



СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
SIBERIAN FEDERAL UNIVERSITY

А. Н. Пупков, В. В. Самарин, Р. Ю. Царёв

ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ

Учебное
пособие

УМО

ИНСТИТУТ УПРАВЛЕНИЯ БИЗНЕС-ПРОЦЕССАМИ
И ЭКОНОМИКИ

ПРИКЛАДНАЯ ИНФОРМАТИКА

Министерство образования и науки Российской Федерации
Сибирский федеральный университет

А. Н. Пупков, В. В. Самарин, Р. Ю. Царёв

Информатика и программирование

Рекомендовано УМО РАЕ по классическому университетскому и техническому образованию в качестве учебного пособия для студентов высших учебных заведений, обучающихся по специальностям 080801.65 «Прикладная информатика», 080801.65.01 «Прикладная информатика в экономике», 080801.65.02 «Прикладная информатика в менеджменте», 080801.65.29 «Прикладная информатика в рекламе», 080801.65.28 «Прикладная информатика в международном бизнесе», 06.06.2012 г.

Красноярск
СФУ
2012

УДК 004.424(07)
ББК 32.973.2-018(я73)
П888

Рецензенты:

А. Н. Антамошкин, д-р техн. наук, проф. Красноярского
государственного аграрного университета;
А. В. Медведев, д-р техн. наук, проф. Сибирского государственного
аэрокосмического университета им. акад. М. Ф. Решетнёва

Пупков, А. Н.

П888 Информатика и программирование : учеб. пособие / А. Н. Пупков,
В. В. Самарин, Р. Ю. Царёв. – Красноярск : Сиб. федер. ун-т, 2012. –
156 с.
ISBN 978-5-7638-2434-6

Рассмотрены базовые понятия информатики. Изложены основные приемы работы и поиска в глобальной сети Интернет, вопросы информационной безопасности и работы с антивирусным программным обеспечением. Описаны синтаксические конструкции языка VBA, приведены примеры программ на VBA в среде Excel. Представлены задания для самостоятельной работы.

Предназначено студентам укрупненной группы направления подготовки специалистов 080801.65 «Прикладная информатика».

УДК 004.424(07)
ББК 32.973.2-018(я73)

ISBN 978-5-7638-2434-6

© Сибирский федеральный
университет, 2012

Учебное издание
Пупков Александр Николаевич, Самарин Виктор Викторович
Царёв Роман Юрьевич
ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ
Учебное пособие

Редактор Л. Ф. Калашник Корректор В. А. Степанова
Компьютерная верстка И. В. Гревцовой

Подписано в печать 05.12.2012. Формат 60х84/16. Бумага офсетная. Печать плоская.
Усл. печ. л. 9,75. Тираж 100 экз. Заказ № 5110.

Издательский центр Библиотечно-издательского комплекса СФУ
660041, г. Красноярск, пр. Свободный, 79. Тел./факс (391) 206-21-49, e-mail: rio@lan.krasu.ru
Отпечатано Полиграфическим центром Библиотечно-издательского комплекса СФУ
660041, г. Красноярск, пр. Свободный, 82а. Тел./факс (391) 206-26-58, 206-26-49
E-mail: print_sfu@mail.ru; <http://lib.sfu-kras.ru>

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ	6
1.1. Понятия «информация», «данные», формы адекватности информации. Меры информации	6
1.2. Информационные системы. Структура и классификация информационных систем	9
1.3. Понятие «информационные технологии». Виды информационных технологий	11
1.4. Архитектура персонального компьютера. Назначение основных узлов. Функциональные характеристики компьютера	13
1.5. Программное обеспечение компьютера. Общая характеристика, состав и назначение основных видов программного обеспечения компьютера	16
1.6. Архивация данных. Программы-архиваторы	18
1.7. Компьютерные вирусы. Антивирусные программы	21
1.8. Компьютерные сети. Особенности построения. Назначение и классификация	26
1.9. Глобальная сеть Интернет. Общая характеристика, особенности построения	32
ГЛАВА 2. MICROSOFT WORD	37
2.1. Знакомство с программой Microsoft Word 2007	37
2.2. Как не следует набирать текст	41
2.3. Редактирование текста	43
2.4. Форматирование текста	48
ГЛАВА 3. ПРОГРАММА MICROSOFT EXCEL 2007	63
3.1. Знакомство с программой Microsoft Excel 2007	63
3.2. Проведение расчетов в программе Microsoft Excel 2007	71
ГЛАВА 4. ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ VISUAL BASIC FOR APPLICATION	79
4.1. Основы программирования на языке VBA	79
4.2. Операторы языка VBA и реализация основных алгоритмических конструкций	90
ЗАКЛЮЧЕНИЕ	110
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	112
Приложение 1. Задания для Microsoft Word	113
Приложение 2. Задания для Microsoft Excel	129
Приложение 3. Задания для Microsoft VBA	152

ВВЕДЕНИЕ

Термин «информатика» возник в начале 60-х гг. XX в. во Франции для выделения области знаний, связанной с автоматизированной обработкой информации с помощью электронно-вычислительных машин.

Информатика – это научная и прикладная область знаний о законах, методах и способах накопления, