисунка, чтобы зафиксировать результат.

ПОВОРОТ

Для поворота рисунков, в том числе рисунков из коллекции клипов, а также фигур служат специальные круглые маркеры зеленого цвета, отображаемые в момент выделения рисунка. При наведении на него мыши курсор меняет вид указателя . В этот момент можно захватить маркер и вращать фигуру.

СТИЛЕВЫЕ ПРЕОБРАЗОВАНИЯ ГРАФИЧЕСКИХ ОБЪЕКТОВ

Графические редакторы содержат множество различных эффектов и стилей, которые можно применять к изображениям.

Прежде всего, можно настроить яркость, контрастность рисунка и даже перекрасить его в иные тона. При перекрашивании можно задать отображение рисунка в оттенках серого цвета или сепии, осветлить его для использования в качестве подложки текста или превратить в черно-белый, можно выбрать один из темных или светлых вариантов. Для возврата к исходным цветам следует выбрать вариант *Без перекраски*.

Кроме того, можно изменить форму рисунка. Например, используя *Стили рисунков*, можно придавать рисунку различные очертания — овальное, с размытыми краями, в перспективе и т. д.

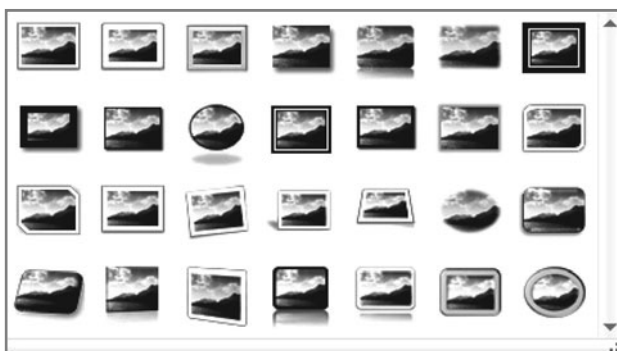


Рис. 2.76. Варианты стилей рисунков

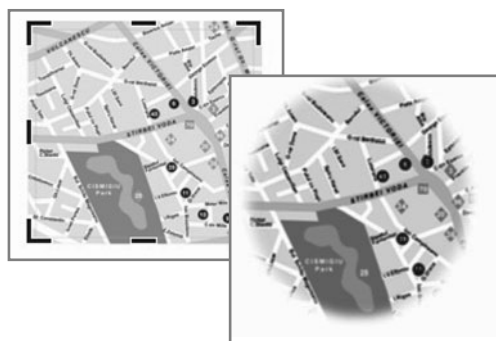


Рис. 2.77. Результат применения к рисунку стиля Овал с размытыми краями

Форматы графических файлов

Существует множество форматов файлов, предназначенных для сохранения графических изображений. Эти форматы определяют способы записи графической информации и форму ее сохранения (метод ее сжатия). Они отличаются качеством изображения, объемом памяти, необходимой для его сохранения, скоростью преобразования, возможностью поддерживать прозрачность, слои, разные цветовые модели и т. д.

Многие форматы используют для сохранения изображения **алгоритм сжатия с потерей качества**. Считается, что часть информации в изображении избыточна для человеческого восприятия и может быть отброшена без особого ущерба. Это позволяет значительно уменьшать объемы файлов. Однако зачастую алгоритмы сжатия приводят к необратимой потере информации, и тогда преобразованное изображение не может быть восстановлено в первоначальном качестве. **Алгоритмы сжатия без потерь** позволяют восстанавливать качество исходного изображения.

Форматы растровых графических файлов

- **JPEG** (от названия фирмы-разработчика **Joint Photographic Experts Group** — Объединенная группа экспертов по фотографии). Расширения файлов — **jpeg, jpg, jpe**.

Один из самых распространенных графических форматов файлов. Применяется для хранения фотографий и других сложных графических изображений. Использует для сжатия изображения эффективные алгоритмы с потерей качества (они преобразуют группы пикселей, расположенные по соседству по горизонтали и по вертикали, в один пиксель с усредненным цветом). При со-

хранении файла можно указать степень сжатия изображения, в зависимости от которой файл будет занимать разный объем памяти (и иметь разное качество).

Формат не поддерживает прозрачность изображения. Часто используется для предоставления графики в Интернете.

Целесообразно использовать этот формат для изображений с плавными переходами между цветами (в таком случае не столь заметна потеря качества).

- **BMP, Bitmap Picture** (от *англ.* BitMap — битовая карта). Расширение файлов — **bmp**.

Универсальный формат, который сохраняет изображение попиксельно, без сжатия. Поддерживает различную глубину цвета, вплоть до максимальной — 48 бит. При этом для глубины цвета меньше 16 бит используется полноцветная палитра глубиной 24 бита. Качество хранимых изображений очень высокое, но, соответственно, файлы имеют большой размер.

Поддерживается многими графическими редакторами, поэтому может быть использован для обмена с ними.

- **TIFF** (от *англ.* Tagged Image File Format — формат файлов изображений, снабженных тегами). Расширения файлов — **tif, tiff**.

Один из наиболее широко распространенных форматов, используемых при подготовке графиков для полиграфии; поддерживается всеми графическими редакторами.

TIFF-файл состоит из тегов — блоков данных, содержащих информацию о размерах изображения, его цветовой модели, размере палитры, сжатии данных и т. д. Всего определено несколько десятков таких блоков. Само изображение также содержится в отдельном теге. Поэтому формат легко поддается модернизации и расширению.

Включает алгоритмы без сжатия и сжатия без потерь (при сжатии используется поиск в изображениях повторяющихся узоров).

Фактически является стандартом для подготовки изображений в полиграфии, широко используется в системах цифровой картографии и географических информационных системах.

- **GIF** (от *англ.* Graphics Interchange Format — формат для обмена изображениями). Расширение файлов — **gif**.

Использует всего 256 цветов, поэтому файлы требуют небольшого объема памяти. Поддерживает сжатие данных без потери качества. Реализует очень плотное сжатие. Хорошо сжимает файлы, в которых много однородных заливок (логотипы, надписи, схемы). Позволяет сохранять в одном файле несколько последовательных изображений и информацию о времени их показа на экране, что при просмотре дает эффект мультипликации. Поддерживает прозрачность в изображениях.

Часто используется на веб-страницах для сохранения рисованных иллюстраций.

- **PNG** (от *англ.* Portable Network Graphics). Расширение файлов — **png**.

Сжатие растровых изображений (глубиной цвета в 16, 24 или 48 бит) производится без потерь. Создан как свободный формат для замены GIF. В отличие от него, поддерживает до 2^{48} цветов в изображении.

Предназначен прежде всего для использования в Интернете и редактирования графики.

- **RAW** (от *англ.* raw — сырой). Расширение файлов — **raw**. Содержит необработанные (или обработанные в минимальной степени) данные, что позволяет избежать потерь информации. Данные в RAW-файлах могут быть несжатыми и сжатыми — как без потерь, так и с потерями. Используется для работы с высококачественными фотографиями, получаемыми с цифровых фотоаппаратов.

Форматы векторных графических файлов

- **CDR** (от названия программы CorelDRAW). Расширение файлов — **cdr**. Создается программой CorelDRAW. Изображение может содержать как векторное, так и растровое изображение или текст, а также состоять из страниц.
- **EPS** (от *англ.* Encapsulated PostScript — инкапсулированный (заклученный в капсулу) PostScript). Расширение файлов — **eps**. Относительно универсальный векторный формат. Поддерживается большинством векторных редакторов и различными узкоспециализированными программами. Предназначен для обмена графическими данными между различными приложениями. Используется в профессиональной полиграфии, может содержать растровые и векторные изображения, а также их комбинации.
- **SVG** (от *англ.* Scalable Vector Graphics — масштабируемая векторная графика). Расширение — **svg**. Универсальный векторный формат, предназначенный для сохранения векторной и смешанной векторно-растровой графики, текста, анимации. Обрабатывается практически всеми векторными графическими редакторами. Широко применяется в инженерной графике и при разработке веб-сайтов.
- **WMF** (от *англ.* Windows Metafile Format — формат метафайлов Windows). Расширения файлов — **wmf, emf**. Универсальный векторный формат, поддерживаемый почти всеми векторными редакторами, работающими в MS Windows. Используется для хранения коллекции графических изображений MS Clip Gallery. Для сложных рисунков обеспечивает не очень высокое качество. Поддерживает весьма ограниченное число эффектов.

2.4. ПОИСК ИНФОРМАЦИИ

2.4.1. Компьютерные энциклопедии и справочники; информация в компьютерных сетях, некомпьютерных источниках информации. Компьютерные и некомпьютерные каталоги, поисковые машины, формулирование запросов

Поиск информации заключается в ее нахождении в имеющихся информационных фондах.

Учитывая огромные объемы накопленной человечеством информации, для ее поиска требуются специальные средства. С этой целью создаются каталоги

библиотечных, музейных и других фондов, издаются и постоянно обновляются различные энциклопедии, справочники и др.

Наиболее емким и доступным хранилищем информации сегодня является Интернет. В его пространстве существуют миллионы веб-сайтов, и это число постоянно увеличивается. Для поиска необходимой пользователю информации в Интернете существуют различные поисковые средства. Реальными носителями информации о ресурсах, которыми располагает Всемирная сеть, являются поисковые машины (автоматические индексы) и каталоги.

Каталоги составляются людьми — редакторами, просматривающими каждый новый сайт до его включения в индекс, или самими составителями описаний. Качество каталога при этом выше, чем у поисковых машин, но люди не могут успеть за темпами расширения и изменения Интернета.

Каталоги обычно организованы в соответствии с предметной классификацией (по *рубрикам*, или *категориям*) и содержат сведения о веб-страницах (так называемый «видимый» Интернет). Каталог позволяет найти нужный сайт, сужая область поиска по теме и региону.

В русскоязычном каталоге Яндекс (*Yandex*, yaca.yandex.ru) описаны десятки тысяч ресурсов. Ежемесячно с ним работают более двух миллионов пользователей. Автоматическая система регулярно проверяет ссылки на предмет доступности и снимает с публикации недействительные.

Любой каталог предлагает воспользоваться иерархическим деревом при поиске информации. Сначала необходимо выбрать общую тематику, на следующих этапах конкретизировать выбор, следуя подсказкам каталога. В результате будет получен список сайтов, содержащих информацию, соответствующую запросу.

Каталог Яндекс является также и рейтингом описанных в нем ресурсов: сайты в рубриках расположены по убыванию их тематического индекса цитирования — учитывается количество ссылок на сайт с других сайтов, придавая этим ссылкам разный «вес» (то есть значимость) в зависимости от авторитетности ссылающегося сайта.

Однако чаще всего для поиска нужной информации используют специальные **поисковые веб-серверы**. С помощью специальной программы такие серверы ежедневно просматривают интернет-пространство с целью выявления новых веб-серверов, запоминают эту информацию, удаляют информацию о серверах, прекративших существование, собирают информацию о новых веб-страницах и изменениях существующих, удаляют информацию об исчезнувших веб-страницах. Вся найденная информация сортируется и сохраняется в специализированной *базе данных* на поисковом сервере.

Другая программа этого же сервера работает с запросами, поступающими от пользователей сети Интернет. Она производит поиск в базе данных и возвращает пользователю список ссылок на те веб-страницы, которые соответствуют запросу.

Такие роботизированные системы называют **поисковыми машинами** (*search engines*). Как правило, они состоят из трех частей:

- 1) специальная **программа-робот** (ее еще называют *паук* — *spider*), которая постоянно обходит Сеть, находит новые веб-страницы, копирует их содержание в базу данных, сохраняя полные тексты веб-страниц или их компоненты (заголовки, гипертекстовые ссылки, комментарии, названия программ и т. п.), проходит по всем имеющимся ссылкам к другим страницам. Процесс, в ходе которого робот обнаруживает новые и обновленные страницы для добавления в индекс,

называется *сканированием*. Робот осуществляет *индексирование* (или *индексацию*, от *лат.* index — признак, указатель) веб-страниц — анализ содержимого сайта и добавление сведений о нем в базу данных. В сведения о сайте чаще всего входят ключевые слова, ссылки, могут индексироваться изображения и т. д. **Ключевым словом** (keyword) или словосочетанием называется слово, которое отражает содержание документа. Большинство роботов индексируют все слова веб-страницы, за исключением служебных. Например, робот Googlebot поисковой системы Google (google.com) составляет полный индекс всех найденных слов для каждой сканируемой страницы, а также отмечает, в какой части страницы они находятся;

2) база данных/хранилище данных (собственно **индекс**) — все страницы, которые были скачаны роботом и впоследствии обработаны встроенными средствами. База данных содержит проиндексированную информацию всех веб-страниц и их адреса;

3) поисковый механизм, который взаимодействует между пользователем и базой данных. Когда пользователь вводит поисковый запрос, система находит в индексе соответствующие веб-страницы и выдает наиболее релевантные результаты. *Релевантность* — соответствие поискового запроса и представления веб-страниц в базе данных поисковой системы (индексе).

Важными показателями качества поисковой машины являются *объем базы данных* (количество документов), *скорость* обхода Сети (с этим связана скорость обновления информации в базе данных), *алгоритм индексации* (только по ключевым словам веб-страницы или по всему тексту, с учетом правил морфологии языка или без него, с поиском по тэгам, html-заголовкам, ссылкам, подписям к изображениям и др.), а также *дополнительные возможности* (расширенный поиск, поиск похожих документов, ограничение области поиска), удобный пользовательский *интерфейс* и *справочная система*.

Мощная поисковая машина обходит сеть за несколько дней. При этом составляется актуальный и довольно подробный индекс. При каждом новом цикле обхода индекс обновляется и старые недействительные адреса удаляются. Однако автоматизированный подход приводит к тому, что в индекс могут попасть дубликаты (один и тот же документ на разных сайтах, в разных кодировках). Кроме того, часть Интернета для поисковых машин закрыта — например, информация, доступ к которой осуществляется по паролю, базы данных, доступ в которые осуществляется по запросу из формы (а не по ссылке), и др.

Порядок обхода страниц, частота визитов, защита от заикливания, а также критерии выделения значимой информации определяются *поисковыми алгоритмами*.

Для формулировки запроса к поисковой системе можно использовать *язык поисковых запросов*. Он включает логические операторы, учет морфологии языка, регистра слов и т. д. Разные поисковые системы используют различные символы для обозначения логических операторов. Чаще всего используется знак & (для обозначения операции логического умножения — ставится между словами, которые должны встречаться вместе) и знак | (для логического сложения, ставится между словами, если нужно найти одно из них).

Во многих поисковых системах имеется возможность решать сложные поисковые задачи, не пользуясь языком запросов. Для этого существуют формы расширенного поиска, где сложные поисковые условия задаются в простой и наглядной форме.

Пример 1

Расположить приведенные ниже запросы к поисковой системе в порядке возрастания количества страниц, которые будут найдены поисковым сервером по этим запросам.

- А) температура & осадки & облачность
- Б) температура | осадки | облачность
- В) температура & осадки
- Г) температура | осадки

Решение.

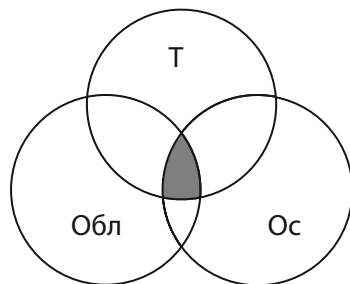
Способ 1. Результатом логической операции И (&) будет «истина», если оба слова встретятся на странице. Результатом логической операции ИЛИ (|) будет «истина», если хотя бы одно слово из запроса встретится на странице.

Наименьшее количество результатов будет найдено в запросе с самыми строгими ограничениями (А). Следующий по количеству результатов — запрос (В), поскольку для него к результатам запроса (А) добавятся страницы, содержащие слова «температура» и «осадки», но не содержащие слово «облачность». Запрос (Г) добавит к этим результатам еще и те страницы, где встречается только одно из слов «температура» и «осадки». Наконец, запрос (Б) припишет к предыдущим результатам те страницы, где встречается слово «облачность». Результат — порядок АВГБ.

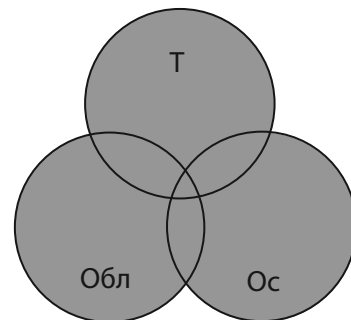
Способ 2 (графический). Используем для отображения решения диаграммы Венна. Они удобны для отображения множеств и обеспечивают наглядность задачи. В теории множеств аналогом логической операции И является пересечение множеств, логической операции ИЛИ — их объединение.

Обозначим множество страниц, содержащих слово «температура», через *T*, множество страниц, содержащих слово «осадки», — через *Ос*, множество страниц, содержащих слово «облачность», — через *Обл*. Тогда диаграммы Венна будут иметь вид:

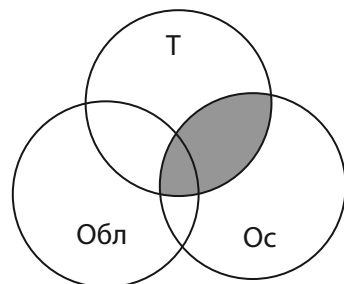
А)



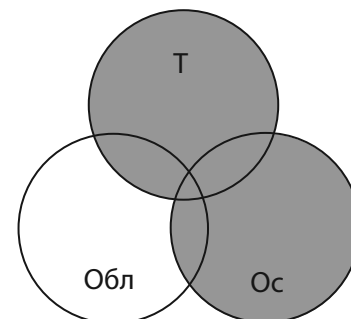
Б)



В)



Г)



Минимальное количество результатов будет найдено по запросу (А) — он содержит больше всего слов, соединенных оператором И; по этому запросу будут найдены страницы, содержащие все три слова одновременно. Максимальное количество результатов даст запрос (Б) — в нем больше всего слов, связанных операцией ИЛИ (будут найдены все страницы, содержащие хотя бы одно из этих слов). Аналогично по запросу (Г) будет найдено больше страниц, чем по запросу (В).

Ответ: АВГБ.

2.5. ПРОЕКТИРОВАНИЕ И МОДЕЛИРОВАНИЕ

2.5.1. Чертежи. Двумерная графика. Использование стандартных графических объектов и конструирование графических объектов: выделение, объединение, геометрические преобразования фрагментов и компонентов

В процессе исследования формальных моделей часто производится их *визуализация* — представление в графическом виде, удобном для восприятия и обработки.

Например, для визуализации алгоритмов используются *блок-схемы*. Чтобы отобразить пространственные соотношения между объектами, создают *чертежи*. Модели электрических цепей отображают с помощью *электрических схем*. Для отображения динамики процесса применяют *анимацию*, строят *графики* изменения величин и т. д. Широкое распространение получили компьютерные *интерактивные визуальные модели*. В них исследователь может менять начальные условия и параметры протекания процессов и наблюдать изменения в поведении модели.

Системы автоматизированного проектирования (САПР, *англ.* CAD — Computer-Aided Design) — это программы для поддержания работы конструкторов и технологов. Такие системы связаны с разработкой чертежей, схем, диаграмм, графическим моделированием и конструированием.

Эти программы позволяют организовать готовые наборы (библиотеки) стандартных элементов чертежей и их многократно использовать. Такие элементы обладают заранее определенными свойствами. Для конструирования новых объектов применяют различные операции к уже имеющимся элементам.

Векторные графические редакторы, предназначенные именно для создания чертежей, называются **системами компьютерного черчения**. Наряду с другими, эти программы также являются инструментами автоматического проектирования.

Средства автоматического проектирования предъявляют высокие требования к технической части систем обработки данных. Для них обычно требуется взаимодействие с графическими системами и базами данных.

Примеры таких программ: AutoCAD, ArchiCAD, P-Cad, Microsoft Visio, EPLAN, Pro/Engineer.

Действия над графическими объектами

Рассмотрим типовые операции над графическими объектами и их фрагментами на примере графических объектов MS Visio. Заметим, что эти операции имеют похожую реализацию и в других системах автоматического проектирования.

ПОРЯДОК РАСПОЛОЖЕНИЯ ФИГУР

Зачастую для выполнения операций над объектами важно их взаимное расположение на листе чертежа. Это расположение объектов относительно друг друга обычно определяется *порядком* появления их на листе. Объекты (фигуры), перенесенные на лист первыми, располагаются позади, а те, что были размещены позднее, — впереди. Порядок можно изменять — как правило, с помощью команд контекстного меню объектов.

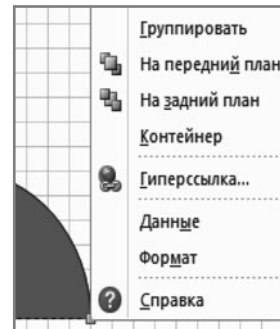


Рис. 2.78. Команды упорядочения в контекстном меню объекта

ПОЗИЦИОНИРОВАНИЕ ОБЪЕКТОВ И НАПРАВЛЯЮЩИЕ

Для **позиционирования** (необходимого расположения) объектов удобно использовать **сетку**, линии которой видны только на экране и не отображаются при распечатке.

Кроме того, при перетаскивании объектов их местоположение размечается пунктирными вертикальными и горизонтальными линиями для выравнивания объектов друг относительно друга. Некоторые вспомогательные (непечатаемые) горизонтали и вертикали на листе — **направляющие** — можно временно зафиксировать. Для этого достаточно оттянуть такие линии мышью на страницу с желаемых позиций линейки. При приближении объектов к направляющим объекты «приклеиваются» к ним, облегчая позиционирование.

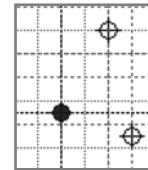


Рис. 2.79. Направляющие точки

Также можно временно расположить на листе **направляющие точки** (рис. 2.79) — перекрестия направляющих линий (оттягивая их от перекрестия линейек в левом верхнем углу).

Удаление направляющих линий и точек, как и других элементов чертежа, обычно осуществляется клавишей Delete.

ВЫРАВНИВАНИЕ И РАСПРЕДЕЛЕНИЕ ФИГУР

Для **выравнивания** нескольких объектов их необходимо предварительно выделить и выбрать одну из команд выравнивания. Как правило, команды выравнивания содержатся в контекстном меню выделенных объектов. Обычно имеются возможности *горизонтального* выравнивания (по левому или правому краю либо по центру) и *вертикального* (по верхнему или нижнему краю либо по середине).

Выравнивание происходит по первой выделенной фигуре. Удобно то, что будущий результат выравнивания отображается на чертеже еще при наведении мыши на команду.

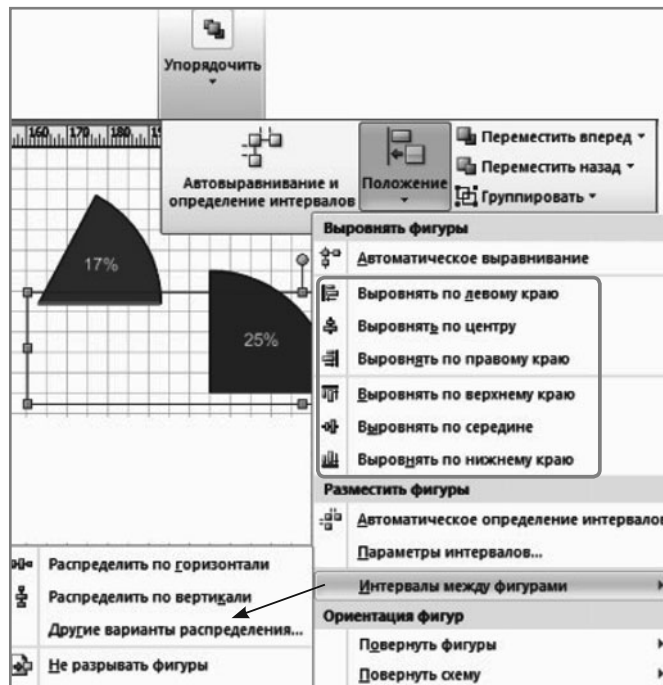


Рис. 2.80. Команды упорядочения и выравнивания объектов

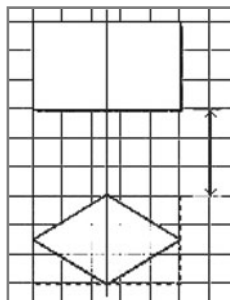


Рис. 2.81. Осевые линии

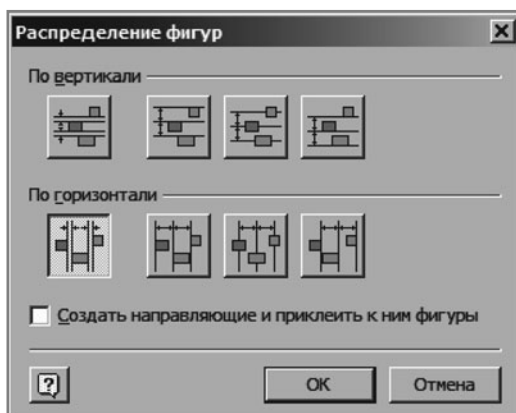



Рис. 2.82. Выравнивание и распределение фигур

Лучше строить чертеж, сразу учитывая будущее выравнивание его элементов. Для этого в программе часто предусмотрен автоматический вывод временных осевых линий в момент перетаскивания фигуры на листе. На рис. 2.80 показаны такие временные линии, возникшие при продвижении ромба возле прямоугольника в момент их выравнивания по вертикали. Если обращать внимание на такие подсказки программы, создание чертежей будет значительно легче.

При необходимости можно равномерно **распределить** объекты между собой — через одинаковые промежутки (выравнивания объектов для этого не требуется). Как для вертикального, так и горизонтального распределения можно задать одинаковые промежутки между фигурами либо равномерно расположить их верхние (срединные, нижние) или левые (срединные, правые) края.

СОЕДИНЕНИЕ ОБЪЕКТОВ

Многие объекты имеют специальные соединительные маркеры () , с помо-

пью которых можно **соединять (склеивать)** эти объекты между собой. При состыковке таких маркеров они приобретают красное обрамление, что говорит о склеивании фигур между собой. Та фигура, к которой была приклеена другая, будет при перемещении «тянуть» вторую за собой. Перемещение же второй фигуры оторвет ее от первой.

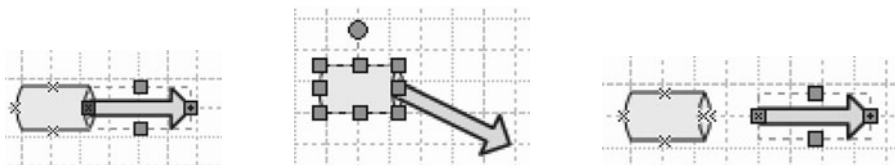


Рис. 2.83. Соединение фигур

К таким маркерам удобно привязывать специальные линии, соединяющие двумерные фигуры, — **коннекторы** (, *соединительные линии*).

Пристыковать линию-коннектор можно не только к определенным соединительным маркерам, но и ко всей фигуре в целом. При этом красная рамка — так называемое *динамическое склеивание* — возникает вокруг всей фигуры. При перемещении таких объектов линия-коннектор будет привязываться к тем точкам, которые будут ближе друг к другу. Можно сказать, что коннекторы обладают «интеллектом» — они не только сохраняют связь между объектами при их перемещении, но и ищут кратчайшие пути соединения этих объектов.

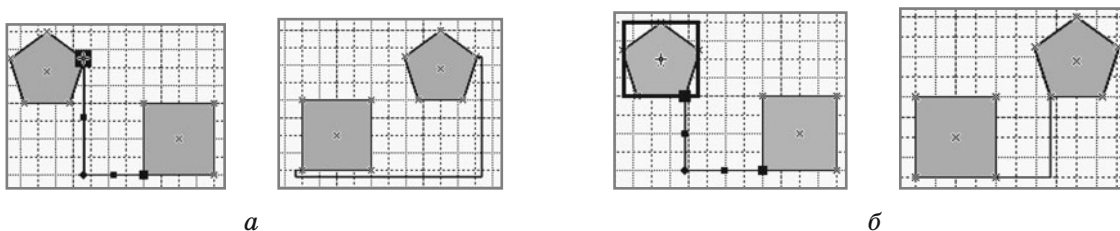


Рис. 2.84. Фиксированное (а) и динамическое (б) склеивание фигур

ГРУППИРОВКА ФИГУР

Если объекты чертежа используются вместе, их желательно **сгруппировать**. Для этого следует выделить их и выбрать команду группировки — например, в контекстном меню объектов. После этого можно изменять размеры группы и перемещать ее как отдельный объект.

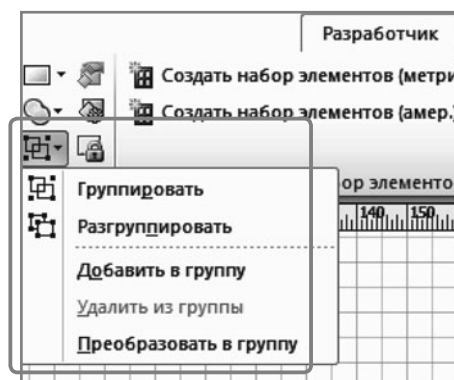
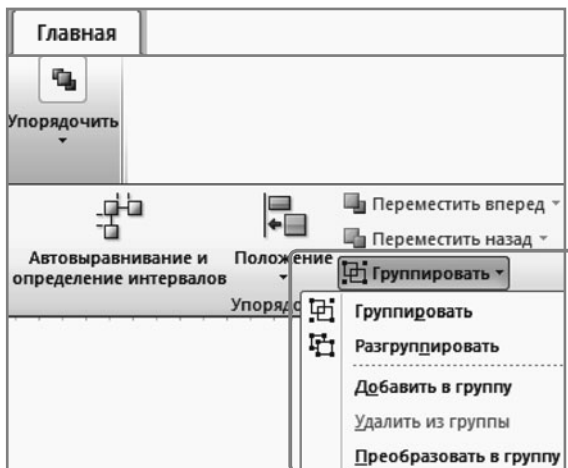


Рис. 2.85. Меню группировки фигур

2.5.2. Диаграммы, планы, карты

Графический способ представления информации позволяет в наглядной форме отобразить многие зависимости между данными, оценить величины, помогает решать задачи, принимать решения.

К графическим средствам можно отнести карты и планы, схемы и чертежи, различного вида графики и диаграммы и др.

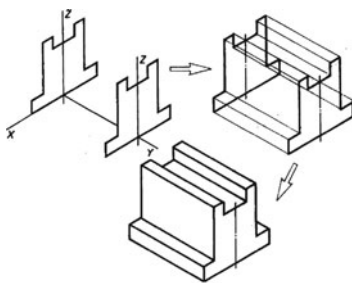


Рис. 2.86. Чертеж детали

Изображение земной поверхности на плоскости в условных знаках, имеющее масштаб 1:10000 или крупнее, обычно называют **планом** (зачастую это топографические планы). Так же называется вид сверху или горизонтальный разрез какого-либо сооружения или предмета. Как правило, план выполняют в крупном масштабе.

Планы зданий, городов, местности помогают ориентироваться и легче воспринимать весь объект в целом.

Электронные **карты** предназначены для визуализации чертежей неба, земли, моря. Существуют топографические, геологические, метеорологические, морские карты, карты полезных ископаемых, звездного неба, автомобильных дорог и проч.

Для отображения информации на картах используют систему условных обозначений, которая включает специальные символы, цветовые поля, различные шрифты и т. п. Условные обозначения с их разъяснением располагают рядом с основным изображением карты и называют **легендой**. Там же указывают и **масштаб** карты.

Электронные карты могут быть представлены в виде отдельных графических файлов или лежать в основе работы специальных приложений (программ). Такие программы могут осуществлять масштабирование, поиск по карте, измерение расстояний, прокладку маршрутов и многое другое. Поисково-информационные картографические сервисы в Интернете позволяют осуществлять эти и другие операции в режиме онлайн.

Зачастую необходимо наглядно отобразить зависимость какой-либо величины от другой. С этой целью используют **графики**.

Графики — это одна из разновидностей диаграмм. В общем случае **диаграмма** — это геометрическое представление данных, показывающее их соотношение и выполненное в символьном виде одним из способов визуализации. Диаграммы в основном состоят из геометрических объектов (точек, линий, фигур различной формы и цвета) и вспомогательных элементов (осей координат, условных обозначений, заголовков и т. п.). Существуют тысячи разновидностей диаграмм. В большинстве случаев каждый вид диаграмм можно представить в **двумерном** (плоскостном) и **трехмерном** (объемном, пространственном) виде.

Чертежи — контурные изображения различных объектов. К чертежам, используемым в конструкторской деятельности, предъявляются определенные требования по их оформлению. В них, как правило, используются проекции объектов, созданные по правилам начертательной геометрии.

Отношение размеров изображения объекта на чертеже к их натуральной величине называется **масштабом**.

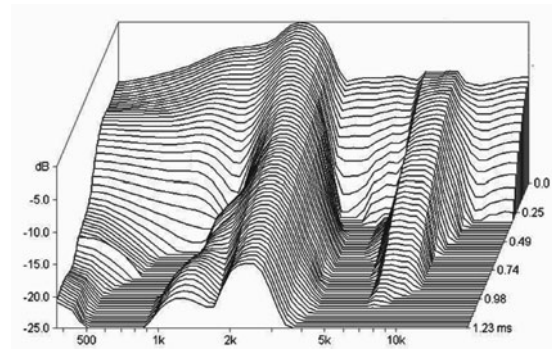


Рис. 2.87. Двумерные и трехмерный графики

Диаграммы позволяют сопоставлять величины, наглядно отображать их изменения во времени, зависимости. Множество разновидностей диаграмм позволяют визуально подчеркнуть какие-либо конкретные особенности отображаемых данных.

Так, графики используют, если надо отобразить математическую зависимость величин друг от друга. Очень часто их применяют, чтобы показать тенденцию изменения данных во времени (или сравнить тенденции изменений сразу нескольких рядов данных). Обычно графики имеют равномерную шкалу.

Столбчатые (столбиковые) и линейчатые (ленточные, полосковые) диаграммы представляют величины данных в виде площадей прямоугольников — соответственно вертикальных или горизонтальных. Столбчатые диаграммы называют **гистограммами**. Обычно ширина всех их прямоугольников одинакова, а значение величин отображается их высотой. Ленточные диаграммы отличаются лишь горизонтальным расположением прямоугольников данных. Обе разновидности диаграмм удобны для демонстрации изменений величины. Часто они используются для сравнения между собой нескольких рядов данных. Иногда вместо прямоугольников в таких диаграммах используют конусы, цилиндры и другие геометрические объекты.

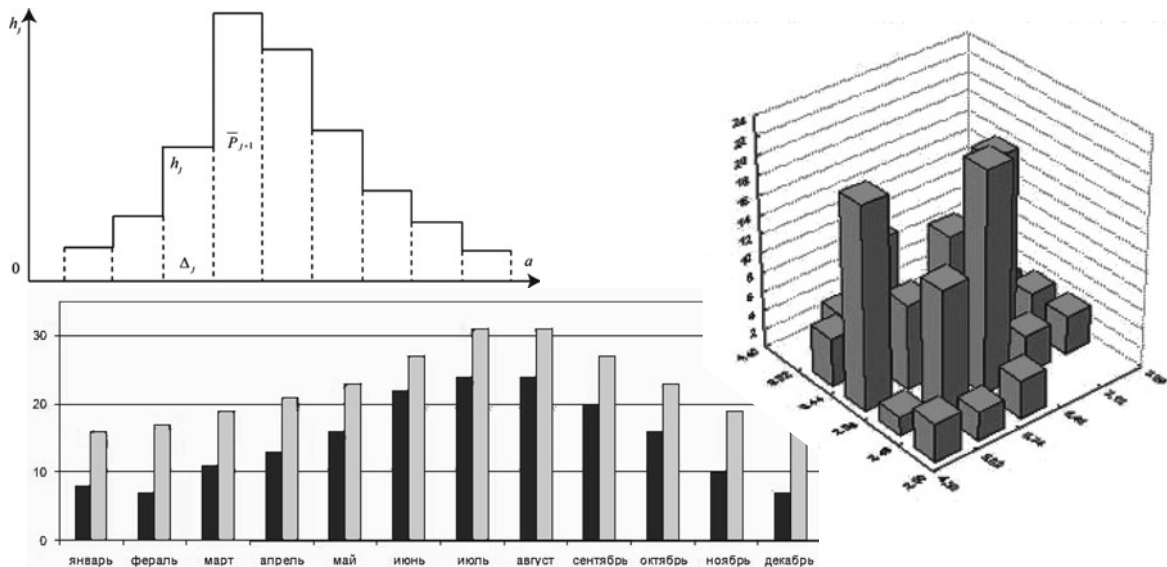


Рис. 2.88. Различные виды гистограмм

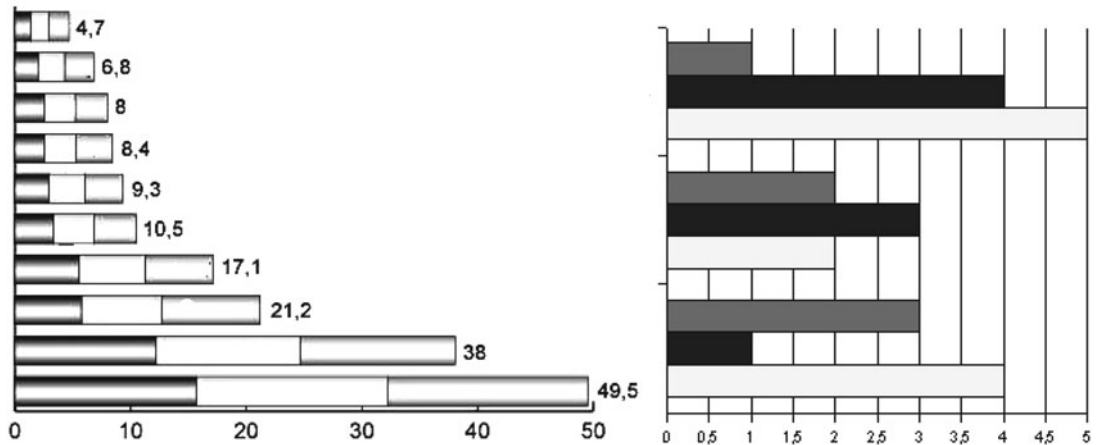


Рис. 2.89. Ленточные диаграммы

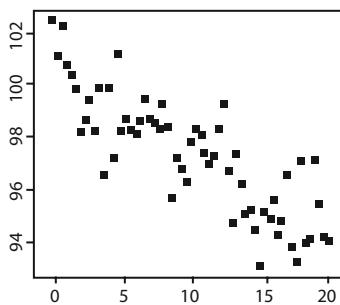
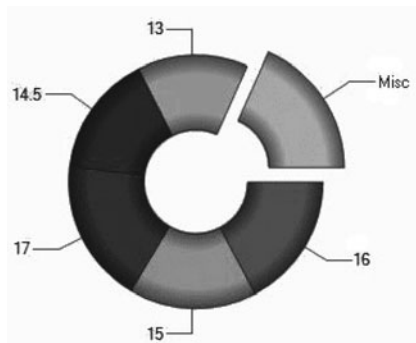
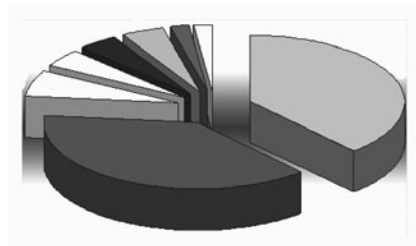


Рис. 2.90. Круговая, кольцевая и точечная диаграммы

Также широко используются **круговые диаграммы**. Круг представляет всю совокупность данных, а каждый его сектор — относительную величину каждого значения. Эти диаграммы удобны, когда нужно показать долю каждой величины в общей совокупности, а особенно — подчеркнуть вклад какой-либо конкретной составляющей.

Кольцевые диаграммы содержат несколько колец — каждое кольцо представляет один ряд данных.

Точечная диаграмма (диаграмма рассеяния) изображает значения двух величин в виде точек на плоскости. Обычно значения независимого параметра откладывают по горизонтальной оси, а значения зависимого — по вертикальной. Точечные диаграммы используют для демонстрации наличия или отсутствия связи между двумя величинами.

Изображение системы, составных частей какого-либо объекта или процесса, их взаимосвязи называется **схемой**. Как правило, такое изображение представляет объект в самых общих чертах. Например, схема двигателя может упрощенно представлять принцип его функционирования, схема химической реакции дает общее представление о природе

участвующих в ней элементов и соединений. Блок-схема алгоритма помогает изложить его основные этапы.

Схематическое представление данных также можно реализовать в виде диаграмм. Например, **диаграммы Вена** схематично изображают в виде пересекающихся кругов все возможные отношения нескольких множеств данных. Каждый круг изображает совокупность элементов одного из множеств. Если какие-либо элементы принадлежат сразу нескольким множествам, им соответствует пересечение кругов, представляющих эти множества. Если на схеме указываются не все возможные пересечения подмножеств, то такое отображение называется **кругами Эйлера**.

Граф — это схема, отображающая объекты в виде точек, а связи между ними в виде линий, соединяющих эти точки. Точки называются *вершинами* (или *узлами*) графа, соединяющие линии — *дугами* (или *ребрами*) графа. Последовательность дуг и соединенных ими промежуточных вершин — *путь* между вершинами графа. Если у отрезков пути заданы направления (в виде стрелок), такой граф называют *ориентированным*, или *направленным*.

С помощью графов отображают маршруты движения пассажиров, перевозки грузов, транспортировки энергетических ресурсов, компьютерные сети. Графы применяют при составлении расписаний транспорта, занятий, соревнований, при анализе и проектировании систем электро-, водо-, газо-, теплоснабжения, для распределения радиочастот сотовой связи с целью достижения максимальной пропускной способности каналов. Графы помогают определить маршруты следования информации в компьютерных сетях (организовать маршрутизацию). Графами являются схемы метрополитенов, структурные схемы молекул, сетевые графики строительства: в них вершинами являются строительные работы, а дугами — связи между ними. Блок-схема алгоритма — это тоже граф, вершинами которого служат функциональные блоки, а дуги (линии) определяют очередность выполнения блоков.

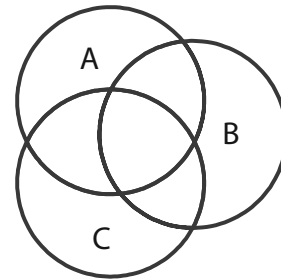


Рис. 2.91. Диаграмма Вена для трех множеств

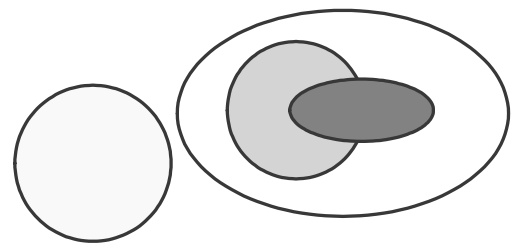


Рис. 2.92. Круги Эйлера для четырех множеств

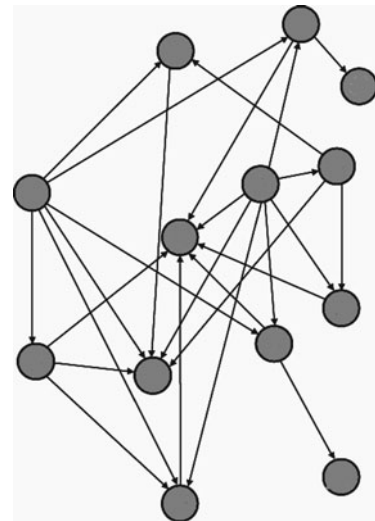


Рис. 2.93. Граф

Графы очень часто используются при решении логических и иных задач.

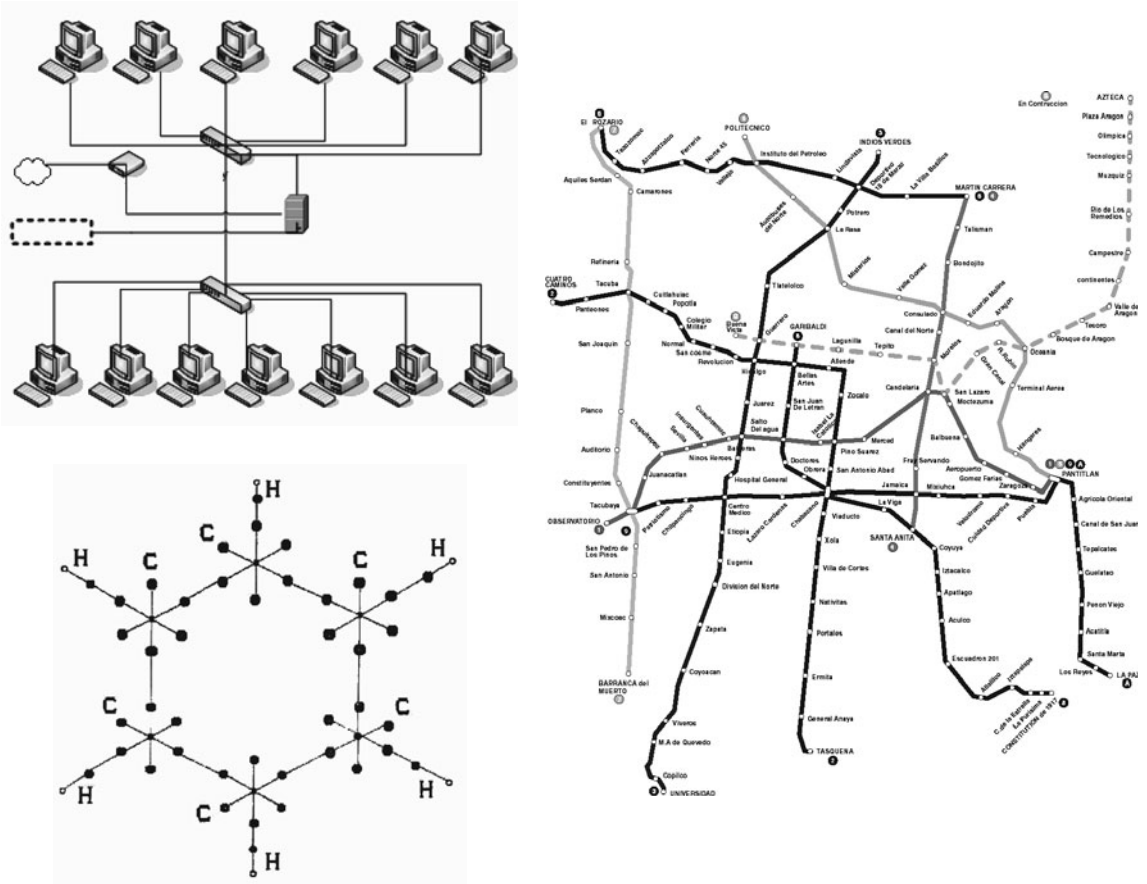


Рис. 2.94. Представление в виде графа компьютерной сети, структуры молекулы, схемы метрополитена

Иерархию (подчиненность) объектов отображают в графическом виде с помощью деревьев. **Дерево** — граф, в котором две любые вершины соединены только одним путем и нет циклов (нельзя из вершины пройти по нескольким различным ребрам и вернуться в ту же вершину).

Название этой схемы связано с тем, что она напоминает перевернутое дерево. Самую верхнюю вершину дерева (не имеющую родительской для нее вершины)

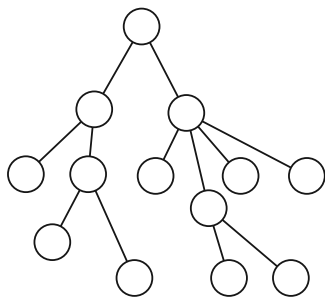


Рис. 2.95. Дерево

называют *корнем дерева*. Линии, связывающие элементы дерева, называются *ветвями*, сами элементы называются *узлами*. Узлы без потомков называются *конечными узлами* или *листьями*. Структуры деревьев похожи на генеалогические деревья, поэтому для отношений между узлами дерева используют названия «родители» и «потомки» (родительские и дочерние узлы).

Многие структуры могут быть представлены в виде деревьев. Например, структура файловых папок в операционной системе, поисковые каталоги в Интернете.

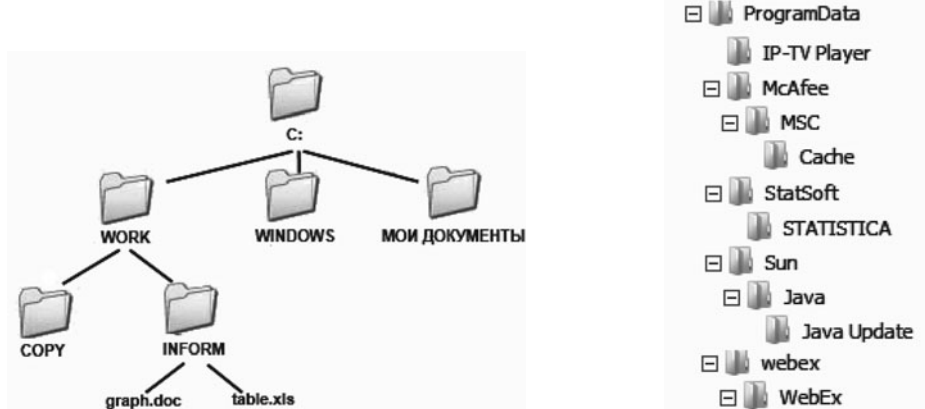


Рис. 2.96. Отображение структуры папок

Деревья — очень удобный инструмент представления информации. Благодаря тому что при обходе дерева невозможны циклы (повторы) в пути, эту форму используют для организации данных в алгоритмах.

В виде древовидного изображения реализуются и **интеллект-карты** — схемы, отображающие процесс мышления человека. Эти схемы имеют множество названий — их называют также *картами разума*, *картами памяти*, *когнитивными** *картами*, *mind map* (англ. карты ума), *ассоциативными картами*, *диаграммами связей*. Они помогают показать взаимоотношения некоторой идеи или понятия и связанной с ней ассоциативно информацией.

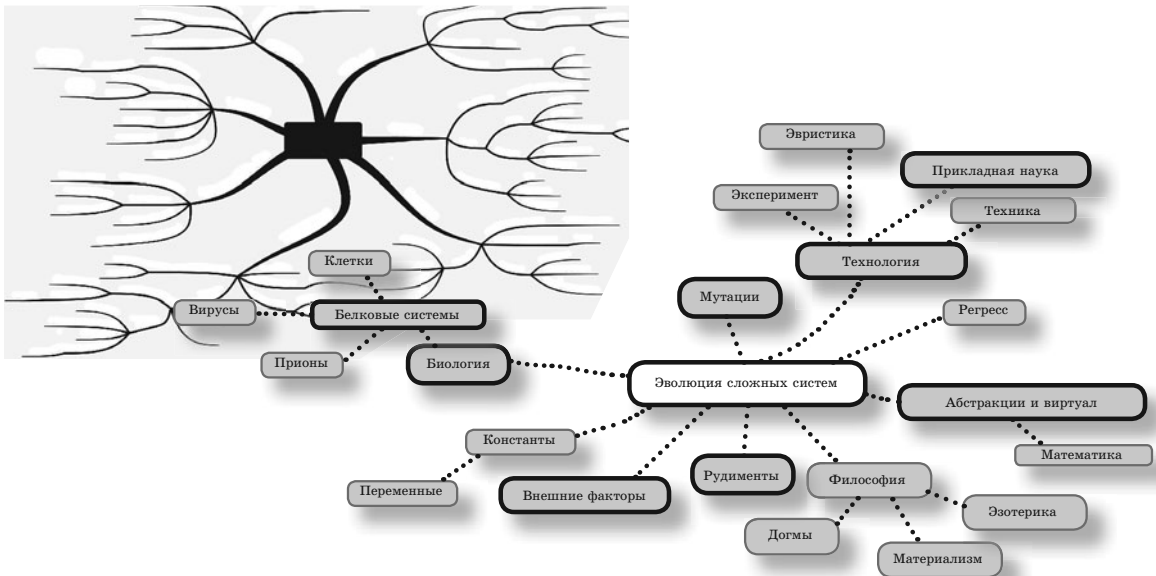


Рис. 2.97. Примеры интеллект-карт

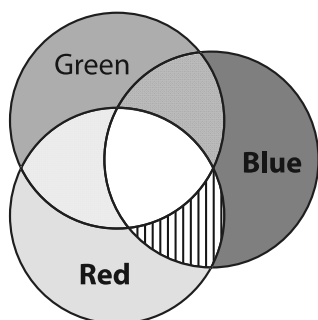
* *Когнитивный* — связанный с познанием.

Для этого в центре схемы размещают ключевое понятие, а радиально вокруг него — в произвольном порядке связанные с ним по смыслу иные понятия — единичные или сгруппированные в виде ветвей. Такой способ записи позволяет диаграмме неограниченно расти и дополняться.

Используются интеллект-карты для представления идей, их классификации, для наглядности в обучении, для решения задач, принятия решений, способствуют зарождению новых идей.

Графическое представление данных значительно облегчает решение многих задач. Например, диаграммы Венна и круги Эйлера помогают решить задачи, где требуется определить или сравнить множества значений. Для этого надо каждое множество изобразить в виде соответствующего круга. Если множества имеют общие элементы, круги должны пересекаться, в ином случае — нет.

► Пример 1

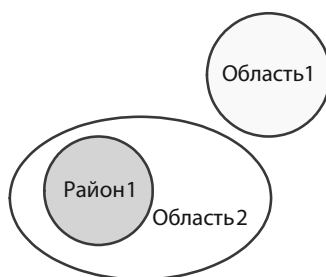


Изображение было создано с использованием трех основных цветов модели RGB. Отобразить с помощью диаграмм Венна цветовой результат наложения элементов рисунка друг на друга.

Решение. Пусть каждый из трех кругов отображает множество элементов рисунка одного из основных цветов RGB — красного, зеленого и синего. Тогда пересечение этих кругов показывает элементы рисунка, окрашенные в результате смешения красного и зеленого (= желтый), зеленого и синего (= голубой), синего и красного (= фиолетовый) цветов, а также их всех трех вместе (= белый).

► Пример 2

Отобразить с помощью кругов Эйлера пересечение трех множеств жителей — области *Область1*, области *Область2* и района второй области *Район1*.

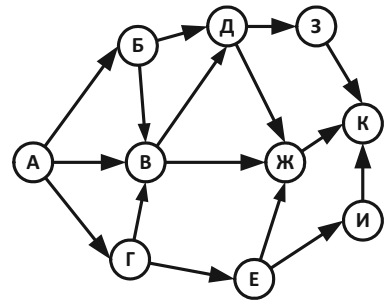


Решение. Одно из множеств (жителей района) является подмножеством другого (жителей *Области2*) — следовательно, один из кругов будет полностью лежать внутри другого. Еще один круг не будет пересекаться с первыми двумя, поскольку множество жителей разных областей не имеют общих элементов.

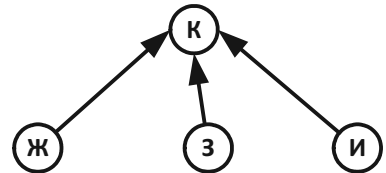
Для решения задач, связанных с маршрутами, используют представление данных в виде графов. Например, если требуется отыскать маршруты между некоторыми пунктами, удобно отобразить имеющиеся сообщения между пунктами в виде графа. Затем в соответствии с направлением отрезков рассчитать пути передвижения. Для этого этапа можно использовать представление маршрутов в виде дерева. При этом корнем дерева надо выбрать конечный или начальный пункт (в зависимости от задачи) и присоединять к нему в виде вершин все доступные пункты. Затем те же операции проделать для всех последующих или только подходящих по условию вершин дерева. По построенному дереву можно будет рассчитать количество путей сообщения или выписать конкретный необходимый маршрут.

Пример 3

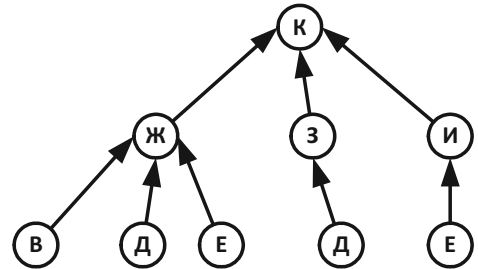
На рисунке изображена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?



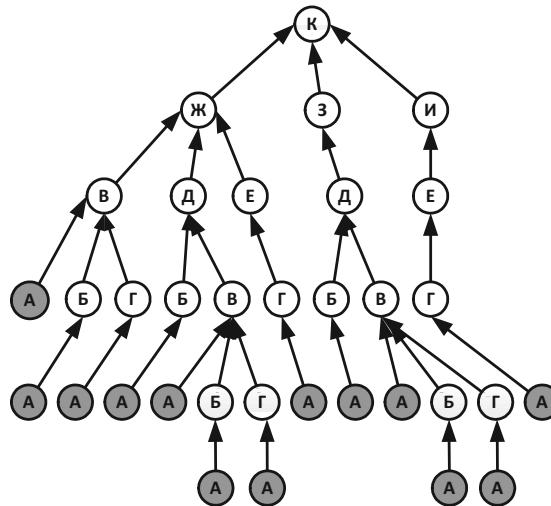
Решение. Способ 1. Граф всех возможных передвижений уже задан. Построим дерево решений задачи — всех маршрутов из А в К. Для этого изобразим город К как корень дерева. Соединим его со всеми городами, из которых можно попасть в К, — Ж, З, И.



К каждому из листьев дерева — Ж, З, И — присоединим вершины, отвечающие городам, откуда можно добраться до городов Ж, З, И. Для вершины Ж — В, Д, Е. Для вершины З — Д. Для вершины И — Е.



Так продолжим до тех пор, пока в каждой ветви не будет достигнута вершина начала маршрута А.



Подсчитаем количество листьев А в построенном дереве — это и будет количество путей от листьев до корня дерева (т. е. от города А до К). Таких листьев 13.

Способ 2. Эту задачу можно решить и без использования дерева, но с большим количеством подсчетов.

Обозначим через N_M количество различных путей из А в М, где М — любой из заданных городов. Начнем подсчет количества путей в А с конца маршрута — от города К. В город К можно приехать из городов Ж, З, И. Следовательно, количество путей в А будет равно сумме путей в каждый из этих трех городов:

$$x = N_{\text{Ж}} + N_{\text{З}} + N_{\text{И}}$$

Аналогично подсчитаем количество путей в каждый из трех городов Ж, З, И:

$$N_{\text{Ж}} = N_{\text{В}} + N_{\text{Д}} + N_{\text{Е}}$$

$$N_{\text{З}} = N_{\text{Д}}$$

$$N_{\text{И}} = N_{\text{Е}}$$

Продолжим подсчеты до тех пор, пока не будет достигнут город А в каждом из маршрутов.

$$N_{\text{В}} = N_{\text{А}} + N_{\text{Б}} + N_{\text{Г}}$$

$$N_{\text{Д}} = N_{\text{Б}} + N_{\text{В}}$$

$$N_{\text{Е}} = N_{\text{Г}}$$

Учтем, что $N_{\text{А}} = 1$:

$$N_{\text{В}} = N_{\text{А}} = 1$$

$$N_{\text{Г}} = N_{\text{А}} = 1$$

Тогда предыдущие выражения примут вид:

$$N_{\text{В}} = 1 + 1 + 1 = 3$$

$$N_{\text{Д}} = 1 + 3 = 4$$

$$N_{\text{Е}} = 1$$

Подставим найденные величины в начальные выражения:

$$N_{\text{Ж}} = 3 + 4 + 1 = 8$$

$$N_{\text{З}} = 4$$

$$N_{\text{И}} = 1$$

$$x = N_{\text{Ж}} + N_{\text{З}} + N_{\text{И}} = 8 + 4 + 1 = 13$$

Ответ: 13.

2.5.3. Простейшие управляемые компьютерные модели

Информационную модель, реализованную на компьютере, называют компьютерной моделью.

Компьютерные модели позволяют проводить моделирование многократно, повысить точность, достоверность и информативность результатов.

Основные этапы компьютерного моделирования:

- Постановка задачи и ее анализ.
- Построение информационной модели.
- Разработка метода и алгоритма исследования модели.
- Разработка компьютерной модели.
- Проведение компьютерного эксперимента.
- Анализ результатов.

При *постановке задачи* необходимо в первую очередь учитывать цель моделирования. Цель определяет, какие свойства объекта являются существенными, главными для моделирования, а какими можно пренебречь. При постановке задачи указывают, какие именно результаты и в каком виде должны быть получены в итоге моделирования. Анализ моделируемого объекта должен определить, какие данные потребуются для моделирования и есть ли какие-либо ограничения для них.

В ходе *построения информационной модели* предстоит определить параметры модели и их взаимосвязи. Необходимо выбрать условные обозначения и описать

объект в выбранной знаковой системе. Это могут быть математические формулы, отображающие связь параметров модели, или структурные модели — графики, схемы, чертежи. Может быть составлено словесное описание или использованы геометрические конструкции либо записи логических умозаключений.

На основе созданной информационной модели выбирается *метод получения результатов* и вырабатывается *план действий*, разрабатывается *алгоритм*.

При разработке компьютерной модели надо тщательно подобрать *средства реализации*, наиболее подходящие для конкретной задачи и созданной информационной модели. Например, если была построена табличная, схематическая или математическая информационная модель — подходящими могут оказаться электронные таблицы, для текстовой разумно использовать текстовый редактор, для математических, схематических моделей и алгоритмов — языки программирования. После создания компьютерной модели надо проверить ее правильность, чтобы исправить на этом этапе все ошибки, если они были допущены.

Перед проведением компьютерного эксперимента необходимо составить его *план*. В соответствии с этим планом провести испытания компьютерной модели. Как правило, проводят серию экспериментов для различных значений параметров модели. Зачастую такие испытания проходят в форме диалога человека и компьютера.

Анализ полученных результатов должен привести к ответу на вопросы, которые были заданы при постановке задачи. Возможно, придется внести изменения в модель или даже постановку задачи и повторить некоторые этапы моделирования.

2.6. МАТЕМАТИЧЕСКИЕ ИНСТРУМЕНТЫ, ДИНАМИЧЕСКИЕ (ЭЛЕКТРОННЫЕ) ТАБЛИЦЫ

2.6.1. Таблица как средство моделирования. Ввод данных в готовую таблицу, изменение данных, переход к графическому представлению

Таблица может служить знаковой информационной моделью различных объектов. С ее помощью можно описывать свойства и состояние объекта и его взаимосвязи с внешним миром.

Программы работы с электронными таблицами называют *табличными процессорами*. Они позволяют решать задачи, связанные с числовыми расчетами, анализировать большие объемы данных, создавать двух- и трехмерные графики и диаграммы. Один из наиболее распространенных табличных процессоров — MS Excel.

Основные элементы электронной таблицы

КНИГИ И ЛИСТЫ

Документ MS Excel называется *книгой*, или *рабочей книгой*. Каждая книга содержит один или несколько *листов*, поэтому в одном документе можно разместить различные сведения и установить между ними необходимые связи. *Яр-*

лыки этих листов расположены внизу окна (над строкой состояния) и служат для переключения между листами (рис. 2.98).

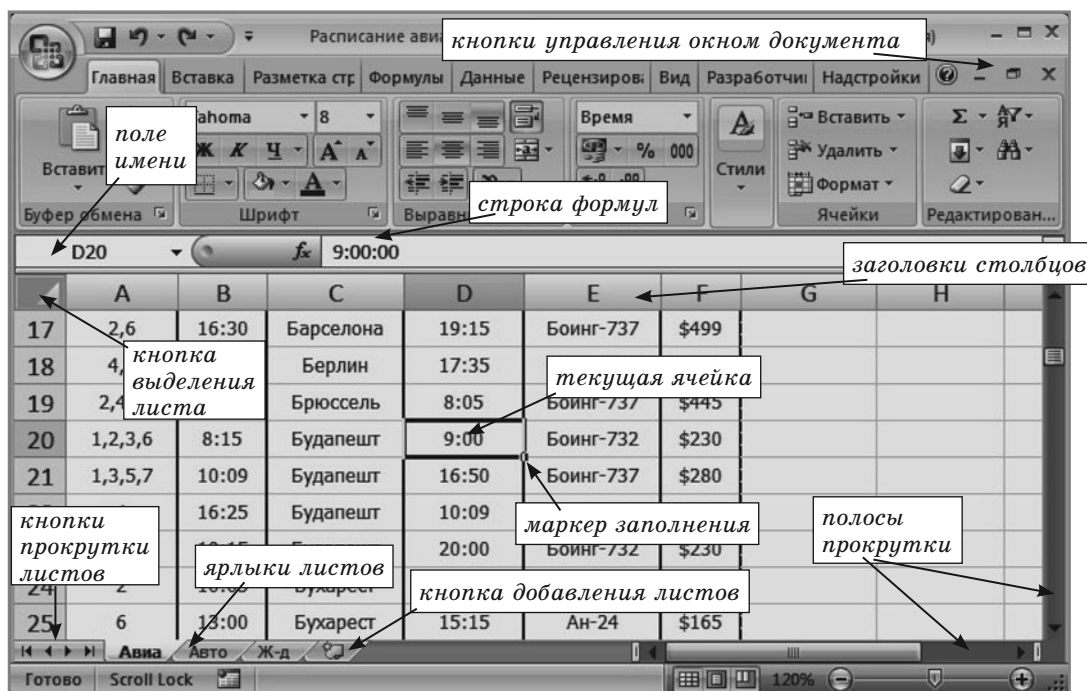


Рис. 2.98. Элементы электронной таблицы


Листы служат для хранения и обработки данных. В общем случае они состоят из *ячеек*, которые, собственно, и содержат данные, формулы и т. д. В книгу можно также вставить *листы диаграмм* для графического представления данных и модули для создания и хранения макросов, используемых при выполнении специальных задач.

В рамках одного проекта разумно использовать разные листы для решения разных задач или размещения различных таблиц. Для разных проектов лучше создать разные файлы.

Первоначально книга содержит 3 листа, именуемых *Лист1*, *Лист2*, *Лист3*. Название текущего листа (с которым в данный момент ведется работа) выделено жирным шрифтом. В книгу можно добавить любое количество листов (ограничения зависят от мощности компьютера). Можно также и удалять листы, перемещать или копировать их в пределах одной книги или из одной книги в другую.

Для **перехода** к другому листу надо щелкнуть мышью по его ярлыку (можно воспользоваться комбинациями клавиш: Ctrl+PgDn — перейти на лист, следующий за текущим; Ctrl+PgUp — на предыдущий лист).

Для операций с листом книги (для добавления, удаления, переименования, копирования, перемещения и др.) служит контекстное меню, вызываемое щелчком правой мыши на ярлыке этого листа (рис. 2.99).

При **добавлении** нового листа (команда *Вставить*) пользователю предстоит сначала подтвердить в появляющемся диалоговом окне, что будет добавлен именно лист. Новый лист будет вставлен перед текущим. Однако проще для вставки листов использовать кнопку  *Вставить лист* (справа от всех ярлыков) — щелчок по ней сразу добавляет лист в конец книги.

При **удалении** листа (команда контекстного меню *Удалить*) надо быть весьма осторожным и иметь в виду, что отменить удаление листа будет невозможно (как, впрочем, и другие действия с листами — вставку, переименование и т. д.).

При **переименовании** листа (команда *Переименовать*) следует учесть, что его название может быть не длиннее 31 символа и не должно содержать квадратные скобки, двоеточие, прямую или обратную наклонную черту. Целесообразно с самого начала работы с листом переименовать его так, чтобы название было информативным.

Для **перемещения** листа необходимо ухватить мышью ярлык с названием и перетащить его на требуемое место. **Копирование** листа осуществляется аналогично, но при нажатой клавише Ctrl. Другой способ — выбрать в контекстном меню листа команду *Переместить/скопировать* и указать, перед каким листом должен быть помещен выбранный лист (при этом для копирования надо установить флажок *Создать копию*). Именно этот способ позволит переместить либо скопировать листы в другую книгу Excel.

Операции вставки и удаления, копирования и перемещения можно осуществлять сразу для целой группы листов. Для этого необходимо выделить листы, щелкая на их ярлыках с нажатой клавишей Ctrl. Для выделения связанного диапазона листов можно щелкнуть на первом из них, нажать клавишу Shift и щелкнуть на последнем.

Чтобы облегчить выбор нужных листов, можно пометить их ярлыки разными цветами (команда контекстного меню *Цвет ярлыка*).

Операции с листами можно выполнять и с помощью ленты — аналогичные вышеописанным команды находятся в выпадающих списках кнопок группы *Ячейки* на вкладке *Главная*.

Одновременно может быть открыто несколько книг Excel. При открытии очередной книги ее окно размещается поверх всех предыдущих. Перечень всех открытых книг отображается в выпадающем списке кнопки *Перейти в другое окно* в группе *Окно* (рис. 2.100) на вкладке ленты *Вид*. Выбор названия книги из этого списка сделает ее активной (разместит ее окно поверх остальных).

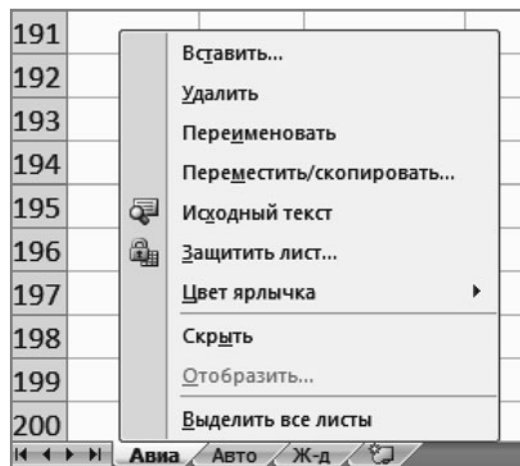


Рис. 2.99. Контекстное меню листа

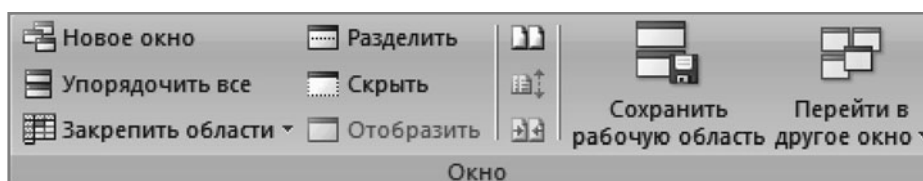


Рис. 2.100. Группа кнопок ленты для работы с окнами документов

Можно просматривать на экране несколько книг одновременно. Для этого в группе *Окно* вкладки *Вид* следует нажать кнопку *Рядом* — книги будут расположены одна над другой (рис. 2.101).

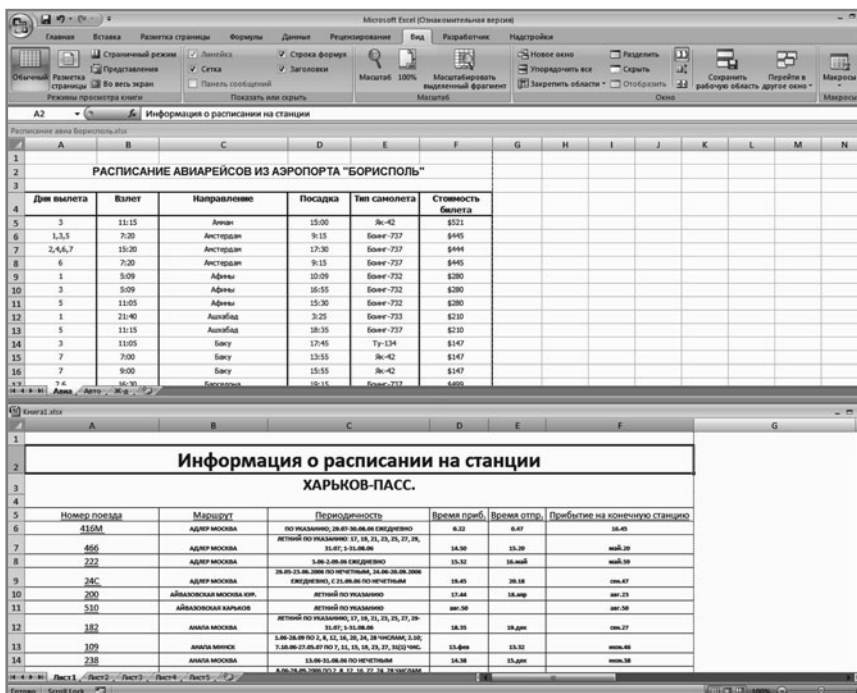


Рис. 2.101. Расположение рядом двух документов Excel

Можно отобразить одновременно все открытые книги. Для этого в группе кнопок *Окно* предназначена кнопка *Упорядочить все*.

Одновременно активным может быть только лист одной из книг. Заголовок окна активной книги выделен более ярким цветом, в нем имеются кнопки управления окном. Для перехода к окну другой книги следует щелкнуть в любом месте этого окна. Для перехода между открытыми книгами используются комбинации клавиш: **Ctrl+F6** или **Ctrl+Tab**, в обратном направлении — **Ctrl+Shift+F6** или **Ctrl+Shift+Tab**.

ЯЧЕЙКИ

Лист данных представляет собой двумерный массив. Он разбит на сеть *ячеек*, объединенных в строки и столбцы.

Каждый столбец имеет свой код, обозначаемый буквами латинского алфавита — от **A** до **Z** и далее **AA**, **AB**, **AC** до **ZZ**, затем **AAA**, **AAВ** и т. д. до **XFD**. Каждая строка имеет свой порядковый номер, начиная с **1** до **1048576**. Таким образом, максимальный размер листа — **1048576** строк на **16384** столбца. По сути, такие размеры листа позволяют считать его практически бесконечным.

Код столбца и номер строки формируют уникальный **адрес ячейки**, например: **A12**, **D7** и т. д. **Полный адрес ячейки** включает также название рабочей книги и рабочего листа:

[имя_книги]имя_листа!адрес_ячейки
 например: **[Финансы.xlsx]Лист1!B20**

Если обращение к ячейке происходит в той же книге, то имя книги не указывают:

имя_листа!адрес_ячейки
например: Лист1!B20

Чаще всего к ячейке обращаются на том же листе, поэтому не указывают и имя листа: B20.

Для обращения к прямоугольному диапазону ячеек через двоеточие указывают адрес левой верхней и правой нижней ячеек диапазона, например A1:C30.

Кроме адреса, ячейке можно присвоить уникальное имя (и даже не одно) и обращаться к ней по этому имени (об именах см. ниже в п. 2.6.2 «Ввод математических формул и вычисления по ним»).

Ячейка, в которой отображаются вводимые данные, называется **текущей**. В отдельный момент времени существует только одна текущая ячейка, при этом на экране она имеет жирную границу. Адрес текущей ячейки отображается в поле адреса (называемом также полем имени, см. рис. 2.98), а заголовки ее строки и столбца выделяются цветом.

Ячейки могут содержать данные — текстовые, числовые, даты, время и т. п. — и формулы. Кроме того, для пояснения содержимого ячеек можно прикреплять к ним примечания (такие ячейки помечаются в правом верхнем углу красным треугольником).

СТРОКА ФОРМУЛ

Если ячейка содержит формулу, последняя отображается в **строке формул**, а в самой ячейке будет виден результат расчета этой формулы.

Строку формулы можно развернуть (в случае слишком длинной формулы для ее полного обозрения) и свернуть обратно с помощью кнопки увеличения/уменьшения высоты в конце строки справа. При желании высоту строки формул можно изменить перетаскиванием ее нижней границы вниз.

Ввод и редактирование данных

ВВОД ДАННЫХ В ЯЧЕЙКИ

Excel позволяет вводить данные в произвольные ячейки любого листа. В ячейках могут содержаться числа, текст, даты, время суток, а также формулы.

Числа могут состоять только из следующих символов: 0 1 2 3 4 5 6 7 8 9 + — () , . / € % E e. Все другие сочетания цифр и нецифровых символов рассматриваются как текст. Стоящие перед числом знаки «плюс» (+) игнорируются. В качестве разделителя целой и дробной частей числа используется знак, указанный в настройках операционной системы. В русифицированной версии Windows таким знаком является запятая (,).

Каждую операцию ввода данных следует завершать нажатием клавиши Enter или Tab. В первом случае курсор после завершения ввода переместится в ячейку ниже текущей, во втором — правее текущей. Те же клавиши в комбинации с нажатой клавишей Shift перемещают курсор в обратном направлении — соответственно вверх или влево от текущей ячейки. Для отказа от ввода данных служит клавиша Esc.


Введенные в ячейку значения отображаются в строке формул.

Редактировать данные можно непосредственно в самой ячейке (для этого надо дважды щелкнуть на ячейке либо нажать F2) или в строке формул (выделив ячейку и установив курсор в строку формул).

Если длина вводимых данных превышает ширину ячейки, Excel преобразует их внешний вид. Числа с большим количеством цифр могут быть на экране округлены (рис. 2.102, ячейка A3) либо выведены в научном формате (рис. 2.102, ячейка B3). Если ширина ячейки совсем мала, вместо чисел, дат или времени в ячейке отображаются знаки ### (ячейка A4) — эти символы подсказывают, что результат не помещается в ячейке и следует, например, расширить столбец.

	A	B	C
1			
2	Среднее число продаж	общая сумма продаж	
3	5,5555556	2,22222E+11	
4	#####		

Рис. 2.102. Отображение данных в ячейках Excel

Длинное текстовое значение будет отображено полностью и «заслонит» соседнюю ячейку, если она пуста (рис. 2.102, ячейка B2); но если в соседней ячейке есть другие данные — в исходной будет видна лишь часть введенного текста (как в ячейке A2). Чтобы этот текст отобразить в одной ячейке в несколько строк, можно воспользоваться кнопкой  *Перенос текста* в группе *Выравнивание* вкладки *Главная*.

Вне зависимости от отображения на листе данные в ячейке сохраняются корректно; увидеть сохраняемые значения можно в строке формул.

ПЕРЕМЕЩЕНИЕ КУРСОРА В ТАБЛИЦЕ

Для перемещения по ячейкам таблицы используются клавиши со стрелками, Tab (для перехода к ячейке справа), Shift+Tab (к ячейке слева), Home (в начало строки), Ctrl+Home (в начало листа). Комбинация клавиш Ctrl+Backspace прокручивает экран к текущей ячейке, Ctrl+клавиша со стрелкой — к краю текущей области данных (к последней непустой ячейке или краю листа).

Перемещение курсора на один экран вверх или вниз обеспечивают соответственно клавиши PgUp или PgDn, на один экран вправо или влево — соответственно Alt+PgUp или Alt+PgDn.

Клавиша End используется несколько иначе, чем обычно принято в приложениях: ее нажатие включает/отключает режим КНЦ, при котором клавиша со стрелкой перемещает курсор (однократно) к краю текущей области данных.


Для перемещения к другому листу таблицы достаточно щелчка мышью по его ярлыку. Однако если книга содержит большое число листов, некоторые ярлыки могут быть не видны, скрываясь под горизонтальной полосой прокрутки — тогда надо сначала сдвинуть диапазон видимости листов с помощью пиктограмм со стрелками-треугольниками (слева от ярлыков). Кроме того, удобно использовать для перехода комбинации клавиш Ctrl+PgDn (к следующему листу), Ctrl+PgUp (к предыдущему листу).

Пример 1

Создать таблицу данных о клиентах (рис. 2.103).

	A	B	C	D
			Населенный	
1	Фамилия	Имя	пункт	Телефон
2	Смирнова	Настасья	Москва	957-80-56
3	Трофименко	Ольга	Рязань	433-76-53
4	Федоренко	Павел	Рязань	635-43-37
5	Воронов	Андрей	Тула	921-78-83
6	Чернова	Мария	Москва	606-76-32

Рис. 2.103. Данные для ввода в таблицу Excel

Решение. Для создания таблицы прежде всего следует создать новый документ Excel (например, комбинацией клавиш Ctrl+N) и сохранить его (например, комбинацией клавиш Shift+F12). В процессе сохранения нужно присвоить файлу имя и выбрать папку для его размещения. В дальнейшем в процессе работы не забывать регулярно сохранять файл нажатием кнопки  Сохранить.

Теперь следует установить курсор в ячейку A1 и последовательно набирать данные, завершая ввод в каждую ячейку нажатием клавиши Enter (чтобы вводить данные по столбцам) или Tab (чтобы осуществлять ввод по строкам). Обратите внимание, что при вводе текста «Рязань» в ячейку C4 и текста «Москва» в ячейку C6 Excel «помогает» набрать данные. Такой механизм называется **автозавершением ввода**: после набора первого символа текста программа предлагает повторить текст из того же столбца C, начинающийся на такой же символ (рис. 2.104, а). Чтобы согласиться с предложением автозавершения ввода, достаточно нажать Enter. Если предложенный вариант не подходит, следует просто продолжать набор данных.

После окончания набора таблица будет иметь вид, схожий с таблицей на рис. 2.104, б, — длинные текстовые строки не поместились в ячейки, данные выровнены по левому краю ячеек, внешний вид строки заголовка ничем не отличается от остальных строк. (Однако данные в таблице должны быть введены верно — например, убедитесь, что в ячейке C1 содержится текст «Населенный пункт», а не его фрагмент: установите курсор в эту ячейку и посмотрите на содержимое строки формул — см. рис. 2.104, б.) Чтобы оформить таблицу так же, как на рис. 2.103, ее надо будет отредактировать.

	A	B	C	D
			Населенный	
1	Фамилия	Имя	пункт	Телефон
2	Смирнова	Настасья	Москва	957-80-56
3	Трофименко	Ольга	Рязань	433-76-53
4	Федоренко	Павел	Рязань	635-43-37
5	Воронов	Андрей	Тула	921-78-83
6	Чернова	Мария	Москва	606-76-32

Москва	957-80-56
Рязань	433-76-53
Рязань	635-43-37
Тула	921-78-83
Москва	606-76-32

а

б


Рис. 2.104. Автозавершение ввода (а) и результат ввода данных (б)

ВЫДЕЛЕНИЕ ДАННЫХ

Выполнение большей части команд и задач в Microsoft Excel становится возможным после выделения ячеек, с которыми необходимо произвести те или иные действия. Способы выделения описаны в табл. 2.5.

Таблица 2.5

Способы выделения ячеек, диапазонов, листов

Цель выделения	Действия
Отдельная ячейка	Щелкнуть мышью по ячейке или перейти к ней, нажимая клавиши перемещения
Диапазон ячеек	Протащить указатель от первой до последней ячейки диапазона
Несмежные ячейки / диапазоны ячеек	Выделить первую ячейку (или первый диапазон ячеек), затем, удерживая нажатой клавишу Ctrl, выделить остальные ячейки (или диапазоны)
Большой диапазон ячеек	Указать первую ячейку, затем, удерживая нажатой клавишу Shift, щелкнуть по последней ячейке диапазона. Для перемещения к последней ячейке можно использовать полосы прокрутки
Все ячейки листа	Щелкнуть по кнопке  в левом верхнем углу листа, где пересекаются заголовки строк и столбцов
Всю строку	Щелкнуть по заголовку строки или нажать Shift+пробел
Весь столбец	Щелкнуть по заголовку столбца или нажать Ctrl+пробел
Смежные строки / столбцы	Протащить указатель по заголовкам строк (столбцов) или выделить первую строку (первый столбец), затем, удерживая нажатой клавишу Shift, выделить последнюю строку (столбец)
Несмежные строки / столбцы	Выделить первую строку (первый столбец), затем, удерживая нажатой клавишу Ctrl, выделить остальные строки (столбцы)
Смежные листы	Щелкнуть мышью по ярлыку первого листа, затем, удерживая нажатой клавишу Shift, щелкнуть по ярлыку последнего
Несмежные листы	Щелкнуть мышью по ярлыку первого листа, затем, удерживая нажатой клавишу Ctrl, выделить ярлыки остальных листов

Вставка, удаление и очистка данных

ВСТАВКА ЯЧЕЕК, СТРОК, СТОЛБЦОВ, ЛИСТОВ

Для добавления (пустых) ячеек, строк и столбцов используется команда *Вставить* в группе *Ячейки* вкладки *Главная* (рис. 2.105). Чтобы добавить в таблицу строку или столбец, надо выбрать из выпадающего списка этой команды соответственно пункт *Вставить строки на лист* либо *Вставить столбцы на лист* (рис. 2.106, а). Строка или столбец будут вставлены перед текущей позицией. Если были выделены несколько строк/столбцов, то столько же строк/столбцов будут вставлены перед выделенным диапазоном.

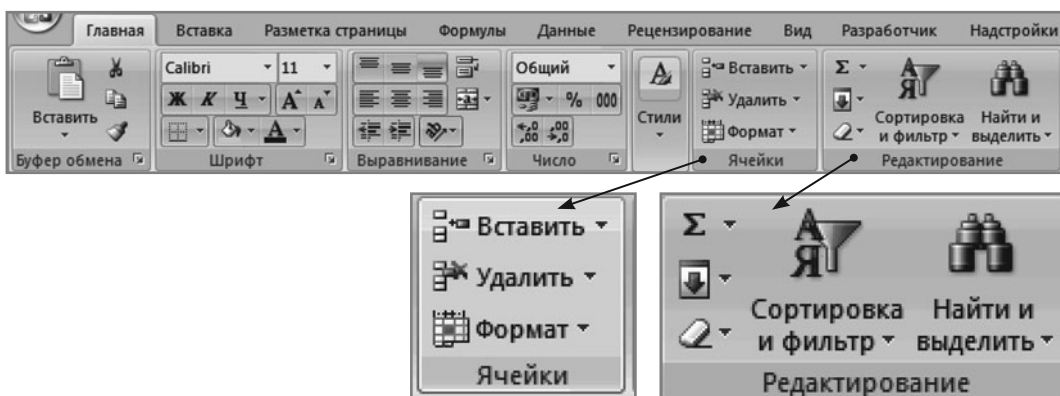


Рис. 2.105. Группы Ячейки и Редактирование вкладки Главная

Для добавления ячеек можно использовать как саму команду *Вставить*, так и пункт *Вставить ячейки* в ее выпадающем списке. При этом имеющиеся на листе ячейки раздвигаются, освобождая место для вставляемых. Направление этого сдвига необходимо указать в момент вставки (рис. 2.106, б).

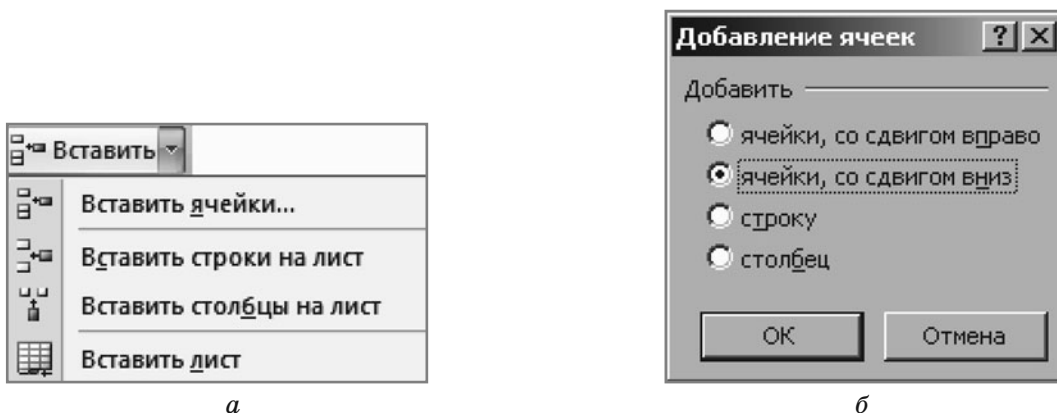


Рис. 2.106. Команды (а) и параметры (б) вставки элементов листа

Аналогично сдвигаются строки/столбцы таблицы в процессе вставки новых строк/столбцов — в сторону больших номеров. При этом общее число ячеек, строк и столбцов на листе всегда остается неизменным — за счет удаления элементов с дальнего его края.

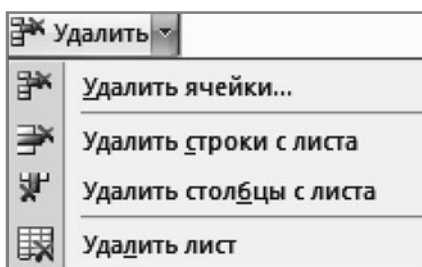
Еще один удобный способ добавления ячеек, строк или столбцов — щелчок правой кнопкой мыши (по ячейке, заголовку строки или столбца соответственно) и выбор из контекстного меню команды *Вставить*.

Для добавления нового листа служит пункт *Вставить лист* в выпадающем списке команды *Вставить* (см. рис. 2.106, а).

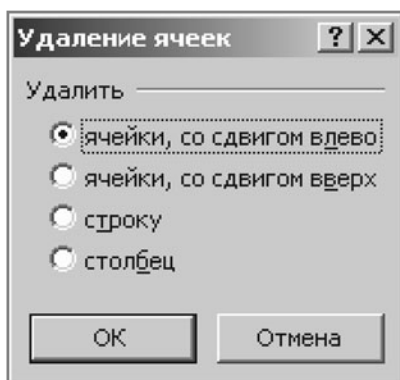
УДАЛЕНИЕ И ОЧИСТКА ЯЧЕЕК, СТРОК, СТОЛБЦОВ, ЛИСТОВ

Можно легко удалять ячейки или очищать их содержимое. При *удалении ячеек* они исчезают с листа и прилегающие к ним ячейки сдвигаются, чтобы заполнить освободившееся пространство, а на их место сдвигаются их соседи (до самого края листа, где добавляются новые ячейки, так что их количество всегда

остаётся постоянным). При *очистке ячеек* их содержимое, форматы или примечания уничтожаются, но сами ячейки остаются на листе.




а



б

Рис. 2.107. Команды (а) и параметры (б) удаления элементов листа

Для удаления используется команда *Удалить* в группе *Ячейки* вкладки *Главная* (рис. 2.105) и пункты из ее выпадающего списка: *Удалить ячейки*, *Удалить строки с листа*, *Удалить столбцы с листа*, *Удалить лист* (рис. 2.107, а). При удалении ячеек потребуется с помощью переключателя *Удалить* задать направление, в котором следует сдвигать прилегающие ячейки (рис. 2.107, б).

Чтобы очистить ячейки, надо их выделить, и в группе *Редактирование* вкладки *Главная* выбрать кнопку  *Очистить*, а затем — один из пунктов *Очистить все*, *Очистить форматы*, *Очистить содержимое* или *Очистить примечания* (рис. 2.108). Если содержимое ячейки очищено, ее значение равно нулю.

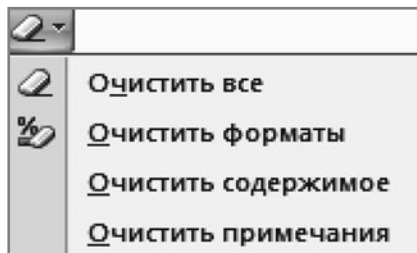


Рис. 2.108. Команды очистки данных ячейки



Пример 2

Изменить данные, приведенные в таблице примера 1 (см. рис. 2.103).

Решение. Продолжим работу с таблицей, внесем изменения в ее данные: исправим имя в ячейке В2, удалим клиента из строки 5, между столбцами В и С добавим и заполним столбец «Дата рождения» (рис. 2.109).

	А	В	С	Д	Е
1	Фамилия	Имя	Дата рожд	Населенн	Телефон
2	Смирнова	Анастасия	12.01.64	Москва	957-80-56
3	Трофимен	Ольга	17.03.85	Рязань	433-76-53
4	Федоренк	Павел	02.11.90	Рязань	635-43-37
5	Чернова	Мария	24.06.79	Москва	606-76-32

Рис. 2.109. Изменение данных о клиентах

Редактирование введенных данных можно проводить либо в строке формул, либо в самой ячейке. В первом случае надо установить курсор в ячейку (B2), а затем щелкнуть в строке формул и отредактировать данные. Завершить редактирование следует нажатием клавиши Enter (или кнопки  в строке формул), а отказаться от исправлений можно, нажав клавиши Esc (или кнопку  в строке формул). Для редактирования содержимого ячейки можно также щелкнуть дважды по ячейке (B2) и внести исправления непосредственно в ней.

Чтобы удалить всю 5-ю строку, воспользуемся, например, ее контекстным меню — щелкнем по номеру строки правой кнопкой мыши и выберем из выпавшего меню команду *Удалить*.

Для добавления столбца используем иной вариант — кнопку на ленте. Установим курсор в любую ячейку столбца C и выберем на ленте команду *Вставить столбцы на лист* (рис. 2.106, а). Заполним добавленный столбец данными. В качестве разделителей при вводе дат можно использовать точку (.), дефис (-), наклонную черту (/) — по окончании ввода даты она будет представлена в своем стандартном формате.

Автозаполнение данных

Для ввода в соседние ячейки последовательности календарных значений, чисел либо комбинации текста и чисел в MS Excel предусмотрен *механизм автозаполнения*. Он позволяет ввести лишь первое значение, а затем автоматически заполнить оставшиеся ячейки указанного диапазона определенной последовательностью.

К встроенным последовательностям относятся списки названий (полных и кратких) месяцев и дней недели, кварталов, дат, времени. На рис. 2.110 приведены примеры таких списков.

Начальное значение	Продолжение ряда			
чт	пт	сб	вс	пн
четверг	пятница	суббота	воскресенье	понедельник
фев	мар	апр	май	июн
февраль	март	апрель	май	июнь
15.02.2014	16.02.2014	17.02.2014	18.02.2014	19.02.2014
квартал 2	квартал 3	квартал 4	квартал 1	квартал 2
27.фев	28.фев	01.мар	02.мар	03.мар
9:20	10:20	11:20	12:20	13:20
				13:20 +

Рис. 2.110. Списки автозаполнения дат и времени

Для создания последовательности надо выделить ячейку с первым значением списка и потянуть мышью в заданном направлении за *маркер заполнения* — маленький черный квадрат в правом нижнем углу выделенной ячейки. Курсор мыши при этом примет форму черного крестика (рис. 2.110), а при протягивании будет видна всплывающая подсказка с очередным значением ряда.

При протягивании курсора вправо и вниз значения календарных последовательностей и комбинаций текста и чисел будут наращиваться, влево и вверх — уменьшаться. Увеличение/уменьшение происходит со стандартным шагом (рис. 2.111).

Начальное значение	Продолжение ряда			
товар 1	пт	сб	вс	пн
товар1	пятница	суббота	воскресенье	понедельник
1-й товар	мар	апр	май	июн
товар 1 заказа 5	март	апрель	май	июнь

Рис. 2.111. Автозаполнение комбинаций текста и чисел

После автозаполнения на экране ненадолго появляется кнопка *Параметры автозаполнения*, с помощью которой можно выбрать различные варианты заполнения данных. Для дат это могут быть варианты *по дням, по рабочим дням, по месяцам, по годам* (они представлены соответственно в строках 2–5 на рис. 2.112). (Можно также использовать при протаскивании правую кнопку мыши и выбирать желаемый вариант из выпадающего контекстного меню.)

	A	B	C	D	E	F	G	H
1	Начальное значение	Продолжение ряда						
2	15.11.2013	16.11.2013	17.11.2013	18.11.2013	19.11.2013			
3	15.11.2013	18.11.2013	19.11.2013	20.11.2013	21.11.2013			
4	15.11.2013	15.12.2013	15.01.2014	15.02.2014	15.03.2014			
5	15.11.2013	15.11.2014	15.11.2015	15.11.2016	15.11.2017			
6								
7	3	3	3	3				
8	книга	книга	книга	книга	книга			
9								
10								
11								
12								

Копировать ячейки

Заполнить

Заполнить только форматы

Заполнить только значения

Заполнить по дням

Заполнить по рабочим дням

Заполнить по месяцам

Заполнить по годам

Рис. 2.112. Варианты автозаполнения дат, чисел и текста

Для текстовых и числовых ячеек наращивания не происходит — данные копируются в смежные ячейки (см. строки 7 и 8 на рис. 2.112). Для организации числовых рядов (например, последовательной нумерации) следует выбирать в параметрах автозаполнения (или в контекстном меню — при использовании правой кнопки мыши) вариант *Заполнить*.

Последовательную нумерацию (с шагом 1) легко организовать, введя в ячейку первое число ряда, и затем протащив маркер заполнения этой ячейки при нажатой клавише Ctrl.

Если требуется иной (отличный от стандартного) шаг автозаполнения, надо ввести два начальных значения в смежные ячейки, выделить их и протащить маркер заполнения в направлении выделения (рис. 2.113). Два начальных значения зададут шаг дальнейшего изменения данных.

Начальные значения		Продолжение ряда		
1	2	3	4	5
1	5	9	13	17
01.01.2014	04.01.2014	07.01.2014	10.01.2014	13.01.2014
вт	чт	сб	пн	ср
товар 1	товар 3	товар 5	товар 7	товар 9

Рис. 2.113. Автозаполнение данных из двух начальных ячеек

Пример 3

Добавить в таблицу, приведенную в примере 1 (рис. 2.103), нумерацию клиентов.

Решение. Для выполнения этой задачи вставим новый столбец перед столбцом А (например, щелкнув правой кнопкой мыши на заголовке столбца А и выбрав команду *Вставить*). Введем число 1 в ячейку А2 и протащим ее маркер заполнения при нажатой клавише Ctrl. Курсор мыши при этом будет иметь вид черного крестика со знаком «+» (рис. 2.114).


	А	В
1	№ п/п	
2	1	
3		
4		
5		
6		
7		


Рис. 2.114. Создание последовательной нумерации

Копирование и перемещение данных. Буфер обмена

Операции перемещения или копирования можно производить как над ячейкой в целом, так и над ее значением, формулой или форматом по отдельности.

ПЕРЕМЕЩЕНИЕ ИЛИ КОПИРОВАНИЕ ДАННЫХ С ПОМОЩЬЮ МЫШИ

Чтобы быстро *перемещать* ячейки в пределах окна, следует их выделить и перетащить мышью границу выделенного диапазона в требуемую позицию. При захвате границы к форме курсора будет добавлена четырехнаправленная стрелка .




Чтобы *скопировать* ячейки, надо перетащить границу выделенного диапазона при нажатой клавише Ctrl. При этом во время выделения и копирования к форме курсора добавляется небольшой знак «+» — .

Если при копировании/перемещении ячеек область вставки уже содержала какие-либо данные, они будут утеряны (предварительно, конечно, Excel запросит согласие пользователя). Чтобы раздвинуть существующие ячейки и вста-

вить между ними новые, при перетаскивании выделенного диапазона следует удерживать нажатой клавишу Shift (для перемещения данных) или клавиши Shift и Ctrl (для копирования).


Переместить или скопировать ячейки за пределы окна либо в другой лист, книгу или приложение описанным способом не удастся — для этого надо использовать *буфер обмена*.

БУФЕР ОБМЕНА

Приемы работы с буфером обмена в основном аналогичны описанным выше для других программ: для копирования/вырезания данных в буфер обмена можно использовать одноименные команды контекстных меню, комбинации клавиш (Ctrl+C или Ctrl+Ins для копирования, Ctrl+X или Shift+Del для вырезания), а также одноименные кнопки  и  в группе *Буфер обмена* на вкладке ленты *Главная* (рис. 2.115). Вставка последнего скопированного/вырезанного в буфер объекта осуществляется аналогично (клавиши Ctrl+V или Shift+Ins, кнопка ).

Необходимо отметить, что при копировании/вырезании данных в буфер обмена последующая их вставка должна происходить сразу же, пока еще видна бегущая пунктирная рамка вокруг исходных данных. В противном случае буфер будет очищен. В этом принципиальное отличие работы с буфером обмена в MS Excel и, например, MS Word.

При вставке из буфера скопированных (но не вырезанных!) данных пунктирная рамка вокруг исходных ячеек сохраняется. Это означает, что скопированный фрагмент можно вставить еще раз в другую позицию. Чтобы убрать бегущий пунктир, надо нажать клавишу Esc.

Кроме того, при вставке из буфера скопированных (но не вырезанных!) данных правее и ниже области вставки появляется кнопка  *Параметры вставки*, которую используют для выбора особенностей вставки (рис. 2.116).

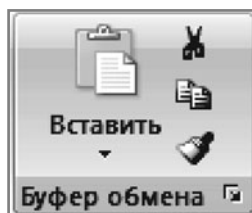


Рис. 2.115.
Группа кнопок
Буфер обмена

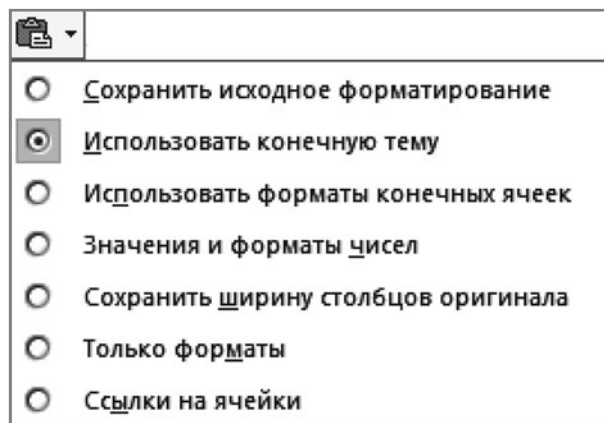


Рис. 2.116.
Параметры вставки
данных из буфера обмена

Специальная вставка

Excel имеет и свои отличительные особенности, касающиеся вставки данных из буфера обмена. Скопировав какие-либо данные с листа Excel в буфер, можно вставить часть этой информации. Например, при копировании в буфер ячейки

листа туда помещается и формула этой ячейки, и результат ее расчета, и примечание, и формат этой ячейки и т. п. При вставке вышеописанными способами на лист будет вставлена вся эта информация.

Вместо содержимого ячеек целиком можно вернуть только часть содержимого ячейки (например, возвращаемое формулой значение без копирования самой формулы).

Для частичной вставки данных или даже их предварительного преобразования служит *специальная вставка*. Для ее использования нужно предварительно скопировать ячейки в буфер обмена и выделить левый верхний угол будущей области вставки. Затем вызвать команду *Специальная вставка* из контекстного меню ячейки или из выпадающего списка кнопки ленты *Вставить* (рис. 2.117, а). Команда отобразит диалоговое окно вариантов вставки данных (рис. 2.117, б) — следует выбрать нужный вариант и нажать *ОК*. Часть вариантов доступна для выбора и в выпадающем меню кнопки *Вставить* (рис. 2.117, а).

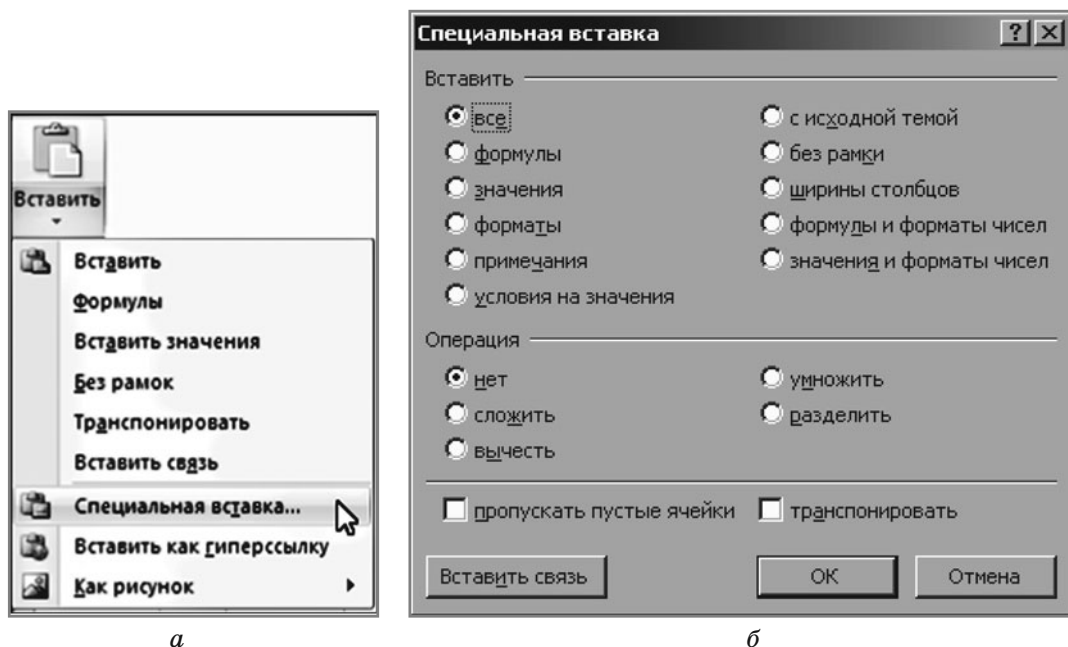


Рис. 2.117. Вызов (а) и окно (б) команды специальной вставки

Вариант *все* аналогичен обычной вставке данных из буфера обмена. Опция *транспонировать* позволяет развернуть таблицу из горизонтальной формы представления в вертикальную (данные из верхней строки области копирования отобразятся в левом столбце области вставки) либо наоборот.

Форматирование данных

MS Excel предоставляет разнообразные возможности оформления табличных данных. Можно быстро отформатировать сразу весь документ, применив встроенные или созданные пользователем темы документа. Можно самостоятельно оформить создаваемую таблицу, установив для каждой ячейки или групп ячеек необходимые параметры форматирования. Можно применить стили, которые представляют собой некоторый набор параметров форматирования.

Самые распространенные инструменты оформления таблиц вынесены на всплывающую мини-панель форматирования (рис. 2.118), которая вызывается щелчком правой кнопки мыши на ячейках.

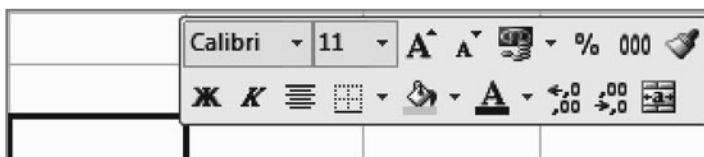


Рис. 2.118. Мини-панель форматирования

Все средства оформления таблиц в Excel удобно собраны в нескольких группах кнопок на вкладке *Главная* (рис. 2.119). Для выбора тем служит одноименная группа на вкладке *Разметка страницы*.

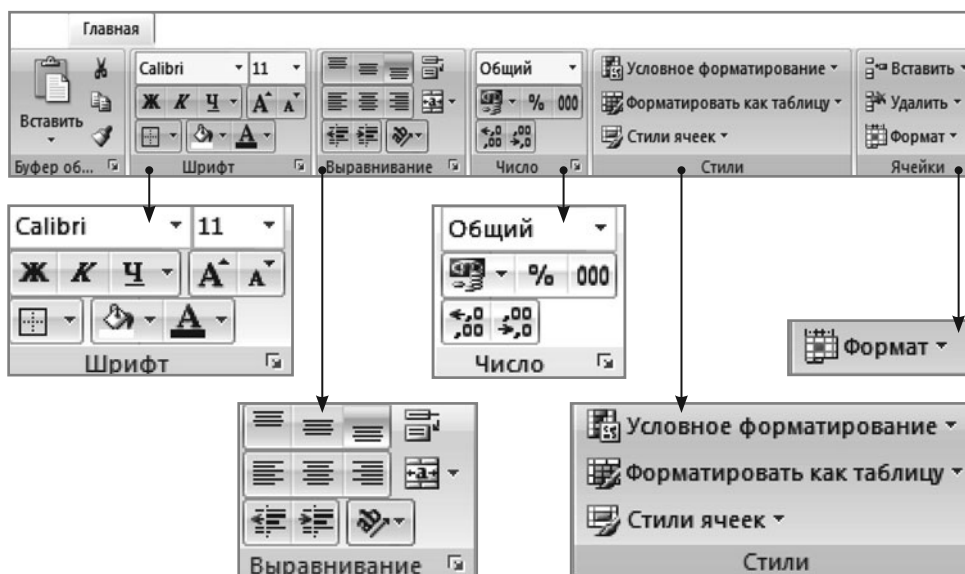


Рис. 2.119. Группы кнопок форматирования на вкладке *Главная*


ФОРМАТИРОВАНИЕ ЯЧЕЕК

Прежде чем форматировать ячейку либо группу ячеек, их предварительно необходимо выделить.

Большинство операций по оформлению ячеек электронных таблиц легко произвести с помощью кнопок ленты. Каждая из кнопок соответствует каким-либо опциям диалогового окна *Формат ячеек*. Для вызова окна *Формат ячеек* можно применить одноименную команду из выпадающего списка кнопки *Формат* на вкладке *Ячейки*. Окно форматирования легко вызвать также с помощью комбинации клавиш *Ctrl+1* или команды *Формат ячеек* контекстного меню ячейки.

ШРИФТОВОЕ ОФОРМЛЕНИЕ, ОБРАМЛЕНИЕ И ЗАЛИВКА ЯЧЕЕК

Шрифтовое оформление содержимого ячеек электронных таблиц аналогично работе со шрифтами в MS Word. Кнопки выбора гарнитуры, кегля и начертания шрифта, цвета текста и заливки сосредоточены в группе *Шрифт* вкладки *Главная*. Там же расположены кнопки увеличения / уменьшения размера шрифта.

Различные виды **обрамления ячеек** можно выбрать с помощью выпадающего списка кнопки  *Границы* в группе *Шрифт* на ленте (рис. 2.120, а). Сначала следует выбрать в этом списке цвет и тип линии будущей рамки и лишь затем указать обрамляемые границы ячейки.

Следует различать обрамление ячеек таблицы и сетку, видимую на экране. Сетка не выводится на печать и предназначена для работы с электронным вариантом таблицы.

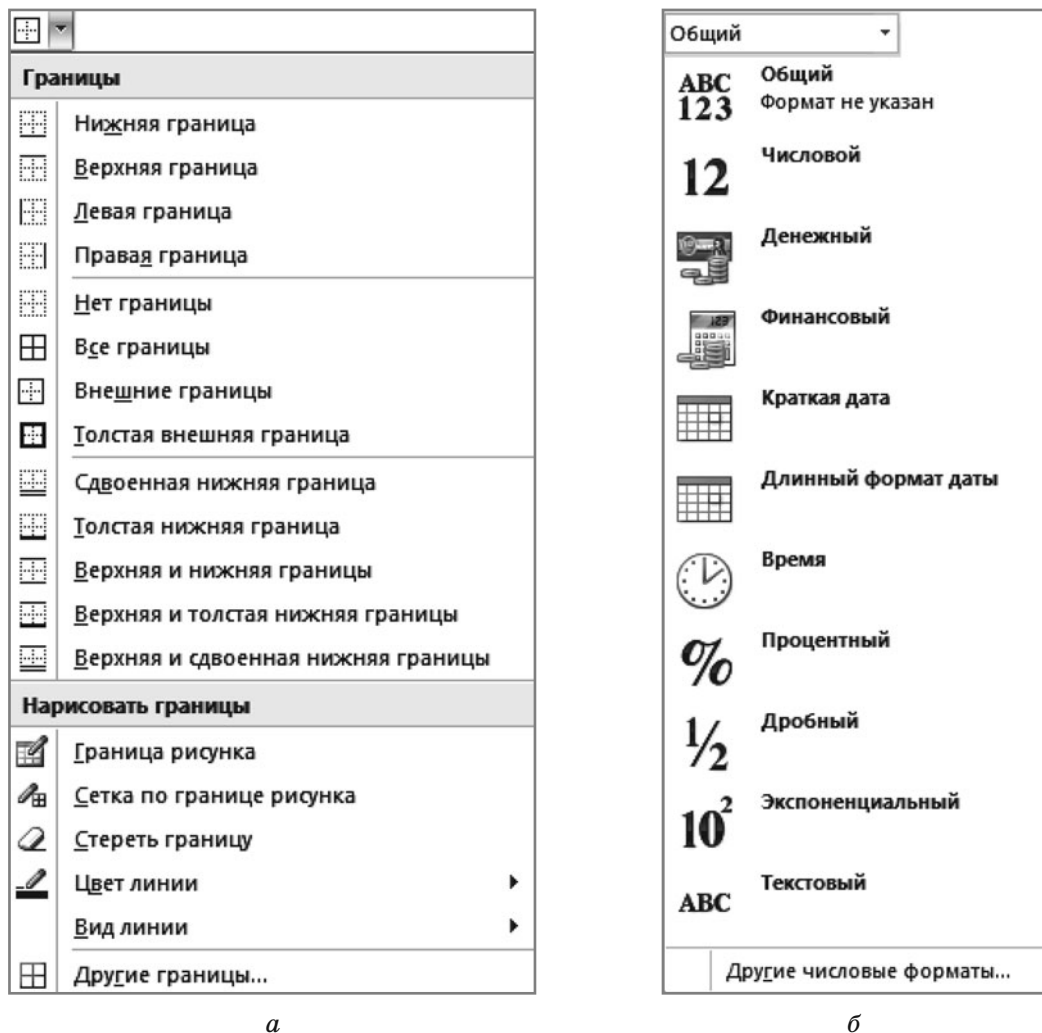




Рис. 2.120. Выпадающие списки кнопок ленты *Границы* (а) и *Числовой формат* (б)

Сплошную **заливку ячеек** выбранным цветом можно задать с помощью кнопки  *Цвет заливки* группы *Шрифт* на ленте.

ФОРМАТЫ ДАННЫХ





В ячейки таблицы Excel могут быть введены числа, текстовые данные и даты (на самом деле даты также, по сути, являются числами). Для выбора представления этих данных на экране существует ряд встроенных форматов. Первоначально для всех ячеек таблицы установлен формат «общий».

Основные **форматы данных** могут быть выбраны на ленте из выпадающего списка (рис. 2.120, б) кнопки  *Числовой формат* в группе *Число*.

Существуют следующие *форматы данных*:

- текстовый;
- числовой;
- денежный;
- процентный;
- формат времени;
- краткий и длинный форматы даты.

Параметры этих форматов в случае их выбора устанавливаются по умолчанию. Так, например, при выборе денежного формата будет указана денежная единица, установленная в среде Windows. Представление даты (в кратком или полном форматах), времени, количество цифр и разделители в записи чисел — будут приведены в соответствии с настройками Windows.

Некоторые **настройки форматов** можно отрегулировать с помощью кнопок ленты (группа кнопок *Число*) (рис. 2.119). Так, можно указать другую денежную единицу с помощью выпадающего списка кнопки  *Финансовый числовой формат*, отделить разделителями группы цифр в числе и установить два десятичных знака после запятой — кнопка  *Формат с разделителями*, увеличить либо уменьшить число десятичных знаков в дробной части числа соответственно с помощью одноименных кнопок  и .

Назначение некоторых категорий форматов данных описано в таблице 2.6.

Таблица 2.6

Категории числовых форматов Excel

Формат данных	Способ отображения данных	Пример
Общий	Без использования специальных средств	-250,777
Числовой	С разделителями групп разрядов, с выровненными десятичными разрядами и специальным отображением отрицательных чисел	-250,78
Денежный	С выровненными десятичными разрядами, символами валюты и специальным выделением для отрицательных чисел	-250,78 руб.
Дата	Как даты, время суток или время суток и даты одновременно	06.09.1900
Время	Как время суток	18:38:53
Процентный	В процентах от 1	25077,70%
Дробный	С обыкновенными дробями	250 7/9
Текстовый	Как текст (числа также будут обрабатываться как текст)	250,777

Применяя различные числовые форматы, можно изменить формат числа, не изменяя само число. Так, в таблице 2.6 первые три примера были приведены для одного и того же введенного числа -250,777. Примеры во всех остальных строках отображают одно и то же положительное число 250,777.

Числовой формат не влияет на фактическое значение ячейки, используемое в MS Excel для расчетов. Если в ячейке содержится число 250,777, а с помощью


формата отображено значение 251, то в формулах с участием этой ячейки будет использоваться фактическое значение 250,777. Фактическое значение ячейки можно увидеть в строке формул.



Часто при вводе денежных сумм пользователь указывает и денежные единицы. Excel воспринимает эти данные как текстовые и в дальнейшем, при попытке произвести расчеты, выдает ошибку. Следует вводить лишь числовые значения и устанавливать для таких ячеек денежный формат.


ВЫРАВНИВАНИЕ ДАННЫХ В ЯЧЕЙКАХ


В группе кнопок ленты *Выравнивание* (рис. 2.119) сосредоточены кнопки, позволяющие организовать расположение данных в ячейке.

По умолчанию текстовые данные в ячейках выровнены по левому краю, а числовые — по правому. Однако такое выравнивание можно изменить.

Кнопки на ленте позволяют провести всего три способа горизонтального **выравнивания** данных (левостороннее, правостороннее и центрированное) и только три — **вертикального** (позволяют прижать содержимое ячейки к ее верхнему либо нижнему краю или расположить посередине). Кнопка  *Ориентация* позволяет легко повернуть текст в ячейке по диагонали или развернуть его вертикально.

Величина абзацных отступов внутри ячейки (т. е. расстояние между текстом ячейки и ее границей) регулируется с помощью кнопок  *Уменьшить отступ* и  *Увеличить отступ*.

Часто приходится располагать заголовок таблицы по ее центру (т. е. по центру нескольких ее столбцов). Для этого текст названия набирают в первой ячейке строки заголовка, затем выделяют все требуемые ячейки в этой строке, объединяют их в одну и центрируют название по этой объединенной ячейке (см., например, заголовок таблицы на рис. 2.122). Для выполнения этой операции в группе ленты *Выравнивание* предназначена кнопка  *Объединить и поместить в центре*.

Для отмены операций объединения надо повторно нажать кнопку  или выбрать из ее выпадающего списка пункт *Отменить объединение ячеек*.

НАСТРОЙКА ОТОБРАЖЕНИЯ СТРОК, СТОЛБЦОВ, ЛИСТОВ

Для строк и столбцов программа позволяет настроить их высоту либо соответственно ширину, а также скрыть и вновь отобразить их на экране.

Изменить **ширину столбца** проще всего с помощью мыши, потянув ею за вертикальный разделитель справа от заголовка этого столбца. Аналогично для изменения **высоты строки** надо захватить и переместить разделитель снизу от ее номера-заголовка. Курсор мыши в этот момент принимает форму двунаправленной стрелки. На рис. 2.120 показан процесс изменения ширины столбца J и высоты строки 11.

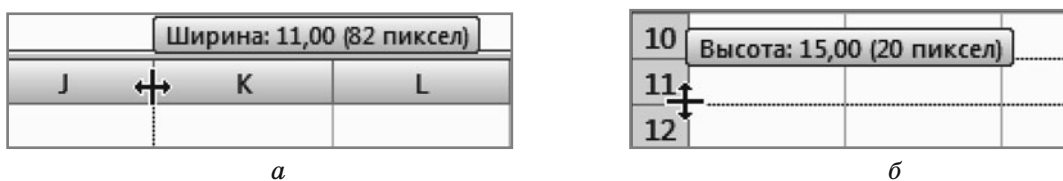


Рис. 2.121. Регулирование ширины столбца (а) и высоты строки (б)

Можно подобрать такую ширину столбца (высоту строки), которая точно соответствовала бы размеру введенных в него данных (**автоподбор размеров**). Для этого нужно дважды щелкнуть мышью по разделительной линии между заголовками необходимых строк или столбцов.

Полезной является возможность временно **скрыть** выделенные строки/столбцы. Это может пригодиться, например, если таблица содержит большое количество столбцов, они не помещаются на экране, а в данный момент нужна информация лишь из некоторых. В таком случае можно временно убрать ненужные столбцы с экрана, а позже вернуть их обратно. Для скрытия или отображения столбцов надо подтянуть мышью их правую границу к левой (для строк — соответственно нижнюю к верхней).

Команды *Скрыть*, *Отобразить*, *Ширина столбца* (*Высота строки*) присутствуют также в контекстных меню столбцов (строк).

Пример 4

Дополнить и отформатировать таблицу, приведенную в примерах 1–3.

Решение. Сначала добавим в конец таблицы столбец с указанием стоимости оформленных клиентами заказов. Затем добьемся корректного отображения данных в ячейках и выровняем их, установим оформление ячеек, оформим шапку таблицы и добавим ее заголовок (рис. 2.122).




	A	B	C	D	E	F	G
1	Данные о клиентах						
2	№ п/п	Фамилия	Имя	Дата рождения	Населенный пункт	Телефон	Стоимость заказов
3	1	Смирнова	Анастасия	12.01.64	Москва	957-80-56	255,00руб.
4	2	Трофименко	Ольга	17.03.85	Рязань	433-76-53	312,00руб.
5	3	Федоренко	Павел	02.11.90	Рязань	635-43-37	1 012,00руб.
6	4	Чернова	Мария	24.06.79	Москва	606-76-32	25 444,00руб.


Рис. 2.122. Форматирование таблицы клиентов

При вводе денежных сумм надо набрать лишь числовые значения и установить для ячеек денежный формат, тогда в них будут отображаться денежные единицы. Появление в ячейках знаков ##### (рис. 2.123) говорит о недостаточной ширине столбца (данные просто не вмещаются в столбец G), и его просто нужно расширить.


fx	12345
G	H
Стоимость заказов	
255,00р.	
322,00р.	
#####	
#####	


Рис. 2.123. Ввод денежных значений

Для того чтобы текст в ячейках шапки таблицы располагался в несколько строк, как в ячейках D2, E2, G2 на рис. 2.122, выделим все ячейки шапки (A1:G1) и нажмем кнопку  *Перенос текста* в разделе *Выравнивание*. Не снимая выделение с ячеек, организуем горизонтальное и вертикальное выравнивание в них с помощью кнопок  и  в этой же группе. Все так же не снимая выделение ячеек, установим

для них шрифтовое оформление в разделе *Шрифт*. Для цветового оформления можно использовать кнопку  *Стили ячеек* раздела *Стили*.

Теперь необходимо добиться, чтобы ширина каждого столбца соответствовала данным в его ячейках. Самый простой способ — воспользоваться автоподбором: выделить все столбцы таблицы (проведя мышью по заголовкам A:G) и дважды щелкнуть на любой разграничительной линии заголовков столбцов. Аналогично проведем автоподбор высоты первой строки таблицы. Установим выравнивание данных во внутренних ячейках таблицы в соответствии с рис. 2.121.

Для создания заголовка добавим одну строку перед таблицей (например, с помощью команды контекстного меню ячеек первой строки), наберем в ячейке A1 текст «Данные о клиентах» и, выделив ячейки A1:G1, воспользуемся кнопкой  *Объединить и поместить в центре* в разделе *Выравнивание*.

И наконец, установите оформление ячеек — для этого удобнее всего, выделив ячейки таблицы, выбрать желаемый вариант из выпадающего списка кнопки  *Границы* в разделе *Шрифт*.

2.6.2 Ввод математических формул и вычисления по ним

Простейшие расчеты

Несмотря на всю вычислительную мощь Excel, расчеты в нем организованы очень удобно и понятно. Достаточно установить курсор в любую ячейку листа, нажать клавишу = и ввести расчетную формулу, после чего нажать Enter. Результат отразится в этой же ячейке. Формула расчета будет видна в строке формул (рис. 2.124).

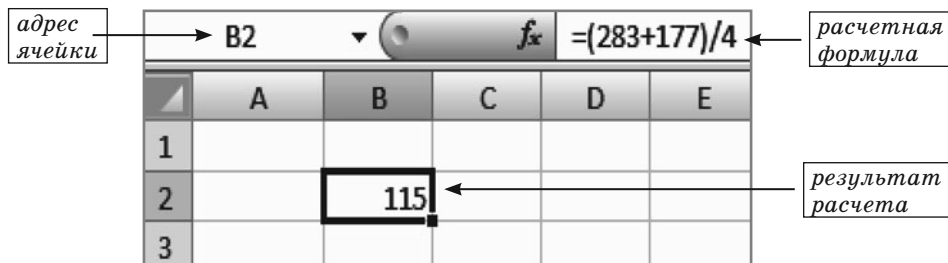




Рис. 2.124. Пример расчета в Excel

Завершить ввод формулы можно также, нажав в строке формул кнопку подтверждения , отказаться от ввода — нажав кнопку  или клавишу Esc (эти кнопки появляются в строке формул во время набора или редактирования формулы).

Знак «=» должен стоять обязательно в первой позиции формулы — перед ним не может быть пробела. Пробелы внутри формулы значения не имеют.

Если требуется исправить расчетное выражение, следует установить курсор в строку формул и внести в нее изменения.

В формулах предусмотрено использование нескольких видов операций, основные из них — арифметические и операции сравнения.

Арифметические операторы + (сложение), — (вычитание, отрицание), * (умножение), / (деление), % (процент), ^ (возведение в степень). Результатом арифметических операций являются числа.

К операторам сравнения относятся = (равно), > (больше), < (меньше), >= (больше либо равно), <= (меньше либо равно), <> (не равно). Результатом сравнения является одно из двух логических значений — либо ИСТИНА, либо ЛОЖЬ.

Когда в формуле встречается несколько операций, они выполняются в том порядке, в котором они перечислены в табл. 2.7. Операции одинакового приоритета вычисляются слева направо. Чтобы изменить этот порядок, фрагменты выражения заключают в круглые скобки.

Таблица 2.7

Приоритет выполнения операторов

Оператор	Описание
–	Отрицание
%	Процент
^	Возведение в степень
* и /	Умножение и деление
+ и –	Сложение и вычитание
&	Конкатенация (объединение нескольких текстовых строк в одну)
=, <>, >, <, >=, <=	Сравнение

Кроме операторов формула может содержать **константы** (числовые и текстовые, причем текстовые должны быть заключены в кавычки), функции и ссылки на ячейки. Например, выражение

$$= (A7 - 11) * \text{КОРЕНЬ}(B3)$$

включает *адрес ячейки A7*, *числовую константу 11* и *функцию КОРЕНЬ*. Формула будет вычислена в следующем порядке: сначала разность содержимого ячейки A7 и константы 11; затем — квадратный корень из содержимого ячейки B3, после чего оба результата будут перемножены.

Адреса ячеек и диапазонов во время редактирования формулы, их содержащей, выделяются в формуле разными цветами. При этом сами ячейки и диапазоны на листе подсвечиваются рамками тех же цветов — это значительно облегчает процесс редактирования.

Формулы автоматически пересчитываются при внесении изменений в ячейки, на которые они ссылаются. По умолчанию MS Excel автоматически пересчитывает все формулы книги при ее открытии.

Пример 1

Рассчитать стоимость товара на основании его цены, количества и скидки.
Решение. Введем исходные данные на лист (рис. 2.125, а).

	A	B	C	D
	Цена товара	Количество, шт.	Скидка	Итоговая стоимость
1				$f_x = A2*B2*(1-C2)$
2	72,30 руб.	5	10%	

D	E
Итоговая стоимость	
325,35 руб.	

Рис. 2.125. Расчет стоимости: а — исходные данные; б — результат

Установив курсор в ячейку D2, наберем формулу
 $=A2*B2*(1-C2)$

Для этого надо набрать на клавиатуре знак «=», затем щелкнуть на ячейке A2 (для вставки адреса в формулу), нажать знак умножения «*», щелкнуть на ячейке B2, еще раз нажать «*», затем открывающую скобку, 1, знак «-» и открывающую скобку. По окончании ввода нажать Enter.

В ячейке D2 отобразится результат расчета, а сама формула будет видна в строке формул, если вернуть курсор в ячейку D2 (рис. 2.125, б).

При форматировании таблицы правильным будет установить для ячеек A2 и D2 денежный формат, а для C2 — процентный.

Адресация в формулах

При перемещении ячеек, строк или столбцов ссылки на них (адреса) в формулах изменяются с учетом их нового положения. Таким образом, обновление формул происходит автоматически.

Если удалить ячейку, адрес которой входит в формулу, то формула вернет значение **#ССЫЛКА!**.

Если ячейку, содержащую формулу, переместить в другое место, то формула не изменится.

Если ячейку, содержащую формулу, скопировать в другое место, то адреса в формуле ячейки-копии сместятся в соответствии с направлением, куда была скопирована ячейка.

На рис. 2.125 можно увидеть результат копирования формулы из ячейки C5 в ячейки C4, C6 и B5.

Тот же результат, что и при копировании формулы, будет получен, если ячейку с формулой не копировать, а протащить ее за маркер заполнения (рис. 2.110) в соседние ячейки (этот процесс называется *транслированием*, или *трансляцией* формулы).

При смещении формул адреса ячеек в формулах изменяются, т. е. являются **относительными** адресами.

	A	B	C	D
1				
2		7		
3				
4			=B1+1	
5		=A2+1	=B2+1	=C2+1
6			=B3+1	

Рис. 2.126. Изменение формулы в результате копирования ячейки

Однако часто адрес в формуле требуется сохранить. В таком случае применяют **абсолютный** адрес — перед кодом столбца и номером строки добавляют знак \$ (например: \$B\$2, \$AC\$511).

Абсолютные адреса используют, если в вычислениях участвуют некие постоянные величины. Например, в расчете времени падения тела учитывают постоянную свободного падения, в расчете грузоперевозок — грузоподъемность одного автомобиля; при обмене валют — курс этой валюты и т. д.

Если в ссылке зафиксирована только одна часть адреса (например, \$C5, E\$11), то такие ссылки называются **смешанными**. При транслировании формулы, содержащей такие ссылки, будут меняться лишь незакрепленные части адреса.

Механизм смешанной, абсолютной и относительной адресации представляет собой очень важную характеристику электронных таблиц. Вместе с механизмом автозаполнения данных он служит мощным средством автоматизации расчетов.

Пример 2

Определим стоимость израсходованной электроэнергии по ежемесячным показаниям счетчика (рис. 2.127). (Для ввода названий месяцев удобно воспользоваться автозаполнением.)

месяц	показания счетчика, кВт·ч	
	начало месяца	конец месяца
январь	15120	15330
февраль	15330	15510
март	15510	15660
апрель	15660	15780
май	15780	15910

Рис. 2.127. Исходные показания для расчета

Решение. Рассчитаем сначала объем израсходованной электроэнергии — добавим столбец «Разница» (рис. 2.128), в ячейку D6 введем формулу

$$=C6-B6,$$

затем транслируем эту формулу на весь столбец D. Для ввода адресов ячеек в формулу рекомендуется щелкать мышью по этим ячейкам, а не набирать адреса на клавиатуре. Напомним, что для трансляции удобно использовать двойной щелчок мышью на маркере заполнения ячейки D6.

Поскольку цена 1 кВт · ч электроэнергии постоянна для всех месяцев, ее следует занести в отдельную ячейку. Лучше, если эта ячейка (например, B2 на рис. 2.127) будет находиться над основной таблицей — тогда сразу видно, на основании каких данных производится расчет.

Теперь можно написать формулу для расчета стоимости. В ячейку E6 надо ввести формулу

$$=D6*B2$$

Эта формула будет верна для первой строки таблицы (первого месяца расчета). Однако при попытке транслирования ее на весь столбец E в последующих

ячейках будут наблюдаться ошибки #ЗНАЧ! и неверные результаты. Так, уже для второй строки таблицы формула примет вид =D7*B3 вместо необходимой =D7*B2.

Чтобы адрес B2 не замещался в формуле при ее трансляции, его следует зафиксировать. Для этого установим курсор в формулу ячейки E6 в адрес B2 и нажмем клавишу F4. (Эту фиксацию можно было провести сразу еще при наборе формулы.) В результате формула примет вид

$$=D6*\$B\$2$$

и ее можно транслировать на весь диапазон столбца E.

В случае изменения тарифов на электроэнергию достаточно будет изменить значение в ячейке B2 — и результаты в ячейках E6:E10 будут автоматически пересчитаны.

E6		fx =D6*\$B\$2			
	A	B	C	D	E
1					
2	цена 1 кВт·ч электроэнергии	3,05 руб.			
3					
4		показания счетчика, кВт·ч			
5	месяц	начало месяца	конец месяца	разница	стоимость
6	январь	15120	15330	210	640,50 руб.
7	февраль	15330	15510	180	549,00 руб.
8	март	15510	15660	150	457,50 руб.
9	апрель	15660	15780	120	366,00 руб.
10	май	15780	15910	130	396,50 руб.

Рис. 2.128. Результат расчета

Пример 3

Для демонстрации смешанной адресации ячеек создадим таблицу умножения (рис. 2.129). Добьемся, чтобы была введена одна формула умножения — в левую верхнюю ячейку таблицы — и транслирована на всю таблицу.

Для этого введем исходные данные с помощью автозаполнения: набрав в ячейке B1 число 1, протащим маркер заполнения этой ячейки до J1 с нажатой клавишей Ctrl. Аналогично заполним первый столбец.

Установив курсор в левую верхнюю ячейку тела таблицы (B2), введем формулу

$$=B1*A2$$

B2		fx =B\$1*\$A2								
Книга1.xlsx										
	A	B	C	D	E	F	G	H	I	J
1		1	2	3	4	5	6	7	8	9
2	1	1	2	3	4	5	6	7	8	9
3	2	2	4	6	8	10	12	14	16	18
4	3	3	6	9	12	15	18	21	24	27
5	4	4	8	12	16	20	24	28	32	36
6	5	5	10	15	20	25	30	35	40	45
7	6	6	12	18	24	30	36	42	48	54
8	7	7	14	21	28	35	42	49	56	63
9	8	8	16	24	32	40	48	56	64	72
10	9	9	18	27	36	45	54	63	72	81

Рис. 2.129. Таблица умножения

Если оставить оба адреса в формуле относительными, то при трансляции формулы вправо она будет приобретать в последующих ячейках вид

$$=C1*B2$$

$$=D1*C2$$

и т. д. Если первый множитель в формуле должен изменяться, то второй (ссылка A2) должен был оставаться неизменным. Для этого следует закрепить в исходной формуле (введенной в ячейку B2) имя столбца A в ссылке A2:

$$=B1*$A2$$

Аналогично рассмотрим ситуацию трансляции формулы вниз по таблице, при которой формула будет приобретать в последующих ячейках вид

$$=B2*$A3$$

$$=B3*$A4$$

и т. д. В данном случае первый множитель должен оставаться неизменным, для чего следует в первоначальной формуле (в ячейке B2) закрепить номер строки у первого множителя B1:

$$=B$1*$A2$$

(см. рис. 2.129).

Теперь при транслировании этой формулы вниз по столбцу B первый множитель (B\$1) в формуле не будет изменяться, т. к. номер строки 1 закреплен знаком \$, а во втором множителе (\$A2) изменяться будет лишь номер строки 2, т. к. код столбца A закреплен знаком \$.

Аналогично, при транслировании формулы вправо по строке 2 в первом множителе (B\$1) будет изменяться только код столбца B (номер строки 1 закреплен), а второй множитель (\$A2) изменяться не будет (код столбца A закреплен).

Формулу следует транслировать на весь столбец B и, не снимая выделения, протянуть маркер заполнения вправо до столбца J.

Проверить правильность формулы можно в произвольной ячейке таблицы. Например, в C8 формула должна иметь вид:

$$=C$1*$A8$$

Имена ячеек и диапазонов

MS Excel позволяет присвоить ячейке, диапазону ячеек, таблице, формуле или постоянному значению (константе) собственные имена. Формулы, использующие такие имена, становятся значительно понятнее, ими легче оперировать, в них проще и быстрее найти ошибки. Формула

= Доход–Расход

яснее для понимания, чем, например,

=G10–G14

В качестве имени могут быть использованы уникальные последовательности букв, цифр, точки и знака подчеркивания длиной не более 255 символов. Имя должно начинаться с буквы, символа подчеркивания или обратной наклонной черты. Строчные и прописные буквы не различаются. Использование пробела запрещено. Вместо него обычно используют точку или знак подчеркивания.

Имена не должны иметь сходства с адресами ячеек, строк или столбцов. Например, недопустимыми являются следующие имена: D5, Z\$100, R1C7, C, R. Примеры допустимых имен: *Города*, *Данные.о.продажах*, *ИТОГ*, *Кв_3*.

Одной ячейке (диапазону и т. д.) может быть присвоено несколько имен.

Самый простой способ присвоить имя ячейке или группе ячеек — выделить ее, ввести имя в поле имени (которое расположено слева в строке формул — см. рис. 2.98) и нажать Enter. Теперь это имя можно использовать в формулах. Созданное таким образом имя будет известно во всей книге.

Встроенные функции

НАЗНАЧЕНИЕ ФУНКЦИЙ

В программу встроен обширный набор готовых формул, называемых функциями Excel. Сколь сложными (или простыми) бы ни были расчеты этих формул, для пользователя все действия сводятся к следующему: надо вставить имя функции в ячейку листа и указать для нее исходные данные (*аргументы функций*).

Перечень функций очень широк — имеются формулы для всевозможных математических вычислений, проверки свойств и значений, расчета различных статистических показателей, работы с датами и текстами и т. д. Можно использовать функции для округления данных, поиска значений, сравнения и т. п.

Общий вид любой функции Excel в формуле следующий: сначала указывается название функции, затем (без пробела) в скобках через точку с запятой перечисляются все ее аргументы. Регистр шрифта значения не имеет (поскольку программа все равно преобразовывает строчные буквы в прописные). Например, функция

=ОКРУГЛ(D2;1)

округлит значение из ячейки D2 до одного десятичного разряда. Функция

=СУММ(B3;C3;200)

вычислит сумму значений из ячеек B3, C3 и числа 200.

ВСТАВКА ФУНКЦИЙ В ЯЧЕЙКУ

Для правильной записи функции надо знать ее название, а также последовательность, назначение и тип данных ее аргументов.

Чтобы не набирать функцию с клавиатуры (в строке формул либо прямо в ячейке), в Excel имеются специальные средства для вставки функций. Для

работы с формулами и функциями на ленте имеется отдельная вкладка — *Формулы* (рис. 2.130).

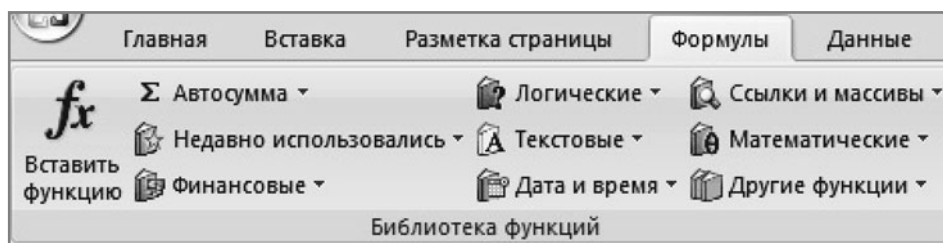


Рис. 2.130. Группа Библиотека функций вкладки Формулы

Чаще всего в расчетах применяется суммирование данных — поэтому для вставки функции суммирования (СУММ) на ленте имеется даже две кнопки. Первая — кнопка Σ *Сумма* в группе *Редактирование* на вкладке *Главная* (рис. 2.105), вторая — кнопка Σ *Автосумма* в группе *Библиотека функций* вкладки *Формулы* (рис. 2.130). Обе кнопки идентичны.

Один из наиболее простых способов рассчитать сумму значений ячеек: установить курсор в ячейку, где требуется получить результат суммирования; нажать кнопку Σ на ленте и, выделив ячейки суммирования мышью, нажать Enter (для выделения ячеек надо протянуть по ним мышью).

В результате в ячейку будет вставлена функция СУММ с заполненными аргументами — адресами выделенных ячеек. Результат расчета функции будет виден непосредственно в ячейке, а формула расчета — в строке формул.

Если ячейка результата примыкает к диапазону, который требуется суммировать, его, скорее всего, не придется даже выделять — Excel сам предложит вариант выделения ячеек (обведя их пунктиром), как правило верный. Если же предложение программы будет ошибочно, надо выделить правильный диапазон и нажать Enter.

Эту же кнопку Σ можно использовать для вычисления среднего значения, максимума или минимума нескольких значений и подсчета количества чисел. Для этого надо воспользоваться выпадающим списком кнопки (рис. 2.131).

Команда *Среднее* вставит на лист функцию СРЗНАЧ, команда *Число* — функцию СЧЁТ (подсчитывает количество чисел в выделенном диапазоне), *Максимум* и *Минимум* — функции МАКС и МИН соответственно.

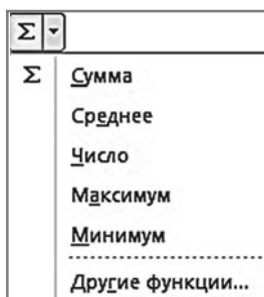


Рис. 2.131.

Выпадающий список кнопки суммирования

Команда *Другие функции* выведет на экран диалоговое окно *Мастер функций* (рис. 2.132), с помощью которого можно выбрать любую функцию из более чем 350 встроенных в программу.

Это же окно можно вызвать и другими способами:

- нажав кнопку fx *Вставить функцию* в группе *Библиотека функций* вкладки *Формулы* (рис. 2.130);
- щелкнув по кнопке fx в начале строки формул (рис. 2.124) и др.

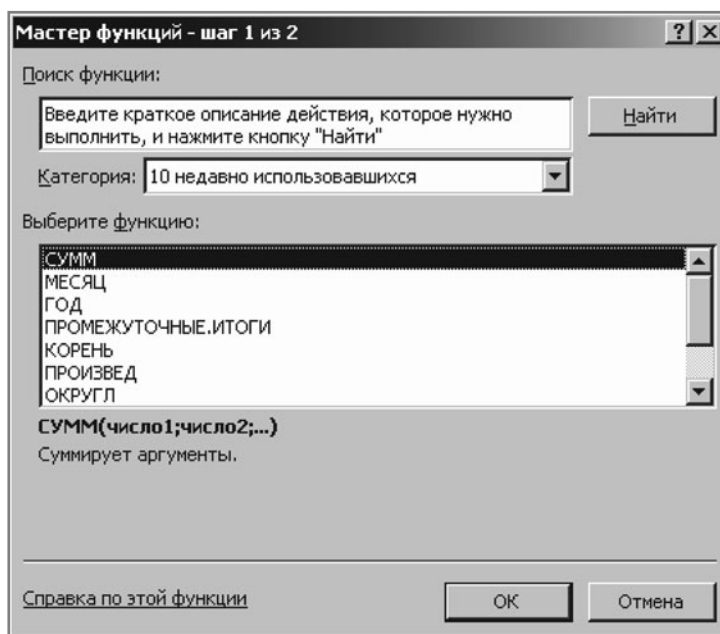


Рис. 2.132. Диалоговое окно выбора встроенной функции Excel

МАСТЕР ФУНКЦИЙ

Мастер функций помогает вставить функцию на лист Excel. Сначала предлагается выбрать необходимую функцию из списка (рис. 2.132). Для удобства выбора все функции разбиты на несколько **категорий**. В первую очередь надо выбрать категорию в одноименном выпадающем списке окна мастера функций (рис. 2.133), а затем в основной части окна *Выберите функцию* указать нужную функцию и нажать *ОК*.

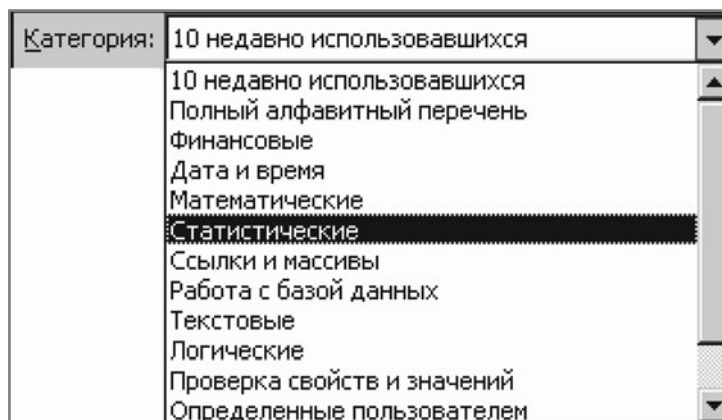


Рис. 2.133. Список категорий встроенных функций


Если неизвестно, к какой категории относится нужная функция, можно выбрать *Полный алфавитный перечень* и отыскать ее в полном списке по алфавиту.

Если же неизвестно и ее название, можно найти функцию, описав ее назначение в поле *Поиск функции* и нажав рядом кнопку *Найти*. Так, набрав в поле

поиска фразу «возведение числа в степень», увидим в поле выбора только необходимую функцию СТЕПЕНЬ.

Ниже поля выбора функции выводится краткое ее описание (см. описание функции СУММ на рис. 2.131). Эта подсказка также поможет сделать правильный выбор.

Функции наиболее используемых категорий можно выбрать и без вызова диалогового окна — на вкладке ленты *Формулы* в группе *Библиотека функций* для этого присутствуют кнопки *Финансовые*, *Логические*, *Текстовые* и т. д. (рис. 2.130).

Функции, которые использовались недавно, можно быстро найти в диалоговом окне выбора с помощью псевдокатегории «10 недавно использовавшихся» (рис. 2.133) или нажав на ленте кнопку  *Недавно использовались*.

В любом случае выбор функции завершится появлением на экране окна аргументов функции (рис. 2.134). Это окно содержит ряд полей, в которые предстоит ввести значения аргументов. Названия обязательных аргументов выделены полужирным шрифтом, а необязательные аргументы набраны обычным прямым шрифтом. При указании аргументов можно ориентироваться на подсказки в нижней части окна. После ввода значений надо нажать *ОК*.

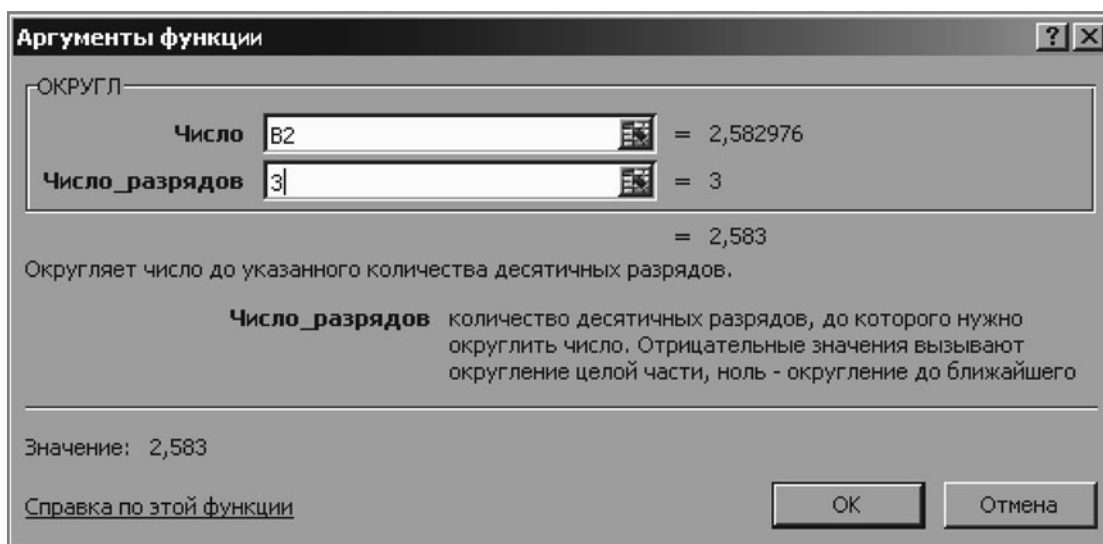





Рис. 2.134. Окно аргументов функции

Кнопка  в каждом поле аргумента позволяет временно убрать панель формул с экрана (точнее, свернуть ее в полосу) — чтобы были видны ячейки листа. Выбрав на листе необходимые ячейки, надо вернуть панель формул в раскрытое состояние с помощью этой же кнопки, имеющей теперь вид . Адреса выбранных ячеек автоматически будут вставлены в соответствующее поле аргумента.

Значения аргументов отображаются в верхней части окна, будущее значение функции — непосредственно под ними. Будущий результат расчета всей формулы (которая может включать и другие функции, операторы и др.) виден в левом нижнем углу окна.

Все вводимые значения (названия функций, их аргументы, знаки операций и т. д.) отображаются в строке формул. При этом введенные значения аргументов записываются внутри круглых скобок через точку с запятой. Так, для функции, изображенной на рис. 2.134, в строке формул появится запись:

=ОКРУГЛ(В2;3)

Для редактирования введенной функции можно снова вызвать окно ее аргументов. Для этого надо нажать кнопку  в строке формул или на ленте.

КАТЕГОРИИ ФУНКЦИЙ

Для вставки часто используемых функций можно использовать выпадающий список кнопки автосуммирования на вкладках ленты *Главная* и *Формулы*. К этим функциям относится СУММ (суммирование значений), СРЗНАЧ (расчет среднего), СЧЁТ (подсчет количества чисел), МАКС и МИН (определение максимального и минимального значений соответственно).

Остальные функции следует выбрать из списка **категорий функций**. Excel насчитывает их больше десяти — *математические, статистические, логические, текстовые, функции даты и времени, финансовые* и др.

Из **математических функций**, кроме суммирования, часто применяются функции округления и получения целого — их достаточно много: математическое округление (ОКРУГЛ), округление с недостатком (ОКРУГЛВНИЗ и ОКРВНИЗ), с избытком (ОКРУГЛВВЕРХ и ОКРВВЕРХ), получение целого (ОТБР и ЦЕЛОЕ).

Названия большинства функций достаточно содержательны, чтобы догадаться об их назначении: ПРОИЗВЕД (расчет произведения), ЧАСТНОЕ и ОСТАТ (деление нацело и остаток от него соответственно), СТЕПЕНЬ (возведение в степень), КОРЕНЬ (извлечение квадратного корня). (Функция КОРЕНЬ извлекает лишь квадратный корень, а для вычисления любого другого предстоит пользоваться функцией СТЕПЕНЬ.)

Некоторые тригонометрические и логарифмические функции имеют названия, отличные от общепринятых в математике (как, например, TAN — tg, ACOS, ASIN, ATAN — arccos, arcsin, arctg соответственно).

Очень удобна функция СУММПРОИЗВ, которая перемножает попарно значения из двух (или более) диапазонов ячеек, а затем их суммирует. Часто используется также функция частичного суммирования СУММЕСЛИ — складывает значения только тех ячеек диапазона, которые удовлетворяют некоторому условию.

Excel включает более 80 **статистических функций**, четыре из которых (СРЗНАЧ, СЧЁТ, МАКС, МИН) можно вызвать с ленты кнопкой автосуммирования (рис. 2.131). Среди них часто применяется функция СЧЁТЕСЛИ — она подсчитывает количество значений группы ячеек в зависимости от выполнения некоторого условия.

Наряду с функцией СЧЁТ имеется функция СЧЁТЗ, предназначенная для расчета количества любых (не только числовых) значений в указанном диапазоне (см. пример 5).

Кроме вычисления максимального и минимального значений группы ячеек (функции МАКС и МИН), Excel позволяет рассчитать второе, третье по поряд-

ку и т. д. наибольшее либо наименьшее значения (функции **НАИБОЛЬШИЙ** и **НАИМЕНЬШИЙ**).

Логические функции предназначены для проверки выполнения одного или нескольких условий. Три из них — **И**, **ИЛИ**, **НЕ** — возвращают в качестве результата логическое значение **ИСТИНА** или значение **ЛОЖЬ**.

Результат **ИСТИНА** получается в следующих случаях:

- все условия функции **И** — верны;
- хотя бы одно условие функции **ИЛИ** — верно;
- единственное условие функции **НЕ** — неверно.

В противном случае эти функции возвращают значение **ЛОЖЬ**.

Наиболее часто из логических функций используется функция **ЕСЛИ**. Она выдает один из двух вариантов ответа в зависимости от указанного условия. Условие указывается в первом аргументе функции, варианты ответов — соответственно во втором и третьем аргументах. Если условие принимает значение **ИСТИНА** — функция возвращает первый вариант ответа, **ЛОЖЬ** — второй вариант. Один из вариантов ответов может быть опущен.

Пример 4

Дан фрагмент таблицы Excel в режиме вывода формул. Определить значение в ячейке D7.

	A	B	C	D
1				
2	1			
3	1		2	
4		3		=СУММ(A1:A5)
5	1			
6			4	
7				=СУММ(A1:A3;B3:D4;C6)
8				

Решение. Формула в ячейке D7 предписывает сложить значения ячеек диапазонов A1:A3, B3:D4 и ячейки C6.

Сумма значений диапазона A1:A3 равна 2.


Для расчета суммы значений диапазона B3:D4 сначала надо рассчитать сумму в D4: она равна 3. Тогда сумма B3:D4 равна 8.

Общая сумма равна $2 + 8 + 4 = 14$.

Ответ: 14.

Пример 5

Решение. Рассчитать суммарные количество и стоимость товаров на основании данных о продажах (рис. 2.135). Затем вычислить среднюю цену, максимальную скидку и количество наименований проданных товаров. Указать дату проведения расчетов.

Суммарное количество проще всего рассчитать, установив курсор в ячейку C7 и нажав кнопку автосуммирования  на вкладке ленты *Главная* либо *Формулы*. Excel предложит суммировать диапазон ячеек C2:C6, выделив его пунктиром (рис. 2.136), — это соответствует нашей задаче, и достаточно только нажать Enter.

	A	B	C	D	E
1	Наименование товара	Цена товара	Количество, шт.	Скидка	Итоговая стоимость
2	Товар 1	72,30 руб.	5	10%	325,35 руб.
3	Товар 2	128,15 руб.	1	2%	125,59 руб.
4	Товар 3	66,00 руб.	3	5%	188,10 руб.
5	Товар 4	38,99 руб.	1	3%	37,82 руб.
6	Товар 5	101,50 руб.	2	5%	192,85 руб.
7	Всего				

Рис. 2.135. Данные о продажах товаров

Аналогично рассчитывается и суммарная стоимость в ячейке E7 (рис. 2.136).

Для расчета средней цены удобнее всего, установив курсор в ячейку B9, вызвать функцию расчета среднего значения (СРЗНАЧ) с помощью команды *Среднее* выпадающего списка кнопки автосуммирования (рис. 2.130). На этот раз Excel предложит усреднить ячейки, не соответствующие заданию, поскольку B9 не примыкает непосредственно к диапазону цен. Надо, не обращая внимания на выделенный диапазон, обвести ячейки с ценами (от B2 до B6) и нажать Enter.

C	D
Количество, шт.	Скидка
5	10%
1	2%
3	5%
1	3%
2	5%
=СУММ(C2:C6)	
СУММ(число1; [число2]; ...)	

Рис. 2.136. Применение автосуммирования

	A	B	C	D	E
1	Наименование товара	Цена товара	Количество, шт.	Скидка	Итоговая стоимость
2	Товар 1	72,30 руб.	5	10%	325,35 руб.
3	Товар 2	128,15 руб.	1	2%	125,59 руб.
4	Товар 3	66,00 руб.	3	5%	188,10 руб.
5	Товар 4	38,99 руб.	1	3%	37,82 руб.
6	Товар 5	101,50 руб.	2	5%	192,85 руб.
7	Всего		12		869,71 руб.
8					
9	Средняя цена:	81,39 руб.			
10	Максимальная скидка:	5%			Дата расчета:
	Количество				
11	наименований товаров:	5			23.01.2010

Рис. 2.137. Рассчитанные итоги продаж

Аналогично рассчитывается и максимальная скидка (с помощью функции МАКС).

Для подсчета количества в выпадающем списке кнопки автосуммирования есть команда *Число*, которая вставляет функцию СЧЁТ. Однако она подсчитывает количество чисел, а в задаче требуется вычислить количество названий. Для этого нужна иная функция — СЧЁТЗ из категории «Статистические». Для ее вставки, например, на вкладке ленты *Формулы* можно воспользоваться кнопкой *Другие функции / Статистические / СЧЁТЗ*. В появившемся окне аргументов функции надо выделить диапазон названий товаров и нажать *ОК*.

И наконец, выведем в отдельной ячейке сегодняшнюю дату расчетов. Для этого воспользуемся функцией СЕГОДНЯ, которая выдает каждый раз дату текущего дня (открыв этот же файл завтра, мы увидим в ячейке завтрашнюю дату и т. д.). Эта функция не имеет аргументов, поэтому для ее вставки достаточно выбрать ее в категории функций «Дата и время» и нажать *ОК*.

Теперь желательно отформатировать таблицу, установив форматы данных (денежный, процентный, даты и др.), обрамление и т. п., а затем сохранить результат. Итоговая таблица показана на рис. 2.137.


Пример 6

Подытожить результаты журнальной подписки на 1-й квартал (исходные данные приведены на рис. 2.138: в столбцах А–D — исходные данные, в столбцах F–I — результаты расчетов).

Решение. Поскольку в расчете будут использованы данные диапазонов В3:В14, С3:С14 и D3:D14, присвоим им имена *журналы*, *кол_во* и *стоимость* соответственно. Для этого, выделив поочередно каждый диапазон, введем их имена в поле имени (слева от строки формул), заканчивая каждый ввод нажатием Enter.

G3 f _x =СУММЕСЛИ(журналы;F3;кол_во)									
	A	B	C	D	E	F	G	H	I
1	Подписка на 1 квартал					Итого за 1 квартал			
2	Месяц	Название журнала	Количество экз.	Стоимость		Название журнала	Всего экз.	Стоимость	Доля в общей стоимости
3	январь	журнал 1	90	765,00 руб.		журнал 1	251	2 133,50 руб.	18,4%
4	январь	журнал 2	63	768,60 руб.		журнал 2	179	2 183,80 руб.	18,8%
5	январь	журнал 3	51	510,00 руб.		журнал 3	132	1 320,00 руб.	11,4%
6	январь	журнал 4	87	2 305,50 руб.		журнал 4	225	5 962,50 руб.	51,4%
7	февраль	журнал 1	82	697,00 руб.		Итого	787	11 599,80 руб.	
8	февраль	журнал 2	73	890,60 руб.					
9	февраль	журнал 3	40	400,00 руб.					
10	февраль	журнал 4	68	1 802,00 руб.					
11	март	журнал 1	79	671,50 руб.					
12	март	журнал 2	43	524,60 руб.					
13	март	журнал 3	41	410,00 руб.					
14	март	журнал 4	70	1 855,00 руб.					

Рис. 2.138. Данные о результатах подписки

Для суммирования в ячейке G3 общего тиража журнала 1 вставим функцию СУММЕСЛИ (категория «Математические»), например нажав кнопку  в начале строки формул. В качестве первого аргумента функции надо выделить диапазон, где будет происходить поиск журнала 1 (B3:B14), — при этом в поле аргумента будет вставлено имя диапазона (*журналы*). Вторым аргументом функции — название искомого журнала; в данном случае это журнал 1, т. е. ячейка F3. Третьим аргументом — диапазон суммирования (C3:C14); при его выделении также будет вставлено имя этого диапазона — *кол_во* (см. строку формул на рис. 2.138).

Перед транслированием формулы вниз по столбцу следовало бы зафиксировать адреса обоих диапазонов — они неизменны для любого журнала. Однако в нашей формуле вместо адресов использованы имена, по умолчанию являющиеся абсолютными ссылками, и закрепления адресов не требуется.

Аналогично строится формула для столбца H (*Стоимость*). В ячейках G7 и H7 следует рассчитать итоговые суммы с помощью автосуммирования.

Для подсчета доли предварительно присвоим имя ячейке H7 (например, *итог_стоим*). Тогда формула в ячейке I3 примет вид

$$=H3/итог_стоим$$

и также не потребует закрепления адреса ячейки итоговой суммы H7.

2.6.3. Представление формульной зависимости в графическом виде

Microsoft Excel предоставляет возможность графического представления данных в виде диаграмм. Они позволяют отобразить данные в удобном для восприятия виде либо подчеркнуть какую-либо их характеристику. Зачастую с помощью диаграмм легче сравнить данные или найти в них закономерности.

На одном листе с данными может находиться несколько диаграмм. Кроме того, диаграмма может располагаться на отдельном специальном листе.

Диаграммы связаны с данными листа, на основе которых они были созданы, и изменяются каждый раз, когда изменяются данные на листе.

Типы диаграмм

В Excel существует более десятка типов встроенных диаграмм, каждый из которых имеет несколько разновидностей. Выбор типа диаграммы зависит от задач, решаемых с ее помощью. Например, хорошим средством для демонстрации вклада отдельных частей в общую сумму является круговая диаграмма, гистограмма же позволяет проиллюстрировать отношение отдельных элементов друг к другу и их изменение во времени.

На рис. 2.139 и рис. 2.140 представлены образцы наиболее часто применяемых типов диаграмм, построенных на основании одной и той же таблицы данных (рис. 2.139, а).

Гистограмма (рис. 2.139, б) представляет собой диаграмму, на которой значения отображены в виде столбцов различной площади. Обычно эти столбцы имеют одинаковую ширину, так что различия между данными выражены в высоте столбцов.

Гистограмма показывает изменение данных за определенный период времени и иллюстрирует соотношение отдельных значений данных.

На **линейчатой диаграмме** (рис. 2.139, в) категории обычно расположены по вертикали, а значения — по горизонтали. Таким образом, уделяется большее внимание сопоставлению значений.

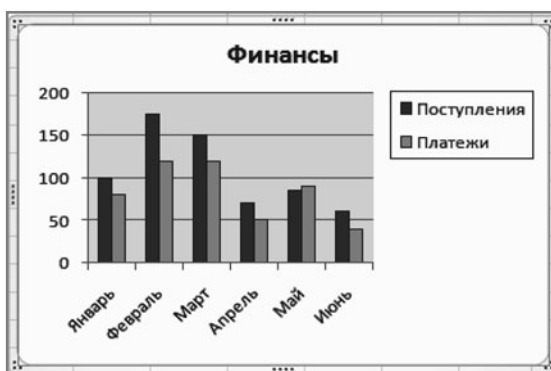
Для гистограммы и линейчатой диаграммы вместо столбцов и полос возможен выбор конусов, цилиндров и пирамид.

График (рис. 2.139, г) отражает тенденции изменения данных за равные промежутки времени. Для графика важна равномерность изменения аргумента.

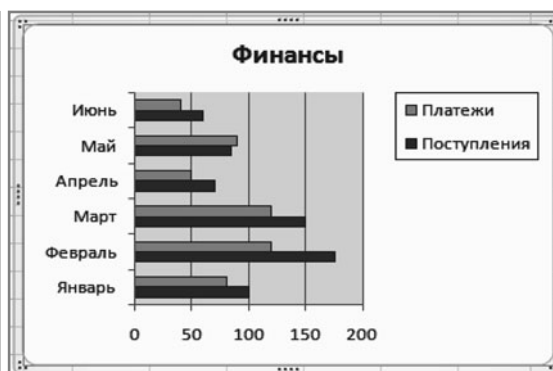
Диаграмма с областями (рис. 2.139, д) представляет собой соединение возможностей графика и гистограммы. По сути, это график, на котором часть координатного пространства под линией окрашена цветом соответствующего ряда. Она сочетает в себе возможности отображения развития процесса и сопоставления данных.

Финансы	Январь	Февраль	Март	Апрель	Май	Июнь
Поступления	100	175	150	70	85	60
Платежи	80	120	120	50	90	40
Сальдо	20	-45	-70	20	5	20

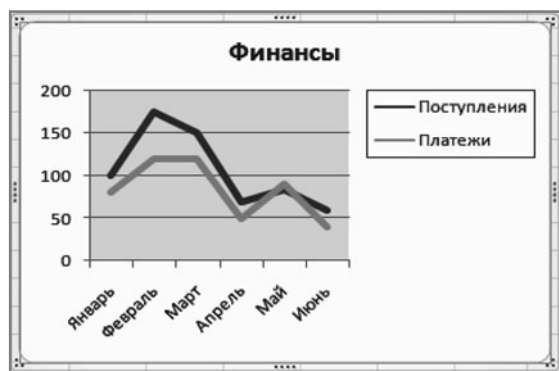
а



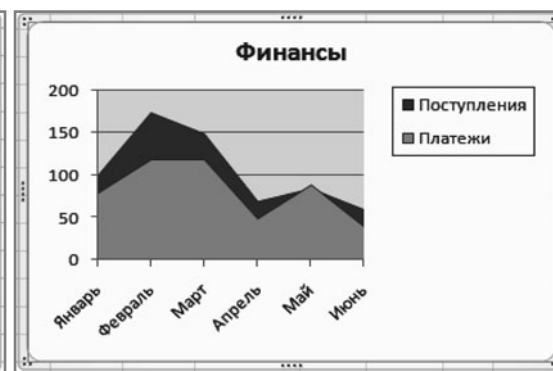
б



в



г



д

Рис. 2.139. Диаграммы Excel: а) исходная таблица данных; б) гистограмма; в) ленточная диаграмма; г) график; д) диаграмма с областями;

Для гистограмм, графиков, линейчатых диаграмм и диаграмм с областями существуют разновидности «с накоплением» и «нормированная». **Диаграмма с накоплением** показывает вклад отдельных элементов в общую сумму. **Нормированная диаграмма** отражает долю каждой компоненты в общей сумме (рис. 2.140, а).

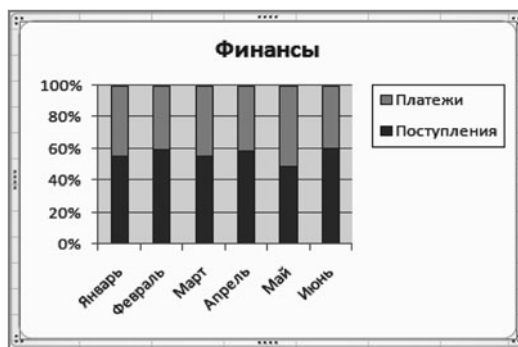
Круговая диаграмма (рис. 2.140, б) показывает величину каждого элемента ряда данных и его вклад в общую сумму. На круговой диаграмме может быть представлен только один ряд данных. Такую диаграмму часто используют, чтобы подчеркнуть какой-либо значительный элемент. Можно выдвигать сектора диаграммы, менять их местами, поворачивать ее.

Как и круговая диаграмма, **кольцевая диаграмма** (рис. 2.140, в) показывает вклад каждого элемента в общую сумму, но в отличие от круговой диаграммы она может содержать несколько рядов данных. Каждое кольцо в кольцевой диаграмме представляет отдельный ряд данных. Кольца можно раздвигать, менять местами и т. п.

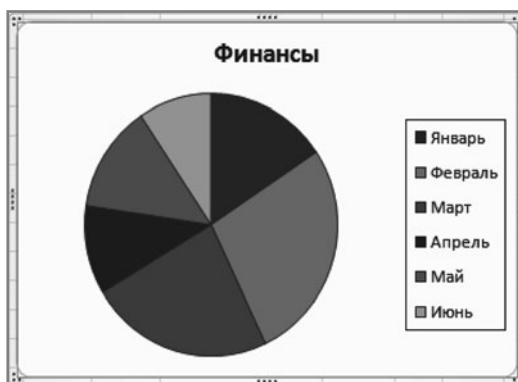
Точечная диаграмма (рис. 2.141, а) отображает взаимосвязь между числовыми значениями в нескольких рядах.

Эта диаграмма представляет две группы чисел в виде одного ряда точек в координатах xu и часто используется для иллюстрирования данных научного характера. При подготовке данных следует расположить в одной строке или столбце все значения переменной x , а соответствующие значения y — в соседних строках или столбцах таблицы. Значения на точечной диаграмме могут быть отмечены различными маркерами, соединены отрезками или сглаживающими линиями (рис. 2.141, б).

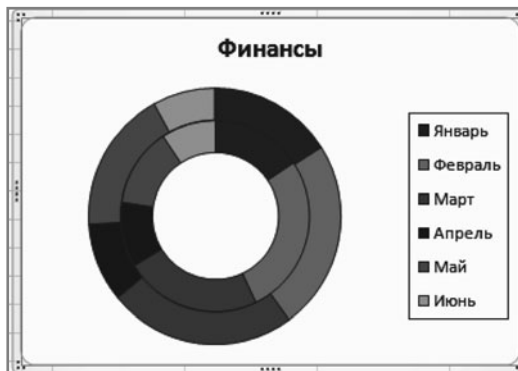
Поверхностная диаграмма (рис. 2.141, в) соединяет в себе возможности точечной диаграммы и графика. Она показывает не только связи между несколькими рядами данных, но и процесс изменения этой зависимости во времени и по категориям данных.



а



б

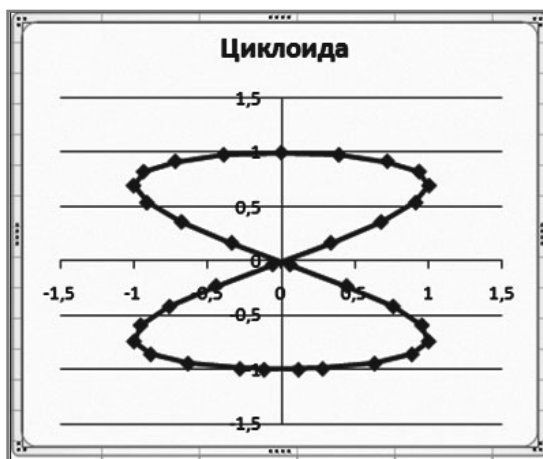


в

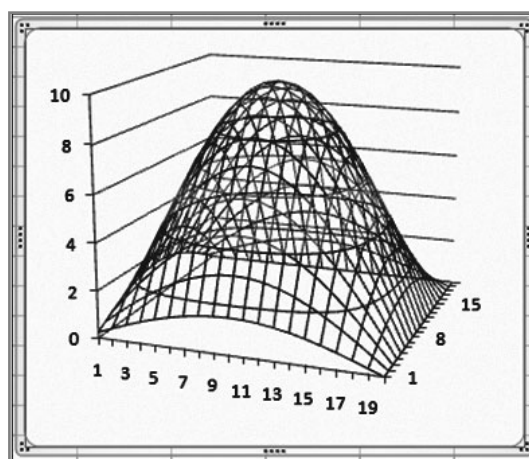
Рис. 2.140. Диаграммы Excel:
а) нормированная гистограмма с накоплением;
б) круговая;
в) кольцевая



а



б



в


Рис. 2.141. Примеры точечных диаграмм (а, б) и поверхности (в)

Построение диаграммы

Для создания диаграммы следует сначала выделить диапазон ячеек с данными для будущей диаграммы. Чтобы в ней были отражены названия строк или столбцов, надо захватывать эти названия при выделении данных. Для выделения разрозненных строк или столбцов следует удерживать дополнительно клавишу Ctrl.

Затем следует щелкнуть на вкладке *Вставка* в группе ленты *Диаграммы* по кнопке с нужным типом диаграмм и выбрать в галерее конкретный вид диаграммы (рис. 2.142). Диаграмма будет создана на том же листе и расположена поверх ячеек в отдельном слое документа.

Кнопки типов диаграмм достаточно наглядны — их значки четко демонстрируют образец будущей диаграммы. Однако в выпадающих списках этих кнопок отображается только часть возможных видов диаграмм — для обзора полного списка надо вывести окно *Вставка диаграммы* (рис. 2.143), щелкнув по кноп-

ке  группы *Диаграммы* (или выбрав команду *Все типы диаграмм*). В левой панели этого окна перечислены возможные типы диаграмм, а правая панель содержит пиктограммы их разновидностей.

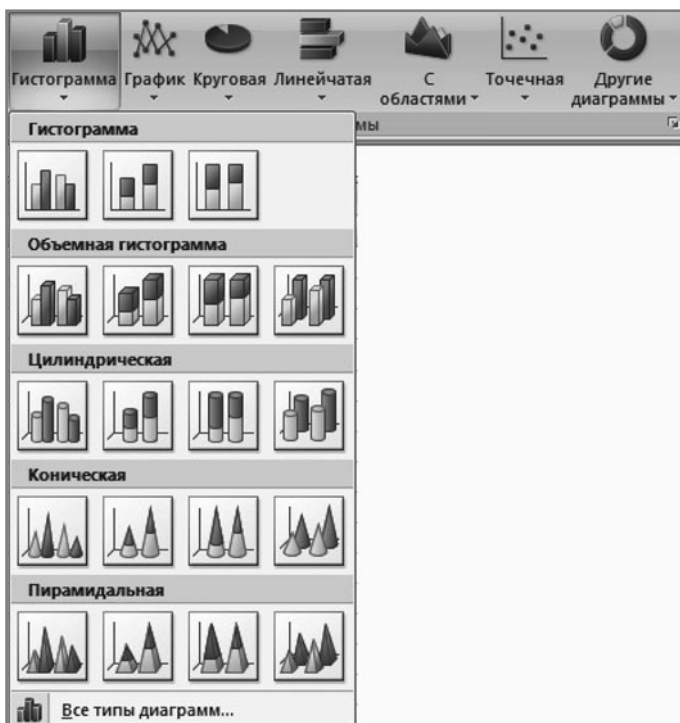


Рис. 2.142. Выбор типа диаграммы с помощью ленты

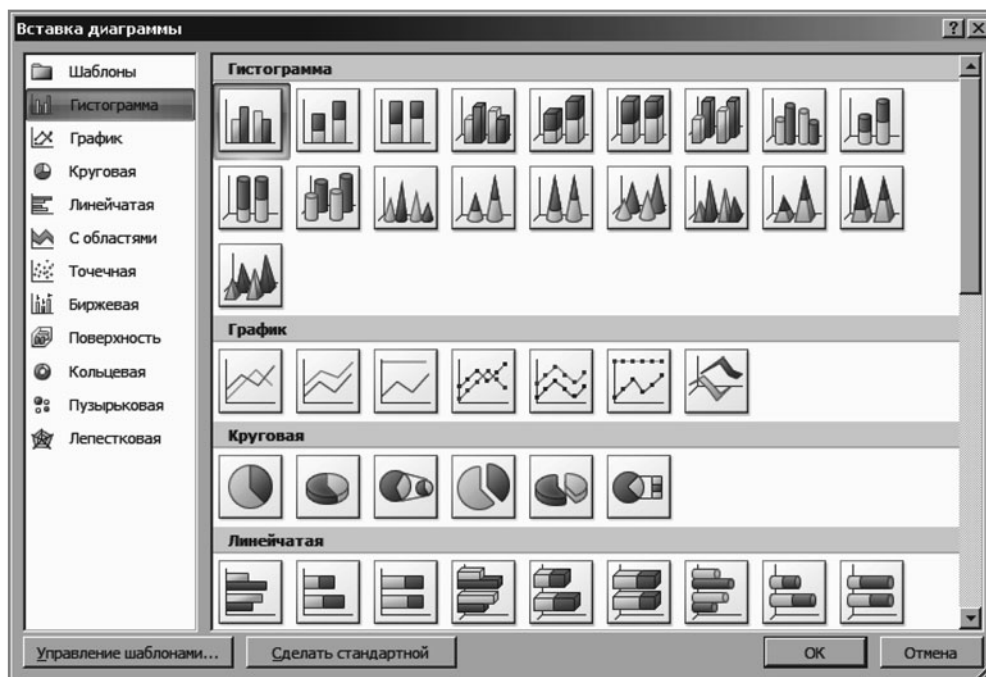


Рис. 2.143. Окно выбора типа диаграммы

Пример 7

Построить диаграмму, изображенную на рис. 2.144, и добавить в нее данные за 3-й квартал. Исходные данные для диаграммы приведены на рис. 2.145.

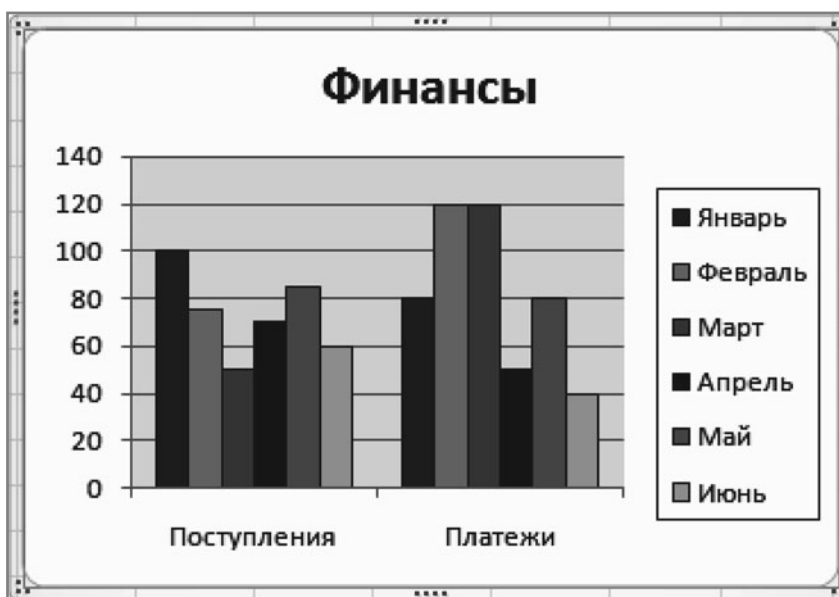


Рис. 2.144. Диаграмма Финансы

	К	L	M	N	O	P	Q	R	S	T
Финансы		Январь	Февраль	Март	Апрель	Май	Июнь	Июль	Август	Сентябрь
Поступления		100	175	150	70	85	60	120	110	90
Платежи		80	120	120	50	90	40	45	35	150
Сальдо		20	-45	-70	20	5	20	75	75	-60

Рис. 2.145. Исходные данные для построения диаграммы

Решение. Для построения диаграммы следует выделить прямоугольный диапазон ячеек, включающий данные, заголовки строк и столбцов (K1:Q3). Затем на вкладке *Вставка* щелкнуть по кнопке *Гистограмма* и выбрать желаемый вид гистограммы. Диаграмма будет расположена по умолчанию на текущем листе.

Добьемся теперь, чтобы диаграмма отображала еще и июльские, августовские и сентябрьские поступления и платежи. Для этого выделим диаграмму — при этом вокруг ее исходных данных на листе появятся цветные границы. Следует мышью расширить синий прямоугольник, охватывающий числовые значения (см. рис. 2.145), до диапазона L2:T3. Одновременно расширится зеленая рамка вокруг заголовков рядов (до блока L1:T1), и диаграмма будет перестроена с добавлением данных.

2.7. ОРГАНИЗАЦИЯ ИНФОРМАЦИОННОЙ СРЕДЫ

2.7.1. Создание и обработка комплексных информационных объектов в виде печатного текста, веб-страницы, презентации с использованием шаблонов

Компьютерная презентация (от *англ.* *presentation* — представление, демонстрация) — последовательность слайдов, содержащих различные виды информации (аудио, видео, графической, текстовой), представленной с применением различных эффектов.

Подготовка презентации

Создание любой презентации обычно включает такие этапы:

- проектирование презентации,
- создание запланированных слайдов,
- установка последовательности и времени демонстрации слайдов презентации.

Процесс *создания презентации* должен обязательно начинаться с *этапа проектирования*. На этом этапе определяется, из каких слайдов должна состоять презентация, тщательно разрабатывается содержимое каждого слайда, продумывается их оформление — создается проект презентации. Слайды не должны быть перегружены ни текстовой, ни графической информацией.

Только после этого начинают непосредственное создание слайдов. Работать над слайдами можно в любом порядке. Последовательность их расположения устанавливается на последнем этапе работы.

В работе над презентацией рекомендуется учитывать следующие основные аспекты.

1. Начинать работу надо со сбора информации по теме презентации, тщательного ее анализа и отбора. На слайды должны попасть только самые значимые данные. В результате анализа должен быть сформирован план подачи материала. Обычно в нем модерируется не более 5–7 пунктов.

2. Желательно, чтобы каждый слайд имел заголовок. На одном слайде должна быть раскрыта одна идея.

3. Следует выбрать необходимый объем текста для размещения на слайде, шрифты и месторасположение текста. Размер шрифта зависит от условий будущей демонстрации, но должен быть не менее 18 пт. На слайде рекомендуют использовать не более тридцати слов и не более пяти пунктов списка. Схемы, диаграммы, таблицы в случае их использования должны быть максимально упрощенными, содержать небольшой объем информации.

4. Если на слайде имеется список, он должен быть параллельным (т. е. первые слова в начале каждого пункта должны находиться в одной и той же грамматической форме — падеже, роде, спряжении и т. д.).

5. При подборе цветов следует учитывать их сочетаемость и воздействие. Цветовая гамма существенно влияет на восприятие презентации, а использование несочетаемых цветов — достаточно распространенная ошибка.

6. Нельзя чрезмерно увлекаться анимацией презентации — избыточные эффекты отвлекают и утомляют. Следует учитывать направленность движений. Так, направление «справа налево», «сверху вниз» оказывают негативное влияние, правильнее задавать движения «слева направо», «снизу вверх».

7. Время показа каждого слайда должно быть тщательно продумано. Рекомендуемая длительность — от двух до пяти минут на слайд.

Начало работы с PowerPoint

Запуск программы PowerPoint аналогичен началу работы с другими продуктами семейства MS Office. Для запуска программы удобно воспользоваться меню Windows *Пуск / Все программы / Microsoft Office / Microsoft PowerPoint*. При запуске программа открывается в так называемом обычном режиме, который позволяет создавать слайды и работать с ними (рис. 2.146).

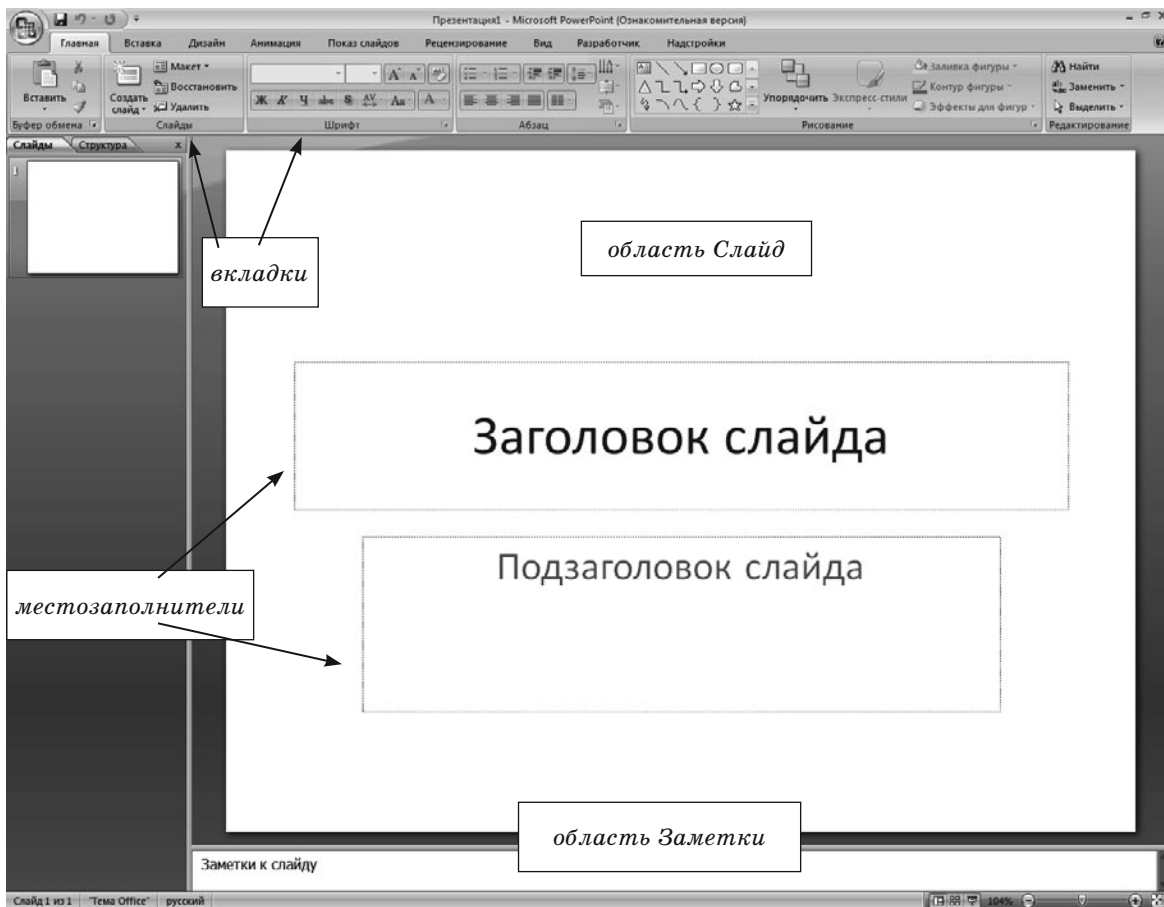


Рис. 2.146. Окно PowerPoint в обычном режиме после запуска программы

В области *Слайд* (рабочее поле) можно работать непосредственно с отдельными слайдами. Под слайдом в PowerPoint подразумевается один кадр презентации. Пунктирные линии на слайде показывают местозаполнители, в которые можно ввести текст или вставить изображения, диаграммы и другие объекты.

Вкладка *Слайды* в левой области окна содержит эскизы всех слайдов презентации. После выбора эскиза соответствующий слайд отображается в области *Слайд*. Эскизы можно перетаскивать мышью, чтобы изменить порядок слайдов в презентации. Кроме того, вкладка *Слайды* позволяет добавлять и удалять слайды.

Размер левой области вкладок можно изменять по ширине, перетаскивая мышью вертикальную границу между вкладками и областью *Слайд* (рис. 2.147).



Рис. 2.147. Изменение ширины области вкладок

Наиболее часто используемые команды вынесены на вкладку ленты *Главная* (рис. 2.148). Это кнопки копирования и вставки объектов, добавления слайдов, изменения макета слайда, форматирования и размещения текста, поиска и замены текста.

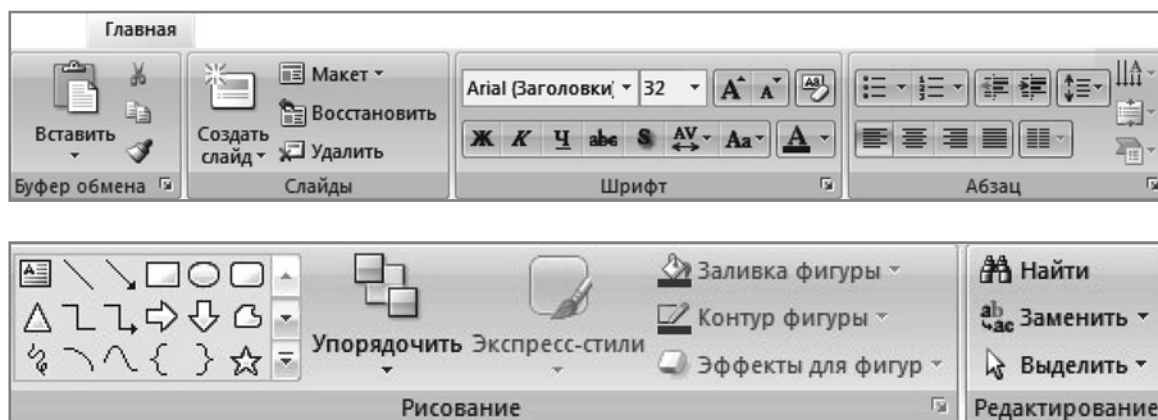


Рис. 2.148. Группы кнопок вкладки Главная

Лента содержит и другие вкладки, например:

- *Вставка* — предназначена для вставки на слайд таблиц, рисунков, схем, диаграмм, текстовых полей, звуков, гиперссылок и колонтитулов.
- *Конструктор* — позволяет выбирать общий вид слайда, фоновый рисунок, шрифты и цветовую схему.
- *Анимация* — включает возможности выбора и настроек анимационных эффектов.
- *Показ слайдов* — позволяет просматривать все слайды или часть презентации, записывать речевое сопровождение и др.





При работе в приложении PowerPoint приходится часто менять режим просмотра. Эту операцию можно быстро выполнить с помощью кнопок  Обычный,  Сортировщик слайдов и  Показ слайдов — эти кнопки находятся в правом нижнем углу окна (рис. 2.149). Они являются частью строки состояния, которая содержит к тому же ползунок масштаба и кнопку  Вписать слайд, которая вписывает слайд обратно в размеры окна после изменения масштаба.



Рис. 2.149. Кнопки режимов просмотра и масштабирования

С помощью трех кнопок, расположенных в левой части строки состояния, можно переходить в наиболее часто используемые режимы просмотра.

Режим сортировщика слайдов (рис. 2.150) отображает эскизы всех слайдов, что удобно для их сортировки и общего взгляда на всю презентацию в целом. Масштаб отображения эскизов регулируется с помощью ползунка в строке состояния — одним движением ползунка можно вывести в окне большее либо меньшее число эскизов соответственного меньшего либо большего размера.

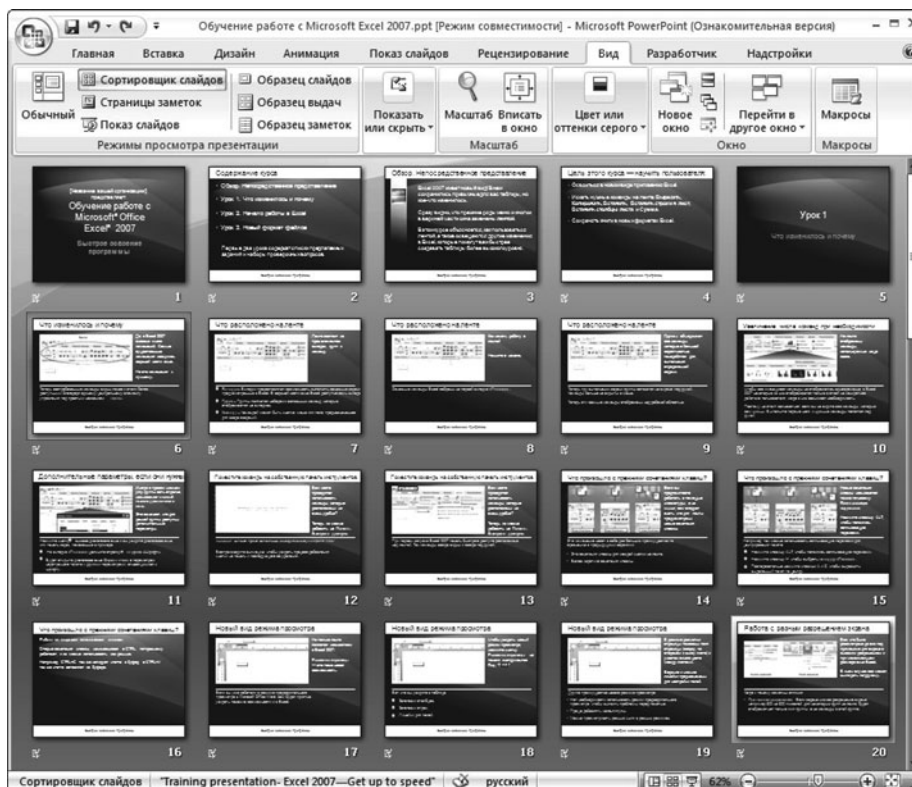



Рис. 2.150. Режим Сортировщик слайдов

Режим Показ слайдов запускает демонстрацию презентации на экране — поочередную смену слайдов с заданными эффектами (звуковыми, анимации и т. п.).

Работа с файлами


Как и при работе с любой другой программой, рекомендуется при создании презентации сразу же сохранить ее в файле, присвоив ей имя, а затем во время работы выполнять повторное сохранение.

Для сохранения, для открытия, создания и других подобных операций с файлами, служит меню кнопки  *Office*. Команды этого меню аналогичны командам MS Word и MS Excel.

Презентация может быть сохранена в формате, позволяющем проводить в дальнейшем ее редактирование (формат .pptx) либо в варианте, когда возможен только лишь ее просмотр в режиме показа слайдов (формат .pps). Для этого в меню предназначены соответственно команды *Презентация PowerPoint* и *Демонстрация PowerPoint*.

Создание презентации

Новая презентация создается автоматически при запуске программы. Она содержит один слайд с двумя *прототипами* (позициями для заполнения определенными данными) — для текстов заголовка и подзаголовка. Однако можно ввести в эти позиции и любую другую информацию. Удалить объект можно клавишей Delete, предварительно выделив щелчком мыши по его границе.

Схема расположения прототипов на слайде называется *макетом*, или *разметкой*. Можно поменять макет с помощью кнопки  *Макет* группы *Слайды* вкладки ленты *Главная* (рис. 2.148). Соседняя кнопка *Создать слайд* (рис. 2.151) позволяет добавить в презентацию новый слайд, выбрав для него макет.

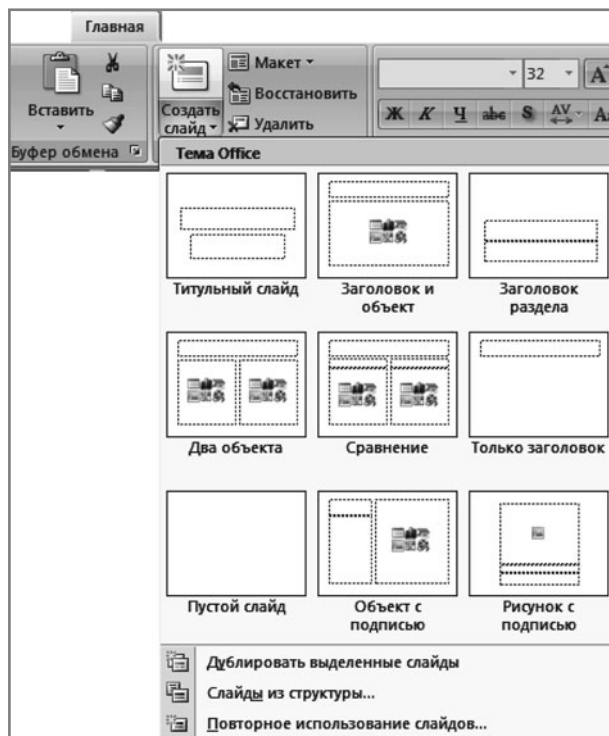


Рис. 2.151. Варианты макетов слайдов

В слайдах, созданных с помощью разметки, место каждого объекта обозначается светлыми пунктирными линиями, внутри располагается подсказка.

Например, для макета *Два объекта* (рис. 2.152) на слайд вставляется три прототипа. Один из них — область заголовка слайда; в два оставшихся можно ввести текст или вставить объекты с помощью полупрозрачных кнопок в центре их областей. Эти кнопки и слова «Заголовок слайда», «Текст слайда» будут удалены при вводе данных. Возможные объекты для вставки: таблица, рисунок, диаграмма, рисунок SmartArt, изображение из файла, видеоклип или звук.



Рис. 2.152. Макет слайдов *Два объекта*

Имеющиеся слайды можно копировать для создания аналогичных им по содержанию и макету. Для перемещения слайдов удобно использовать вкладку *Слайды* или режим сортировщика. Для удаления слайда надо выделить его и нажать клавишу Delete.

Вставка объектов на слайд

На слайдах могут размещаться различные объекты: списки, тексты, рисунки, графики, таблицы, рисунки SmartArt, диаграммы, автофигуры, видеофрагменты и др. Каждый объект можно выделить, переместить, скопировать, удалить, а также изменить его размер или содержимое. Эти действия выполняются для всех объектов одинаково — так же, как и для объектов в других приложениях MS Office.

Для вставки объектов на слайд служит вкладка ленты *Вставка* (рис. 2.153).

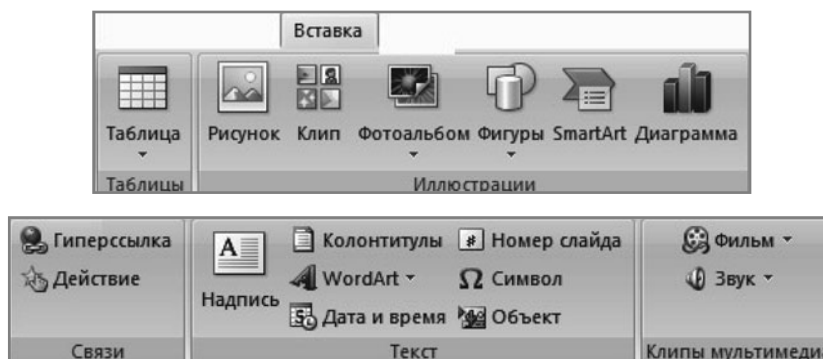






Рис. 2.153. Группы кнопок вкладки ленты *Вставка*

Кроме того, PowerPoint позволяет вставить в слайды их номера (кнопка  *Номер слайда*), а также видео- и звуковые файлы (кнопки  *Фильм* и  *Звук* соответственно). После выбора мультимедиа-файлов нужно указать, должно ли воспроизведение видео и звука стартовать сразу с начала показа слайда или по щелчку мыши на значке.


После вставки мультимедийных данных на слайде будет размещен соответствующий значок  для звука и прямоугольник черного цвета для видео. Прямоугольник можно заполнить каким-либо изображением с помощью команды его контекстного меню *Изменить рисунок*.

Текст на слайд можно вводить в области прототипов. Для этого достаточно установить курсор в слова прототипа «Текст (или Заголовок) слайда» (слова при этом исчезнут с экрана) и начать набор. Можно вставить текст из буфера обмена. Если текст не уместается в рамке, то по мере ввода новых знаков шрифт и межстрочное расстояние будут уменьшаться до нужного размера. Размеры рамок можно изменять, растягивая мышью за угловые и срединные маркеры.

Текст может быть отформатирован с помощью инструментов на вкладке ленты *Главная* (рис. 2.148), а также мини-панели форматирования. Работа с этими инструментами аналогична работе в текстовом процессоре MS Word.

Оформление презентации

PowerPoint позволяет задать общий вид презентации с помощью множества тем. **Тема** — набор элементов оформления, придающих единообразный внешний вид слайдам, используя конкретные сочетания цветов, шрифтов и эффектов.

Выбрать тему можно с помощью кнопки  *Дополнительные параметры* в группе *Темы* вкладки *Дизайн* (рис. 2.154). При наведении указателя мыши на предполагаемую тему срабатывает функция предпросмотра — текущий слайд представляется в соответствующем оформлении.

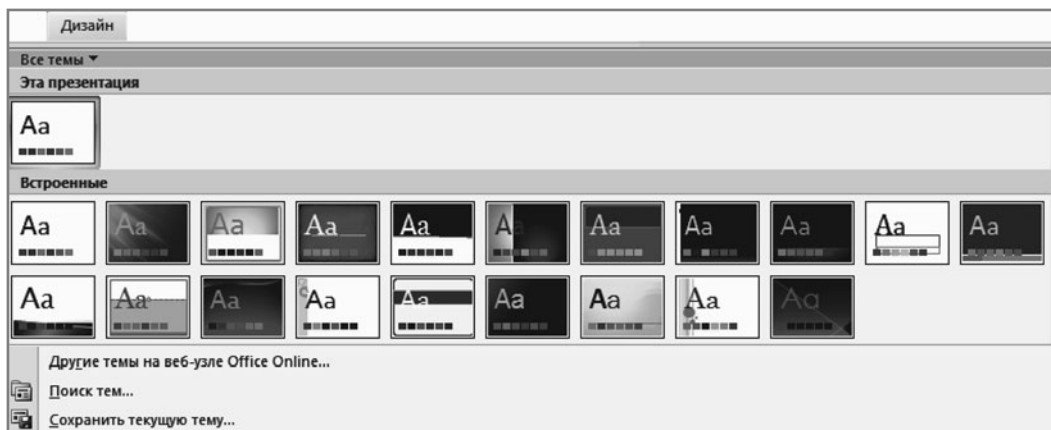





Рис. 2.154. Выбор темы презентации

По умолчанию PowerPoint применяет темы ко всей презентации. Чтобы изменить внешний вид только некоторых слайдов, надо выделить их (например, на

вкладке *Слайды* с нажатой клавишей Ctrl), а затем выбрать в контекстном меню выбранной темы команду *Применить к выделенным слайдам*.

Любую тему можно настроить — изменить используемые цвета, шрифты, линии и эффекты заливки. Для этого используются выпадающие списки кнопок

 *Цвета*,  *Шрифты*,  *Эффекты*.

Использование анимации

Анимация служит для поэтапной выдачи информации и придания презентации большей выразительности.

PowerPoint позволяет настроить смену слайдов с использованием *анимационных эффектов* при переходе от одного слайда к следующему. Среди эффектов имеются стандартные — затухания, растворения, обрезания и стирания — и более сложные переходы — например, колеса и шахматные доски.

Анимация может быть применена как к слайдам в целом, так и к отдельным их элементам.

Чтобы настроить смену слайдов, надо на вкладке ленты *Анимация* в группе *Переход к этому слайду* выбрать один из вариантов перехода (рис. 2.155). При этом работает функция предпросмотра — эффект анимации можно увидеть на текущем слайде.

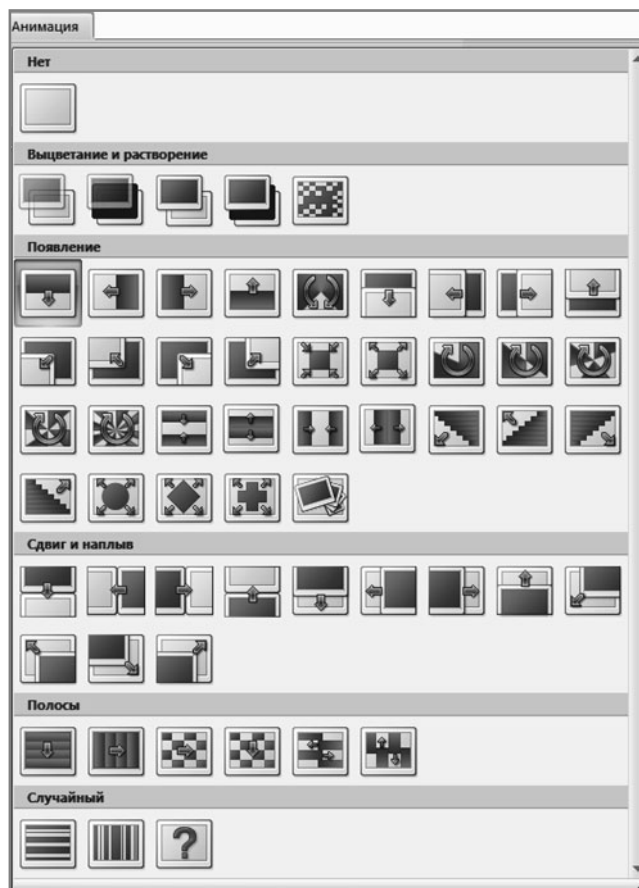


Рис. 2.155. Варианты анимированного перехода к слайду

Можно настроить анимацию для отдельных объектов слайда. Большинство эффектов позволяет воспроизводить звук во время анимации или запускать анимацию фрагментов текста (например, отдельных букв, слов или абзацев). Для элемента можно использовать сразу несколько эффектов — например, вылет элемента списка на слайд, а затем вылет за пределы слайда.

Настройка анимации отдельных элементов слайда производится в области задач *Анимация* (рис. 2.156) (открывается кнопкой-переключателем *Настройка анимации* на ленте).

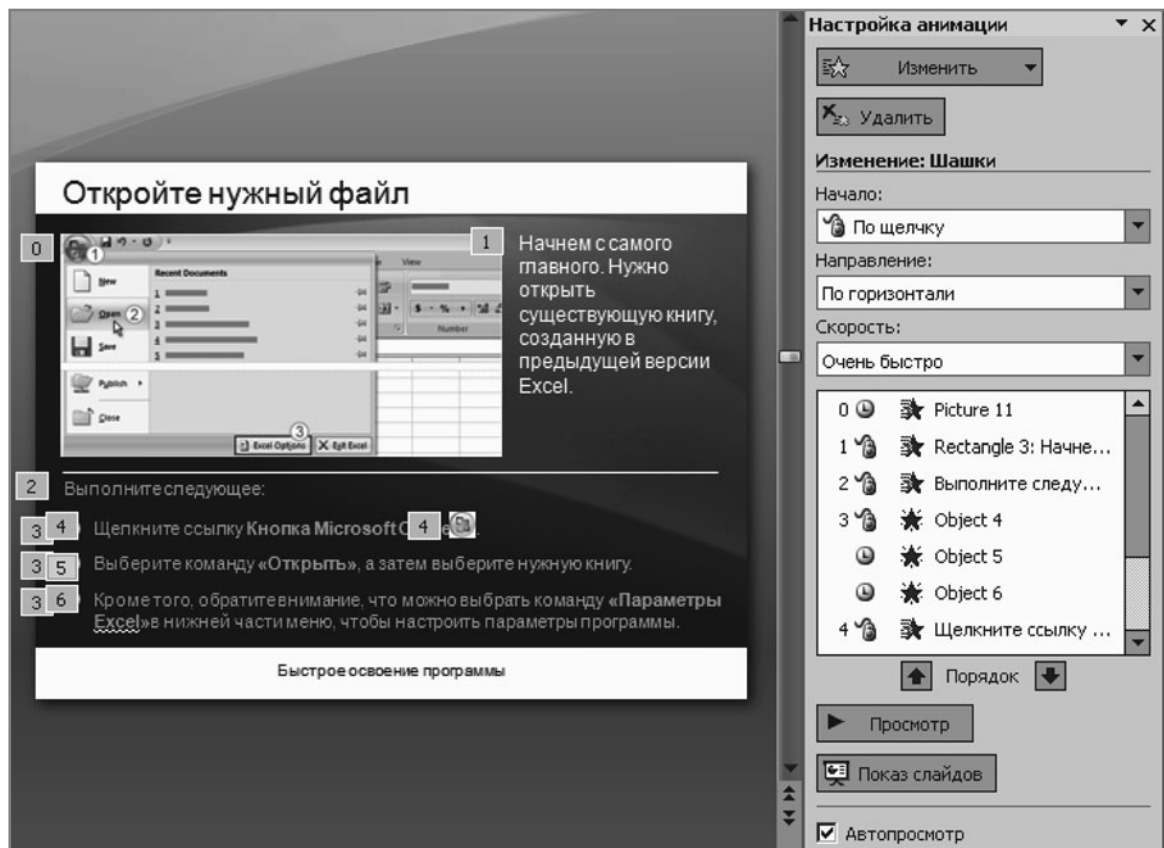


Рис. 2.156. Область задач *Анимация* и числовые метки в области слайда

Следует по очереди выделять объекты слайда и задавать для них в панели параметры анимации.

Порядок выделения объектов соответствует будущему порядку применения эффектов. В области слайдов объекты будут помечаться **числовыми метками**, соответствующими заданному порядку (см. рис. 2.156). Метки видны только при открытой панели *Анимация*.

Для задания **эффекта** (действия) на панели анимации используется выпадающий список кнопки *Добавить эффект* (рис. 2.157). Команда *Другие эффекты* откроет полный список возможных действий (рис. 2.158).

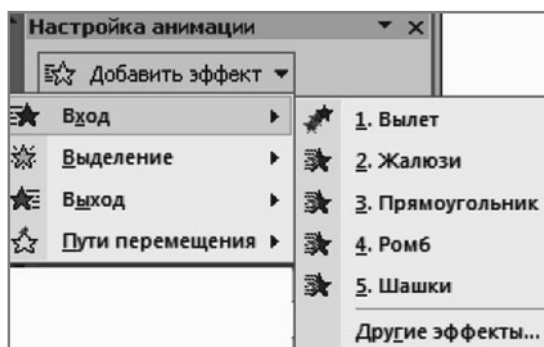




Рис. 2.157. Выбор типа анимации

Настройки зависят от типа выбранной анимации, однако почти для всех ее типов надо указать, когда начинать действие анимации (по щелчку мыши , одновременно с предыдущим эффектом либо сразу после него ) и ее скорость.

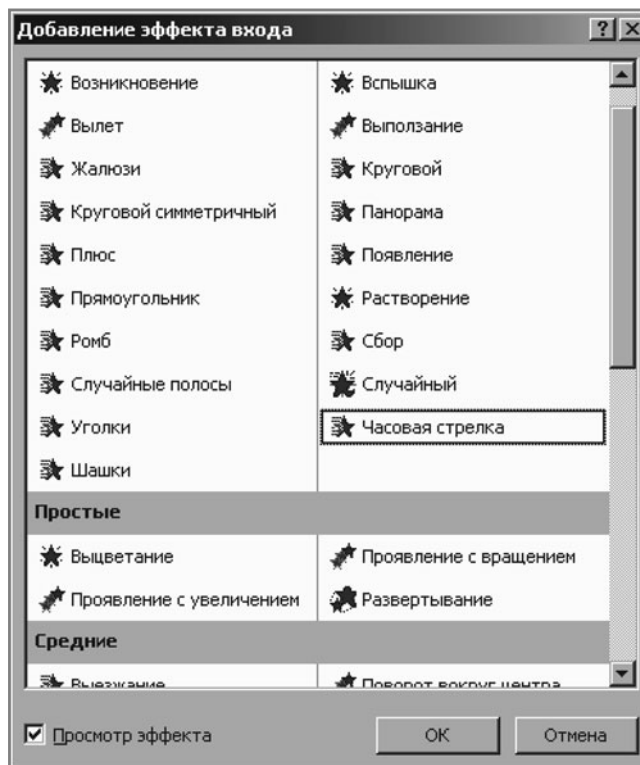


Рис. 2.158. Список эффектов анимации типа Вход

Чтобы изменить **порядок следования** эффектов, в панели анимации имеются кнопки со стрелками. Кнопка *Просмотр* внизу панели (см. рис. 2.156) позволяет просмотреть всю последовательность эффектов анимации всех объектов текущего слайда.

Для **изменения** типа анимации и ее **удаления** служат кнопки сверху панели *Изменить* и *Удалить*.

Показ слайдов

Показ слайдов возможен в любой момент создания презентации.

Для просмотра презентации на экране компьютера в том виде, в каком она будет представлена зрителям, надо выбрать одну из команд в группе *Начать показ слайдов* вкладки *Показ слайдов*:

- *С начала* (с первого слайда);
- *С текущего слайда* (со слайда, отмеченного на вкладке *Слайды*).

Запуск демонстрации с самого начала стартует также по нажатию клавиши F5.

Переходы к следующему слайду осуществляются по щелчку мыши. Чтобы вернуться в обычный режим, в любой момент можно нажать клавишу Esc.

Можно сохранить подготовленную презентацию в формате, который будет открывать ее сразу в режиме показа слайдов. Для этого при сохранении файлов надо указать тип файла *Демонстрация PowerPoint*. Файл будет иметь расширение .ppsx.

Настройка времени презентации

После того, как презентация подготовлена, можно настроить время **автоматической смены слайдов** при будущей демонстрации. Важно предусмотреть достаточно времени и правильно его распределить. Для этого следует провести репетицию презентации.

Чтобы запустить процесс настройки временных параметров, на вкладке *Показ слайдов* в группе *Настройка* надо нажать кнопку *Настройка времени*. Отобразится панель инструментов *Репетиция* и в ее поле *Время слайда* начнется отсчет времени презентации (рис. 2.159).

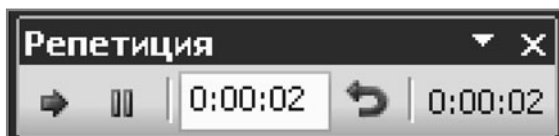





Рис. 2.159. Панель инструментов Репетиция

Кнопка  *Далее* переводит показ к следующему слайду, кнопка  *Пауза* работает как переключатель — приостанавливает и затем продолжает показ,  *Повторить* повторяет запись времени для текущего слайда.

После того как будет установлено время для всех слайдов, откроется режим сортировщика слайдов, в котором отображено время демонстрации каждого слайда.

Интерактивность презентаций

PowerPoint предоставляет возможность настроить поведение презентации в зависимости от действий пользователя. К средствам, способным реагировать на действия пользователя, относятся гиперссылки, управляющие кнопки и триггеры*.

* *Триггеры* — объекты, щелчок на которых приводит к запуску анимации.

Удаленные — для удаленных писем. Как и файлы в Windows, удаленные из ящика письма сначала помещаются в эту папку, и лишь удаление из папки *Удаленные* действительно навсегда уберет сообщение из ящика.



Рис. 2.161. Внешний вид почтового ящика

Возле каждой папки указано общее количество писем в ней и количество непрочитанных писем.

Щелчок по названию какой-либо папки выведет в основной части окна ее содержимое. Его можно упорядочить по столбцам *Отправитель*, *Дата*, *Размер* (щелчком по их заголовкам). По умолчанию список отсортирован по датам, самые свежие письма находятся сверху. Непрочитанные письма выделены жирным шрифтом. В начале (или конце) каждой строки имеется ряд значков — например, скрепка обозначает, что к письму прикреплены файлы, флажок или звездочка позволяют пользователю пометить важность отдельных сообщений.

Для изменений в структуре папок (чаще всего требуется создать новые папки, чтобы рассортировать полученные письма) служит ссылка под списком папок *Управление папками*.

Создание писем

Для создания нового письма предназначена ссылка вверху окна *Написать письмо*. Воспользовавшись ею, пользователь увидит в окне заготовку будущего письма (рис. 2.162).

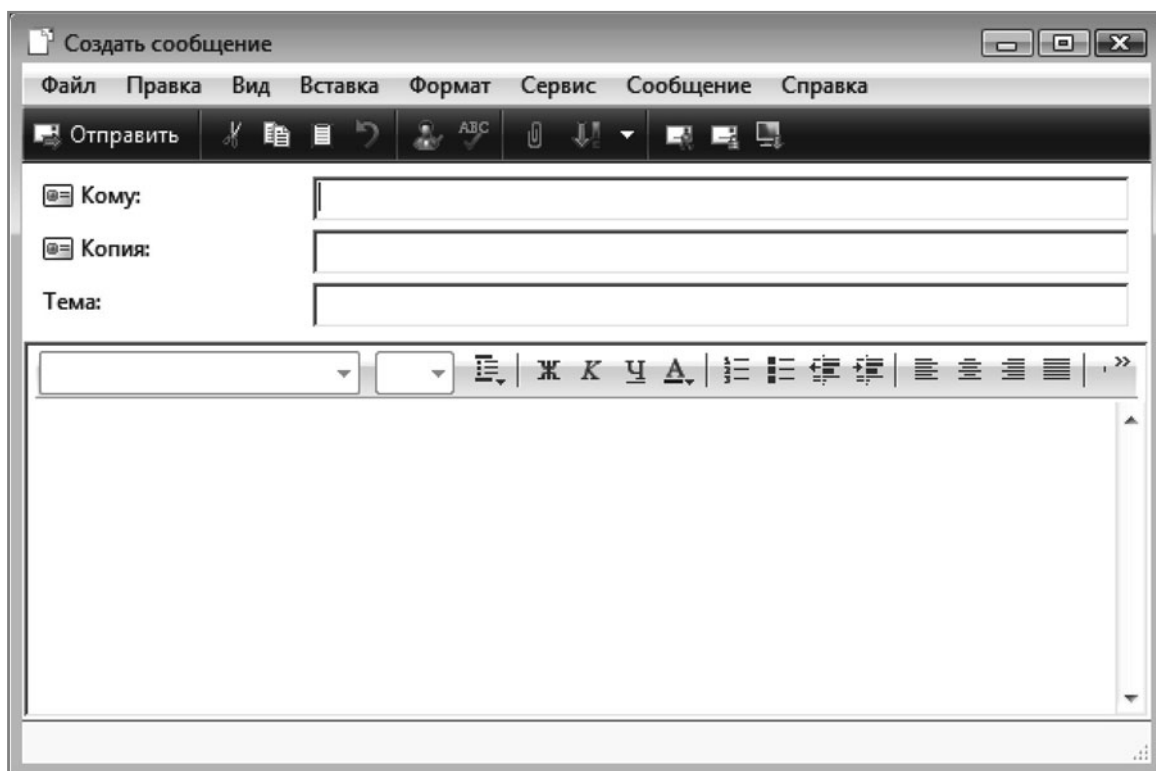


Рис. 2.162. Бланк нового сообщения

В бланке письма надо заполнить **адресные поля**. Этих полей два: *Кому* и *Копия*. В первом из них обычно указывают основных получателей письма, в другом — второстепенных адресатов, от которых не требуется прямой реакции на сообщение, но которые должны быть в курсе событий. Можно, однако, и перечислить все адреса в одном поле через точку с запятой.

Адрес электронной почты состоит из трех частей: имени (или *идентификатора*) пользователя, разделителя @ («коммерческое at», или «обезьянка», или «собачка») и адреса почтового сервера. Идентификатор может содержать только латинские буквы, цифры и знак подчеркивания (некоторые серверы допускают вместо подчеркивания дефис, иные разрешают из дополнительных символов только точку).

При щелчке по кнопкам *Кому* и *Копия* можно выбрать адреса из **адресной книги**.

В поле *Тема* следует указать **тему письма**. При этом сформулировать ее надо кратко и емко.

В панели **текста сообщения** следует набрать текст письма и при желании отформатировать его с помощью панели форматирования.

Для **вложения файлов** в письмо служит кнопка *Файл*. Прикрепляя вложение в электронные письма, необходимо помнить, что некоторые почтовые серверы накладывают ограничения на размер пересылаемых файлов.

Если работа с сообщением не закончена и его надо **временно сохранить**, чтобы дописать и отправить позднее, следует выбрать команду *Сохранить*, после чего письмо будет помещено в папку *Черновики*. Впоследствии можно будет завершить работу над этим письмом и отослать его.

Подготовив письмо, следует воспользоваться встроенным средством **проверки орфографии** — неграмотное письмо с ошибками или опечатками создает негативное впечатление о его авторе. Для вызова механизма проверки можно щелкнуть по кнопке проверки орфографии на панели инструментов.

После того как сообщение подготовлено, его следует **отправить адресату** щелчком по кнопке *Отправить*

Действия над письмами

Для **прочтения письма** надо щелкнуть на нем в списке писем. Внизу письма (рис. 2.163) предлагается небольшое прямоугольное поле для быстрого ответа. Как только курсор будет установлен в него, ниже появится кнопка *Отправить письмо*. Можно написать ответ и отослать его.

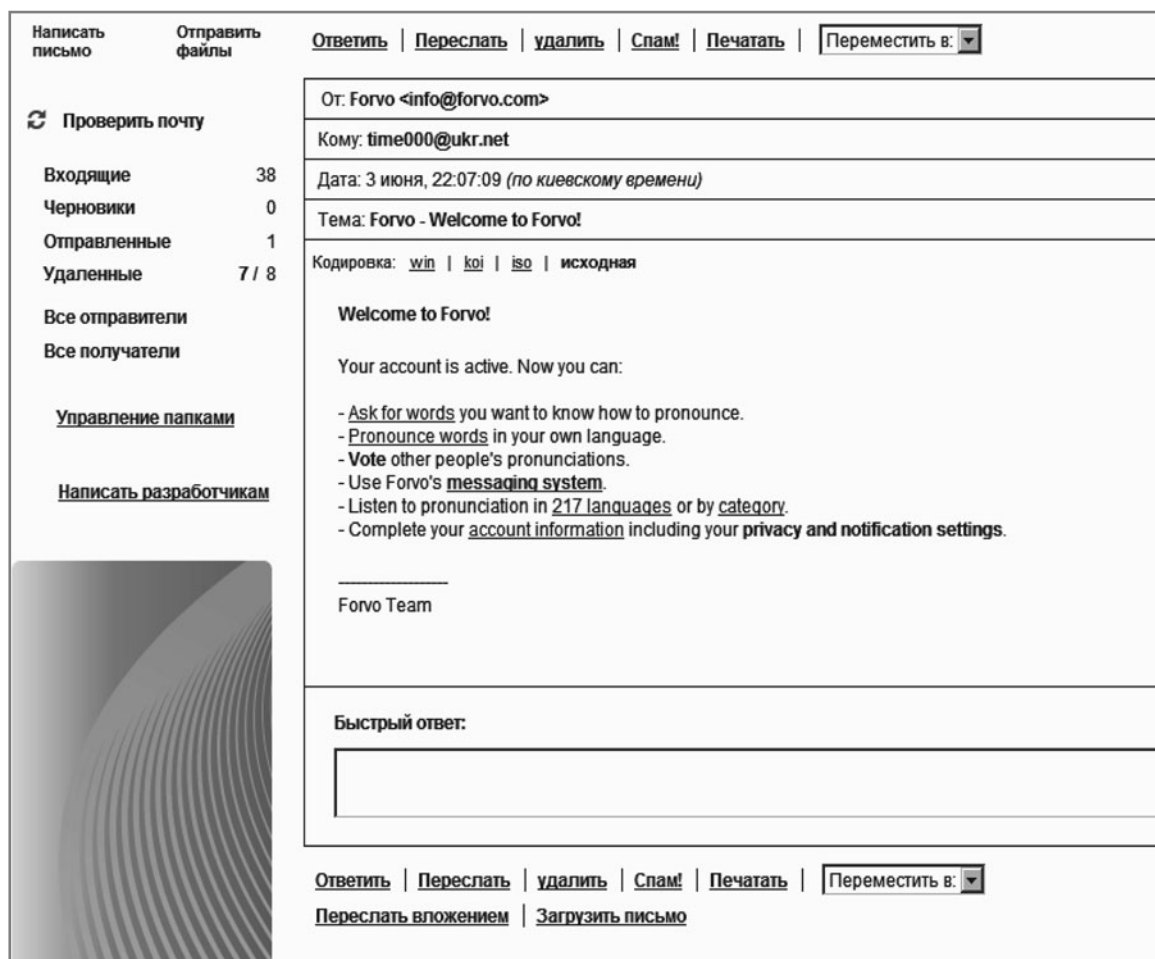


Рис. 2.163. Чтение сообщения

Если письмо содержит **вложенные файлы**, для их просмотра и сохранения на своем компьютере надо нажать кнопку *Скачать*.

Все полученные письма и файлы почтовые серверы проверяют на наличие вируса. Тем не менее, прежде чем открывать полученные из Интернета файлы,

их рекомендуется еще раз проверить на отсутствие вирусов. Для этого следует воспользоваться установленной на компьютере антивирусной программой.

Полученное письмо можно **переслать** другому адресату, воспользовавшись кнопкой *Переслать*. В появившемся подготовленном письме надо указать адресата, которому будет перенаправлено письмо.

Для **ответа** на полученное сообщение имеется кнопка *Ответить*. При нажатии на нее открывается окно создания сообщения, в котором уже заполнено поле адреса, а в теме письма указана тема исходного сообщения, перед которой вставлена пометка *Re:* (сокращение от *англ.* Reply — ответ). Дальнейшие действия пользователя такие же, как при написании нового сообщения.

Ненужные более сообщения следует **удалять** из папок — для этого нужно выделить сообщение в верхней части окна программы и нажать кнопку Удалить. Письмо сначала попадет в папку *Удаленные*, откуда можно удалить его навсегда. Не удаленные пользователем из папки *Удаленные* письма по прошествии времени (через месяц) будут удалены автоматически.

Для **перемещения** письма в другую папку надо выбрать ее название из выпадающего списка *Переместить в*.

Если действия пересылки, удаления, перемещения и др. надо выполнить сразу с несколькими письмами, следует в основном окне почтового ящика пометить их флажками (слева от темы письма), а затем выбрать команды, которые расположены вверху и внизу списка писем (рис. 2.161).

По окончании работы с почтовым ящиком надо «закрыть» его, нажав ссылку *Выйти*. Она неприметна и расположена в правом верхнем углу окна.

2.7.3. Сохранение информационных объектов из компьютерных сетей и ссылок на них для индивидуального использования (в том числе из Интернета)

World Wide Web (WWW, W3, Всемирная паутина или просто Web) — самый распространенный вид интернет-сервиса. В основе его лежит понятие *гипертекста* — текстового документа, содержащего в себе ссылки на другой документ или страницу (*гиперссылки*). Идея гипертекста была разработана Тедом Нельсоном в 1969 году и с тех пор претерпела существенное развитие. Сегодня гипертекстовый документ правильнее назвать гипермедийным документом, поскольку он содержит в себе не только текст, но и графические изображения, звуки, анимацию, фрагменты кодов программ и т. д.

Однако основные идеи, заложенные при его создании, остались прежними: составление гипердокументов из отдельных страниц (которые могут содержаться в разных файлах и даже на разных компьютерах) и возможность перемещения между ними по гиперссылкам.

Адреса веб-ресурсов

Строка символов, которая однозначно идентифицирует любой ресурс в сети Интернет, называется **URL** (Uniform Resource Locator — универсальный локатор ресурсов).

буквы ЕВ. Вслед за ним через косую черту должен быть указан полный путь к файлу. С учетом косых — это буквы И (папка), затем ЖА (файл).



Ответ: ГЗЕВИЖА.

Работа в браузере


Для **просмотра** веб-документов и перемещения между ними используются специальные программы — **браузеры** (от *англ.* to browse — просматривать), или обозреватели. Существует несколько десятков браузеров. Наиболее распространенными на сегодняшний день являются программы Mozilla FireFox (разрабатывают компания Mozilla и множество добровольцев, поскольку браузер является свободно распространяемым ПО), Google Chrome (разработанный компанией Google), Opera (разработка компании Opera Software), Internet Explorer (входящий в ОС Windows, разработка фирмы Microsoft). Практически все браузеры являются бесплатными.

Принципы работы в любом браузере схожи и очень просты. Различаются браузеры степенью защиты, функциональностью, поддержкой различных форматов, технологий и протоколов, возможностью расширения.

Чтобы отобразить содержимое какой-либо страницы в окне браузера, надо ввести ее **адрес** в строку адреса, расположенную сверху окна. После ввода адреса и нажатия клавиши Enter начнется загрузка страницы, а затем ее содержимое будет отображено в рабочей зоне окна браузера.



Если в процессе отображения страницы произошел сбой или надо **повторить вывод** страницы на экран по другой причине (например, данные на странице часто обновляются, как курсы акций на бирже или прогнозы погоды на некоторых погодных серверах), то рядом со строкой адреса для этой цели имеется кнопка  *Обновить* (можно также нажать клавишу F5). При необходимости **остановить отображение** страницы можно воспользоваться кнопкой  *Остановить* или клавишей Esc.

Для **перехода** с просмотренной страницы на другую надо воспользоваться одной из ее гиперссылок. Ссылки бывают текстовыми (гипертекстовыми), тогда их текст выделен цветом и подчеркиванием; бывают графическими — это кнопки с надписями, фотографии, псевдообъемные графические элементы. Часто разные виды гиперссылок дублируют друг друга. На странице могут одновременно присутствовать как текстовые, так и графические элементы с одинаковым контекстом.

Отличить ссылку от любого другого элемента достаточно просто — курсор мыши при наведении на гиперссылку принимает вид руки с вытянутым указательным пальцем — .

Чтобы перейти по ссылке на другую страницу, надо щелкнуть по ней.

Можно вызвать **контекстное меню** гиперссылки (щелчком правой кнопки мыши) — в нем значительно больше способов перехода по ссылке (рис. 2.165). В частности, ее можно открыть в том же самом окне (команда *Открыть*), в новом окне или в новой вкладке того же окна (в этих случаях новое окно программы или новая вкладка будут раскрыты автоматически). Команды в нижней части меню зависят от дополнительного ПО, установленного на компьютере.

После некоторого передвижения по разным сайтам может возникнуть потребность **вернуться** на одну из посещенных ранее страниц. С этой целью слева от адресной строки расположены кнопки навигации. Одна из них —  *Назад* — возвращает на предыдущую страницу (так же как и нажатие клавиши Backspace). Другая становится доступной только после хотя бы однократного возвращения назад по маршруту просмотра страниц и позволяет вернуться по этому пути —  *Вперед*. Выпадающий список рядом с этими кнопками содержит список посещенных страниц — можно выбрать любую из них для возврата к ее просмотру.

Чтобы **выделить** фрагмент содержимого веб-страницы, достаточно обвести его мышью (хотя некоторые разработчики закрывают такую возможность). Выделенный фрагмент можно **скопировать в буфер обмена** Windows для последующей его вставки в какое-либо иное приложение — стандартными для Windows комбинациями клавиш либо командой *Копировать* в контекстном меню или в меню *Правка*.

Для **печати** веб-страницы (всей либо ее части) предназначена команда *Печать* в контекстном меню и в меню *Файл*. Там же имеются команды *Параметры страницы* (для настройки печати) и *Предварительный просмотр*, позволяющая оценить, как веб-страница будет выглядеть на бумаге.

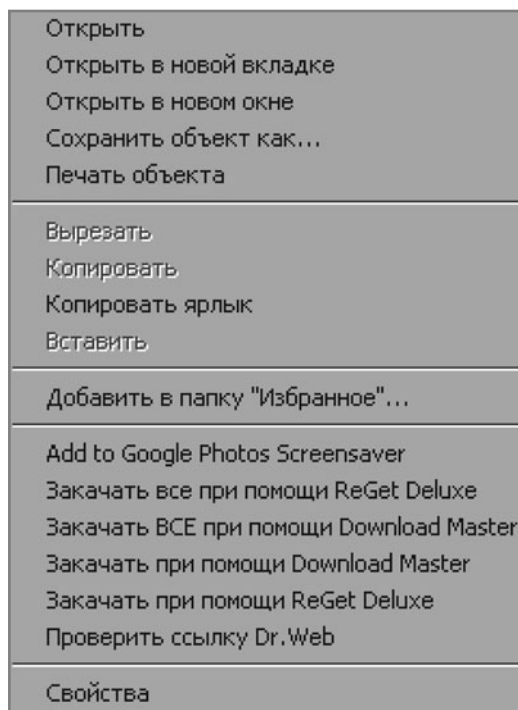


Рис. 2.165. Контекстное меню гиперссылки

Сохранение веб-страниц

Содержимое веб-страницы можно **сохранить в файле** (команда меню *Файл/Сохранить*). В стандартном окне сохранения файла предстоит выбрать формат сохранения. Браузер предлагает несколько вариантов сохранения веб-страницы (рис. 2.166).

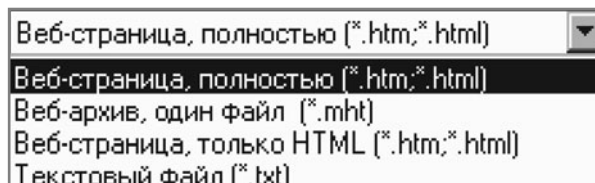


Рис. 2.166. Форматы сохранения веб-страницы

Вариант *Веб-страница полностью* сохранит в файле текстовое содержимое страницы и ее разметку (позиции, где размещены рисунки, кнопки, линии,

таблицы и т. п.). Файл будет иметь расширение `.html` или `.htm`. Сами встроенные элементы будут сохранены каждый в отдельном файле и собраны в папку с таким же названием, как и первый файл (рис. 2.167). Этот файл и эта папка неразрывно связаны. Если переименовать или удалить один из них, связь будет разорвана, и в дальнейшем страница будет отображаться некорректно (без графических и иных элементов). При копировании или перемещении данных следует захватывать и файл, и соответствующую ему папку (впрочем, Windows делает это обычно автоматически).

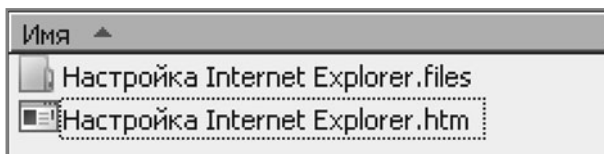


Рис. 2.167. Веб-страница, сохраненная в формате «Веб страница, полностью»

Вариант *Веб-страница, только HTML* сохраняет лишь описанный выше файл и разметку, все внедренные объекты не сохраняются.

Можно преобразовать текст в обычный текстовый формат и сохранить его (только текст!) в формате *Текстовый файл (.txt)*.

Имеется возможность (если не запретили разработчики страницы) сохранить отдельно какой-либо понравившийся элемент веб-страницы — например, рисунок. Для этого в контекстном меню элемента надо выбрать команду *Сохранить рисунок как*.

Закладки

Адреса наиболее ценных или просто интересных веб-страниц желательно сохранить, чтобы впоследствии иметь возможность просмотреть их повторно. С этой целью в браузере существуют *Закладки* (или *Избранное*) — коллекция ссылок, собранных пользователем.

Чтобы воспользоваться любой сохраненной ссылкой, вместо ввода адреса в адресную строку надо всего лишь выбрать из меню *Закладки* (*Избранное*) название желаемой страницы (рис. 2.168).

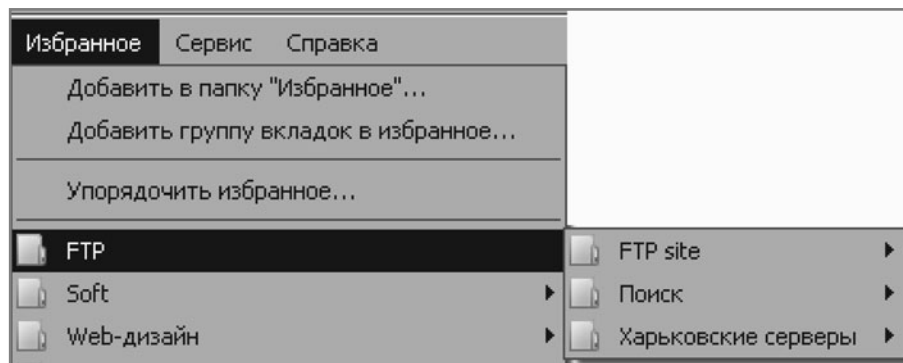


Рис. 2.168. Выбор адреса из Избранного

Для добавления адреса понравившейся страницы надо в момент ее отображения в окне программы вызвать команду меню *Избранное/Добавить в папку «Избранное»* (рис. 2.169).

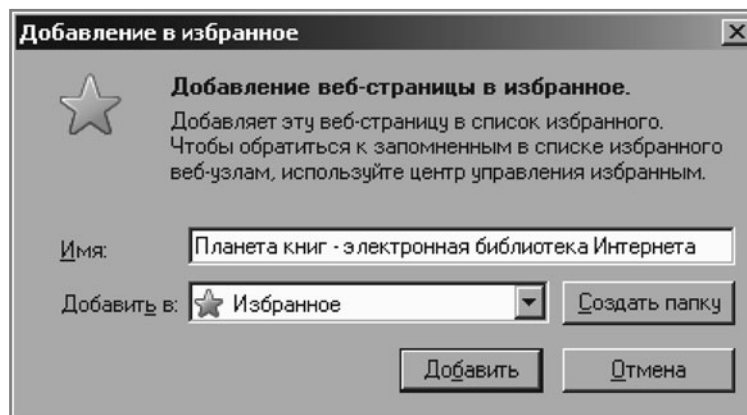


Рис. 2.169. Сохранение адреса в Избранном

Для упорядочения закладок их следует группировать. Для этого при добавлении закладки в меню *Избранное* надо нажать кнопку *Создать папку* и указать название подменю (подпапки). Тогда при добавлении закладки надо будет выбрать в поле *Добавить в* название созданной папки.

2.7.4 Организация информации в среде коллективного использования информационных ресурсов. Примеры организации коллективного взаимодействия: форум, телеконференция, чат

Чат (англ. chat — болтовня, беседа, разговор) — средство общения пользователей по сети в режиме реального времени.

Существует несколько разновидностей чатов. Веб-чаты представляют собой обычную веб-страницу, где можно прочесть последние несколько десятков фраз, написанные участниками чата. Страница чата автоматически обновляется с заданной периодичностью. Примером такого чата может служить сервис Чат!Mail.ru.

С развитием технологий и структуры Сети появилась возможность общения пользователей не только посредством традиционных чатов, но также и с помощью микрофона — по сути, аналог телефонной связи через Интернет. Наиболее популярной программой на сегодняшний день в этом плане является мессенджер **Skype**.


Skype предоставляет возможность бесплатного общения для клиентов своей сети и разрешает совершать платные звонки (в том числе и международные) и на стационарные или мобильные телефоны. Размер оплаты во много раз ниже, чем у стационарных или мобильных операторов.

Программа весьма проста в настройке и использовании — достаточно наличия широкополосного соединения с Интернетом и звуковой аппаратуры — например, микрофона и динамиков (или наушников).

Кроме голосового общения, программа позволяет использовать видеосвязь, обмениваться текстовыми сообщениями, пересылать файлы (эти сервисы бесплатны), использовать голосовую почту и переадресацию телефонных звонков (платные услуги). Чтобы использовать в программе возможности видеосвязи через Интернет, следует обзавестись веб-камерой.

При первом запуске предстоит зарегистрироваться в системе. Затем проверить работу всех устройств с помощью поочередного нажатия кнопок в стартовом окне.

После того как настройка аудио и видеоустройств завершена, можно (и нужно) сделать тестовый звонок. Для этого надо выбрать в левой части окна Skype контакт *Echo/Sound Test Service* и нажать зеленую кнопку (справа) *Позвонить*. Автоответчик тестового центра Skype предложит произнести несколько слов в микрофон и ретранслирует их обратно — если запись слышна, значит, звуковое оборудование в порядке.

Во время работы программы, в трее (системной панели Windows, рядом с системными часами) отображается значок программы , позволяющий открыть свернутое окно или выйти из Skype.

Для звонков надо добавить знакомых в список контактов, для чего узнать у них логин Skype, полное имя или адрес электронной почты. Затем выбрать в меню *Контакты* команду *Новый контакт* и ввести данные в строку поиска (рис. 2.170). После нажатия кнопки *Поиск* в нижней части окна будет выведен список пользователей сети с подходящими координатами. Выбрав пользователя, следует нажать кнопку *Добавить контакт*. После того как этот пользователь согласится добавить себя в список ваших контактов, можно будет общаться с ним по Skype.

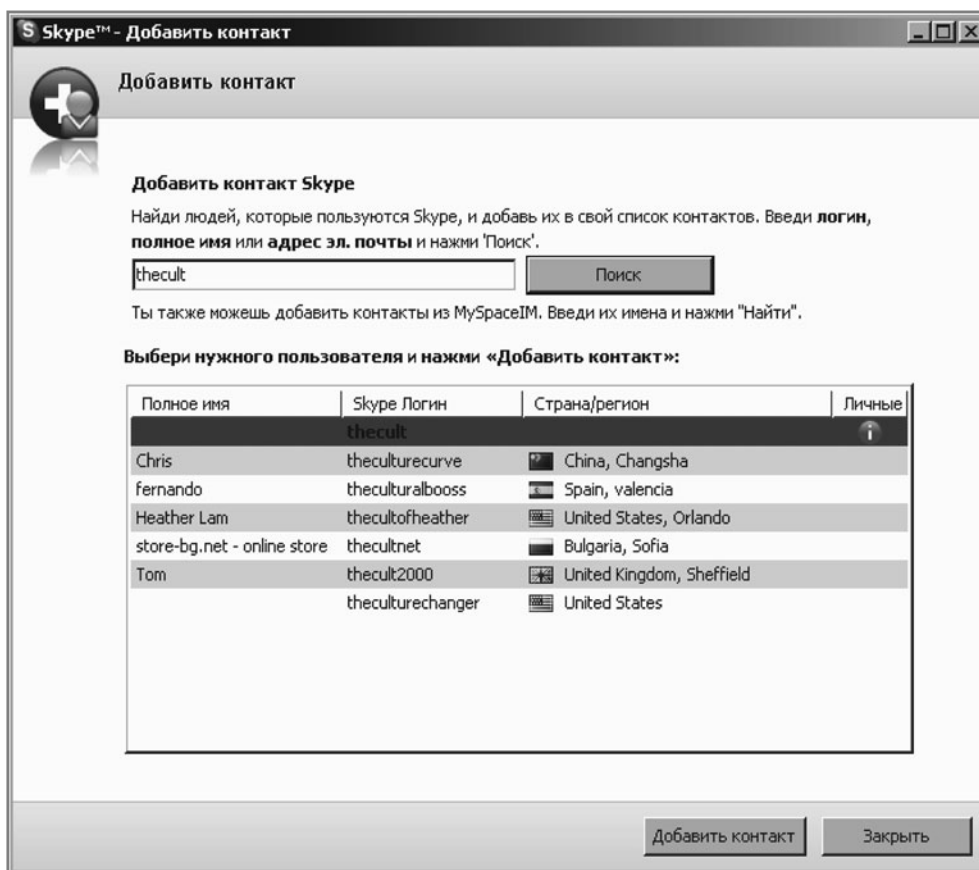



Рис. 2.170. Поиск контакта в сети

Чтобы позвонить зарегистрированному в сети Skype пользователю, надо выбрать его в левой панели окна в списке контактов (рис. 2.171) и нажать кнопку *Позвонить* для обычного звонка или *Видеозвонок* для общения с помощью веб-камеры. В правой части окна будет выведена панель инструментов и появятся кнопки регулировки качества связи. Кнопка *Положить трубку* (с изображением красной телефонной трубки) завершит разговор.

Для отправки текстовых сообщений также следует выделить имя пользователя в левой панели, ввести текст в область сообщений внизу правой панели и нажать кнопку  *Отправить сообщение* (или клавишу Enter). Сообщения доставляются мгновенно и хранятся в истории Skype. Просмотреть историю можно через меню *Разговоры — Просмотреть старые сообщения*.

Размеры области сообщения можно увеличивать, перетягивая мышью ее границы. На панели инструментов сверху имеется кнопка для вставки симпатичных смайликов.



Рис. 2.171. Основное окно программы Skype

Также через Skype можно отправить файл — для этого над областью сообщений имеется выпадающий список *Поделиться* (рис. 2.172), в котором надо выбрать команду *Отправить файл* и указать какой именно. Файл можно отправить, только если получатель находится в сети.

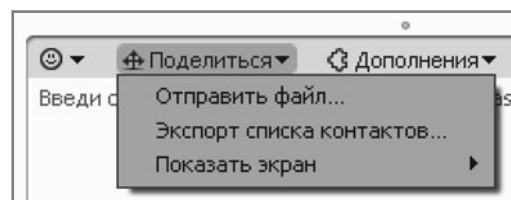



Рис. 2.172. Отправка файла через Skype

Следует иметь в виду, что кнопка  в правом верхнем углу окна не закрывает программу и не завершает работу в сети, а лишь сворачивает окно программы в системный трей, откуда оно может быть развернуто обратно с помощью команды контекстного меню. Для закрытия окна и выхода из сети надо в этом же контекстном меню выбрать команду *Выход* либо воспользоваться командой основного меню *Skype / Выйти*.

Тренировочные тестовые задания к разделу 2

Тема 2.1

При выполнении заданий с выбором ответа обведите кружком номер правильного ответа в задании.

1. Пользователь переходил из каталога в каталог в поисках файла *Документ.txt*, только спускаясь вниз и поднимаясь вверх. На уровень вниз было сделано на 2 перехода меньше, чем на уровень вверх. Каким может быть полный путь к найденному файлу, если местонахождение перед началом поиска *E:\Информатика\Задания\Решения*?
 - 1) *C:\Информатика\Документ.txt*
 - 2) *E:\Информатика\Задания\Документ.txt*
 - 3) *E:\Информатика\Документ.txt*
 - 4) *E:\Информатика\Задания\Решения\Новые\Вариант1\Документ.txt*
2. Пользователь переходил из каталога в каталог в поисках файла *Документ.txt*, только спускаясь вниз и поднимаясь вверх. На предпоследнем шаге поиска пользователь заходил в корневой каталог диска. Каким может быть полный путь к найденному файлу, если местонахождение перед началом поиска *G:\News\Today*?
 - 1) *G:\Документ.txt*
 - 2) *G:\News\Today\Документ.txt*
 - 3) *G:\Last\Документ.txt*
 - 4) *G:\News\Last\Документ.txt*
3. Пользователь переходил из каталога в каталог в поисках файла *Документ.txt*, только спускаясь вниз, поднимаясь вверх и переходя в каталог того же уровня, что и текущий. Было совершено 3 перехода, прежде чем был найден необходимый файл. Каким может быть полный путь к найденному файлу, если местонахождение перед началом поиска *C:\Книги\Авторы*?
 - 1) *C:\Книги\Поэзия\Документ.txt*
 - 2) *C:\Авторы\Документ.txt*
 - 3) *E:\Книги\Документ.txt*
 - 4) *C:\Книги\Документ*
4. Номера сертификатов содержат прописные буквы и цифры в любом порядке. Длина номера сертификата 5 символов. Задействовано 30 различных букв и все десятичные цифры.

Все символы закодировали одинаковым и минимально возможным количеством битов. Каждый номер сертификата кодируют с помощью посимвольного кодирования минимально возможным и одинаковым целым количеством байтов. Определить объем памяти компьютера, необходимый для записи 80 номеров.

 - 1) 80 байт
 - 2) 320 байт
 - 3) 400 байт
 - 4) 16 000 байт

Ответом к заданиям этой части является набор символов (букв или цифр), который следует записать в отведенном в задании поле для записи ответа.

5. На жесткий диск записаны 2 файла размером 1800 байт и 648 Кбайт. Известно, что размер одного кластера диска – 1024 байт. Сколько кластеров заняли оба файла?

Ответ: _____.

6. Два текстовых сообщения одной кодировки объемом 1,5 Мбит и 720 Кбит объединили в одно, а затем удалили из него фрагмент объемом 80 Кбайт. Определите объем получившегося сообщения в килобитах.

Ответ: _____.

7. Объем полученного текстового сообщения в кодировке ASCII составляет 1280 бит. Сколько символов содержит сообщение?

Ответ: _____.

8. Сообщения некоторого языка содержат буквенные символы (в алфавите 22 буквы), арабские цифры и 10 специальных знаков. Какое минимально возможное количество битов позволит закодировать тексты этого языка?

Ответ: _____.

9. Была осуществлена перекодировка текстового сообщения, первоначально записанного в кодировке ASCII, в двухбайтную кодировку Unicode. В результате информационный объем сообщения увеличился на 3,5 Кбита. Сколько символов содержит сообщение?

Ответ: _____.

10. В двух книгах сборника 100 и 120 страниц соответственно. В первом на каждой странице размещены 36 строк по 40 символов, во втором — 30 строк по 40 символов. Каков информационный объем (в байтах) двухтомника, сохраненного в кодировке ASCII?

Ответ: _____.

11. Скорость передачи данных через некоторое соединение составляет 128 000 Кбит/с. Передача данных через это соединение заняла $4/5$ минуты. Каков объем этих данных в мегабайтах?

Ответ: _____.

12. Какое наибольшее количество килобайтов может передаваться за одну секунду по каналу с максимальной пропускной способностью 100 Мбит/с?

Ответ: _____.

Тема 2.2

Ответом к заданиям этой части является набор символов (букв или цифр), который следует записать в отведенном в задании поле для записи ответа.

1. Имеется звуковая запись объемом 110 250 байт, глубина кодирования – 8 бит. Звуковая информация записана с частотой дискретизации 22,05 кГц. Какова длительность звучания такой информации в минутах?

Ответ: _____ .

2. Двадцатиминутная запись звуковой информации занимает на диске 50,47 Мбайт, глубина кодирования равна 16 бит. С какой частотой дискретизации (в килогерцах) записан звук?

Ответ: _____ .

3. Каков объем памяти (округленное целое число мегабайтов) для хранения файла монозвука, длительность звучания которого 10 мин при частоте дискретизации 44,1 кГц и глубине кодирования 16 бит?

Ответ: _____ .

При выполнении заданий с выбором ответа обведите кружком номер правильного ответа в задании.

4. При одноканальной цифровой звукозаписи (монозвук) для записи каждого значения сигнала использовали 32 бит. Значения сигнала фиксировали 48 000 раз в секунду. Полученная запись длится 6 мин, ее результаты сохранены в файле без сжатия данных. Какая из приведенных ниже величин наиболее близка к размеру полученного файла?

1) 66 Мбайт 2) 130 Мбайт 3) 527 Мбайт 4) 9 Мбайт

Тема 2.3

При выполнении заданий с выбором ответа обведите кружком номер правильного ответа в задании.

1. Выберите вариант текста, где правильно расставлены пробелы и оформлены знаки препинания.

1) Существует три основные логические операции : отрицание (НЕ), логическое умножение (И), логическое сложение (ИЛИ).

2) Существует три основные логические операции: отрицание (НЕ) , логическое умножение (И) , логическое сложение (ИЛИ) .

3) Существует три основные логические операции: отрицание (НЕ), логическое умножение (И), логическое сложение (ИЛИ).

4) Существует три основные логические операции: отрицание (НЕ), логическое умножение (И), логическое сложение (ИЛИ).

2. Укажите отличия в шрифтовом и абзацном оформлении двух фрагментов документа.

А) ОСНОВАНИЕ СИСТЕМЫ СЧИСЛЕНИЯ РАВНО КОЛИЧЕСТВУ ИСПОЛЬЗУЕМЫХ В НЕЙ ЦИФР. В ЗАПИСИ ДВОИЧНЫХ ЧИСЕЛ ИСПОЛЬЗУЮТ ЦИФРЫ 0 И 1.

Б) Основание системы счисления равно количеству используемых в ней цифр. В записи двоичных чисел используют цифры 0 и 1.

- 1) гарнитура шрифта;
- 2) начертание шрифта (прямое, курсивное);
- 3) насыщенность шрифта (светлый, полужирный);
- 4) выравнивание текста.

Ответом к заданиям этой части является набор символов (букв или цифр), который следует записать в отведенном в задании поле для записи ответа.

3. В табличной форме представлены усредненные метеорологические данные некоторого района по месяцам.

	Температура	Влажность	Давление
январь	−10	71 %	743
март	2	86 %	721
май	15	62 %	741
июль	25	58 %	786
сентябрь	14	73 %	727
ноябрь	3	88 %	682

Сколько записей удовлетворяют следующему условию:
(Температура > 20) ИЛИ (Давление > 730)

Ответ: _____.

4. В табличной форме представлены усредненные метеорологические данные некоторого района по месяцам.

	Температура	Влажность	Давление
январь	1	65 %	742
февраль	−5	81 %	735
март	10	56 %	747
апрель	16	54 %	742
май	24	70 %	752
июнь	29	82 %	746

Сколько записей удовлетворяют следующему условию:
(Влажность > 66) И (Давление < 750)

Ответ: _____.

5. В табличной форме представлены усредненные метеорологические данные не-которого района по месяцам.

	Температура	Влажность	Давление
июнь	25	65 %	747
июль	29	62 %	751
август	29	69 %	747
сентябрь	22	75 %	738
октябрь	18	81 %	751
ноябрь	11	62 %	744

Сколько записей удовлетворяют следующему условию:
(Температура > 20) ИЛИ (Влажность < 70) И (Давление < 750)

Ответ: _____.

Тема 2.4

При выполнении заданий с выбором ответа обведите кружком номер правильного ответа в задании.

1. Укажите запрос к поисковой системе с наименьшим количеством найденных результатов.
- 1) принтер & (сканер | веб-камера)
 - 2) принтер & сканер | веб-камера
 - 3) принтер & сканер
 - 4) принтер | сканер | веб-камера

Ответом к заданиям этой части является набор символов (букв или цифр), который следует записать в отведенном в задании поле для записи ответа.

2. Упорядочьте по убыванию количества найденных результатов запросы к поисковой системе. В ответе укажите последовательность номеров запросов.
- 1) Марс & Сатурн & Венера
 - 2) Марс & Сатурн
 - 3) Марс & Сатурн | Венера
 - 4) Марс | Сатурн | Венера

Ответ: _____.

3. Упорядочьте по возрастанию количества найденных результатов запросы к поисковой системе. В ответе укажите последовательность номеров запросов.

1) река & озеро & море

3) (река | море) & озеро

2) река & озеро

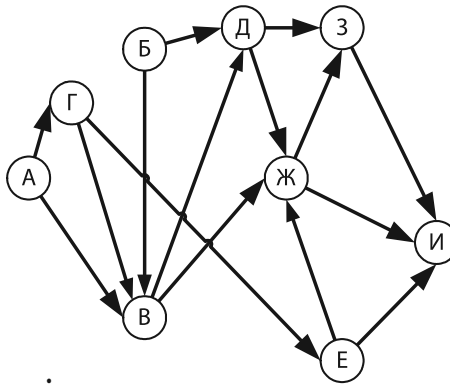
4) река | озеро | море

Ответ: _____.

Тема 2.5

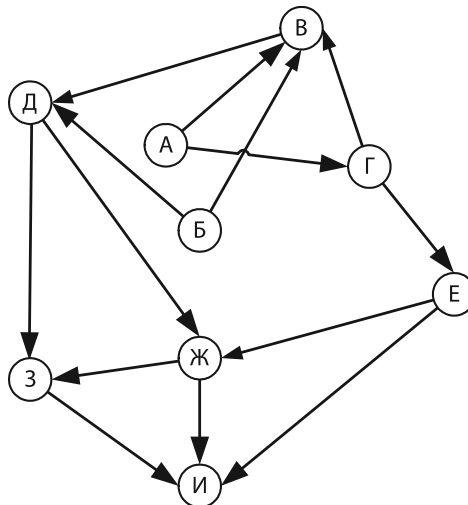
Ответом к заданиям этой части является набор символов (букв или цифр), который следует записать в отведенном в задании поле для записи ответа.

1. На рисунке изображена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город И?



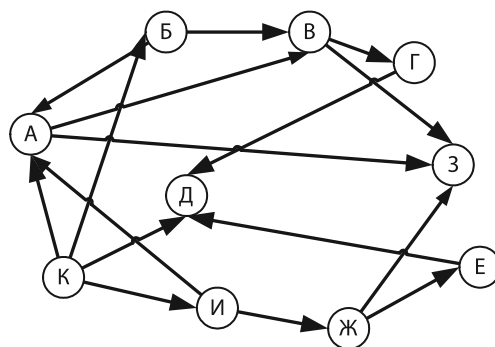
Ответ: _____.

2. На рисунке изображена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город И?



Ответ: _____.

3. На рисунке изображена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города К в город Д?



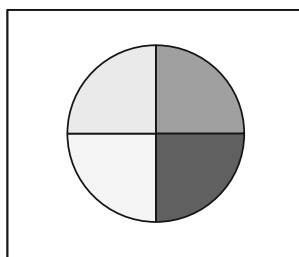
Ответ: _____.

Тема 2.6

При выполнении заданий с выбором ответа обведите кружком номер правильного ответа в задании.

1. Укажите правильный адрес ячейки
 1) C12E 2) B1256 3) 123A 4) B1C
2. В электронной таблице выделена группа ячеек A15:B23. Сколько ячеек входит в этот диапазон?
 1) 18 2) 16 3) 8 4) 11
3. В каком из указанных диапазонов содержится ровно 20 клеток электронной таблицы?
 1) B10:F20 2) C10:D12 3) C3:F7 4) A10:D15
4. Представлен фрагмент таблицы Excel. Какая формула должна находиться в ячейке F4, чтобы диаграмма, построенная по значениям диапазона ячеек C4:F4, соответствовала рисунку?

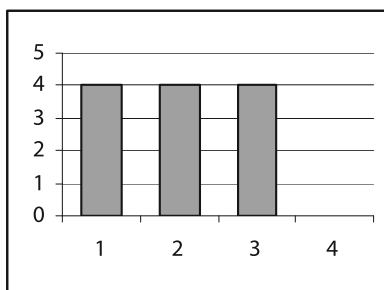
	A	B	C	D	E	F
1						
2						
3		2	3	2	3	4
4			=B3+C3	=C3+D3	=D3+E3	
5						



- 1) =E3+F3 2) =E3+2 3) =C3+3 4) =F3-E3

5. Представлен фрагмент таблицы Excel. Какая формула должна находиться в ячейке F4, чтобы диаграмма, построенная по значениям диапазона ячеек C4:F4, соответствовала рисунку?

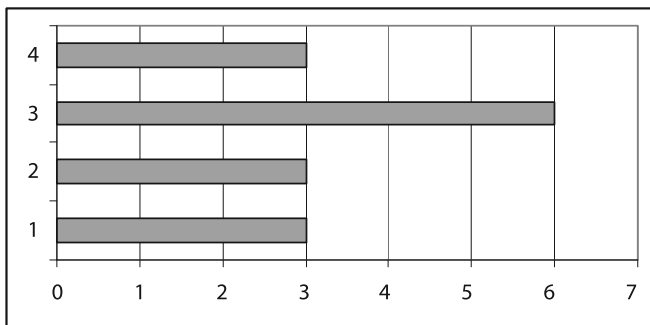
	A	B	C	D	E	F
1						
2						
3		3	5	7	9	11
4			=C3-1	=D3-B3	=E3-5	
5						



- 1) =C3/5-1 2) =C3+D3 3) =(F3+1)/2 4) =E3-2-B3

6. Представлен фрагмент таблицы Excel. Какая формула должна находиться в ячейке F4, чтобы диаграмма, построенная по значениям диапазона ячеек C4:F4, соответствовала рисунку?

	A	B	C	D	E	F
1						
2						
3			7	2	1	=D3+E3+1
4			=F3-1	=E3+2	=D3*3	
5						



- 1) =C4-3 2) =D3/2 3) =F3-3 4) =C3-4

Ответом к заданиям этой части является набор символов (букв или цифр), который следует записать в отведенном в задании поле для записи ответа.

7. Представлен фрагмент таблицы Excel в режиме вывода формул. Определите значение в ячейке E3.

	A	B	C	D	E
1					
2		2	5	4	
3			=СУММ(B2:D2)	=(C3+1)/3	=B2+D3
4					

Ответ: _____.

8. Представлен фрагмент таблицы Excel в режиме вывода формул. Определите значение в ячейке D3.

	A	B	C	D
1		1		
2	5	2	1	=СРЗНАЧ(B2:C4)
3		2	7	=СУММ(B2:D2)
4		9	3	
5				

Ответ: _____.

9. Представлен фрагмент таблицы Excel в режиме вывода формул. Определите значение в ячейке D3.

	A	B	C	D	E
1					
2					
3	2	4	=СРЗНАЧ(A3:B3)	=СУММ(A3:C3)	=СУММ(A3:D3)
4					

Ответ: _____.

10. Представлен фрагмент таблицы Excel. Определите значение в текущей ячейке.

C7		fx =СРЗНАЧ(D2:D5) - СРЗНАЧ(B2:B5)						
	A	B	C	D	E	F	G	H
1								
2			1	6				
3			2	4				
4			3	2				
5		четыре		ноль				

Ответ: _____.

11. Представлен фрагмент таблицы Excel. Определите значение в текущей ячейке.

	C6		fx =ИЛИ(B2<B4; C4>D3)				
	A	B	C	D	E	F	
1							
2			1	4	7		
3			2	5	8		
4			3	6	9		
5							

Ответ: _____.

12. Представлен фрагмент таблицы Excel. Определите значение в текущей ячейке.

	C6		fx =СУММ(C2:C4) + СЧЁТ(B2:B4;D2:D4)					
	A	B	C	D	E	F	G	H
1								
2			1	4	7			
3			2	5	8			
4			3	6	9			
5								

Ответ: _____.

Тема 2.7

При выполнении заданий с выбором ответа обведите кружком номер правильного ответа в задании.

1. Укажите верно записанный URL ресурса.

- 1) smell.uk//files.txt
- 2) ftp:/noname.com
- 3) ww://serv1/serv2/ans.flv
- 4) ftp://uniserv.net/folder2

Ответом к заданиям этой части является набор символов (букв или цифр), который следует записать в отведенном в задании поле для записи ответа.

2. Доступ к файлу *pic.bmp* на сервере *k1000.org* осуществляется по протоколу FTP. Файл находится в папке *cat1*, которая вложена в папку *cat8*. Фрагменты адресов ресурсов закодированы буквами от А до К. Запишите последовательность букв, кодирующих адрес ресурса в сети Интернет.

- | | | | |
|-----------|---------|----------|----------|
| А) k1000 | Б) pic | В) .cat1 | Г) cat1/ |
| Д) /cat8/ | Е) cat8 | Ж) // | З) .org |
| И) .bmp | К) ftp: | | |

Ответ: _____.

3. Доступ к файлу *cool.txt* на сервере *www.fine.com* осуществляется по протоколу HTTP. Файл находится в корневом каталоге сервера. Фрагменты адресов ресурсов закодированы буквами от А до К. Запишите последовательность букв, кодирующих адрес ресурса в сети Интернет.

А) www Б) : В) .fine Г) http
Д) cool Е) // Ж) .txt З) .com
И) http: К) /cool

Ответ: _____ .

4. Доступ к файлу *news.mp3* на сервере *www.news.tv* осуществляется по протоколу HTTP. Файл находится в папке *papers*, находящейся в корневом каталоге сервера. Запишите последовательность букв, кодирующих адрес ресурса в сети Интернет.

А) .news Б) /news В) :/ Г) http
Д) mailto Е) .tv Ж) /www З) /papers
И) ftp К) .mp3

Ответ: _____ .

ОТВЕТЫ

Раздел 1

Тема 1.1

1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.
3	1	4	4	1280	4	1	4	10100110	1445	1АСЗ	443	501	330

Тема 1.2

1.	2.	3.	4.	5.	6.	7.	8.
2	2	5	4	1113	ВЕТЕА	КОНКА	БИТВА

Тема 1.3

1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.	17.	18.	19.	20.
3	4	3	4	2	1	4	2	20	30	30	7	25	2	11	46	12	3	3	4

21.	22.	23.	24.	25.	26.	27.	28.	29.	30.
4	1	4	2	2	1	2	3	4ЗИМ6МА4ЗИ	573645У43342

31.	32.	33.	34.	35.	36.	37.
1000	АСЕВ	СДФВ	АСДВ	1221	21211	1212

Раздел 2

Тема 2.1

1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.
3	3	2	2	650	202	160	6	448	288000	375	12800

Тема 2.2

1.	2.	3.	4.
5	22,05	51	1

Тема 2.3

1.	2.	3.	4.	5.
3	1	3	2	5

Тема 2.4

1.	2.	3.
2	4321	1234

Тема 2.5

1.	2.	3.
21	15	6

Тема 2.6

1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.
2	1	3	2	4	4	6	7	18	2	ИСТИНА	21

Тема 2.7

1.	2.	3.	4.
4	КЖАЗДГБИ	ИЕАВЗКЖ	ГВЖАЕЗБК

Все права защищены. Книга или любая ее часть не может быть скопирована, воспроизведена в электронной или механической форме, в виде фотокопии, записи в память ЭВМ, репродукции или каким-либо иным способом, а также использована в любой информационной системе без получения разрешения от издателя. Копирование, воспроизведение и иное использование книги или ее части без согласия издателя является незаконным и влечет уголовную, административную и гражданскую ответственность.

Справочное издание
анықтамалық баспа

ОГЭ. УНИВЕРСАЛЬНЫЙ СПРАВОЧНИК

Дьячкова Ольга Владимировна

ОГЭ
ИНФОРМАТИКА
Универсальный справочник
(орыс тілінде)

Ответственный редактор *А. Жилинская*
Ведущий редактор *Т. Судакова*
Художественный редактор *А. Кашлев*

Россия. ООО «Издательство «Эксмо»

123308, Москва, ул. Зорге, д. 1. Тел.: 8 (495) 411-68-86.

Home page: www.eksmo.ru E-mail: info@eksmo.ru

Өндіруші: «ЭКМО» АҚБ Баспасы, 123308, Мәскеу, Ресей, Зорге көшесі, 1 үй.
Тел.: 8 (495) 411-68-86.

Home page: www.eksmo.ru E-mail: info@eksmo.ru

Тауар белгісі: «Эксмо»

Интернет-магазин : www.book24.ru

Интернет-магазин : www.book24.kz

Интернет-дүкен : www.book24.kz

Импортер в Республику Казахстан ТОО «РДЦ-Алматы».

Қазақстан Республикасындағы импорттаушы «РДЦ-Алматы» ЖШС.

Дистрибутор и представитель по приему претензий на продукцию,
в Республике Казахстан: ТОО «РДЦ-Алматы»

Қазақстан Республикасында дистрибутор және өнім бойынша арыз-талаптарды
қабылдаушының өкілі «РДЦ-Алматы» ЖШС,

Алматы қ., Домбровский көш., 3«а», литер Б, офис 1.

Тел.: 8 (727) 251-59-90/91/92; E-mail: RDC-Almaty@eksmo.kz

Өнімнің жарамдылық мерзімі шектелмеген.

Сертификация туралы ақпарат сайтта: www.eksmo.ru/certification

Сведения о подтверждении соответствия издания согласно законодательству РФ
о техническом регулировании можно получить на сайте Издательства «Эксмо»

www.eksmo.ru/certification

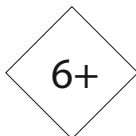
Өндірген мемлекет: Ресей. Сертификация қарастырылған

Дата изготовления / Подписано в печать 29.04.2019. Формат 84x108¹/₁₆.

Печать офсетная. Усл. печ. л. 28,56.

Тираж экз. Заказ

ISBN 978-5-04-103754-3



ЕКСМО.РУ
новинки издательства



**ЭФФЕКТИВНАЯ
ПОДГОТОВКА
К ОГЭ**

ОГЭ



ПОЛУЧИ ВЫСШИЙ БАЛЛ НА ОГЭ!

ИЗДАНИЕ ПОМОЖЕТ:

- сократить время подготовки к ОГЭ;
- повторить все темы курса;
- отработать навыки выполнения заданий разных типов.

ИНФОРМАТИКА

УНИВЕРСАЛЬНЫЙ СПРАВОЧНИК

В серии «ОГЭ. Универсальный справочник» выходят пособия по основным школьным предметам: русскому языку, литературе, математике, истории, обществознанию, биологии, информатике, химии и физике.

ISBN 978-5-04-103754-3



9 785041 037543 >



www.vk.com/eksmo_kids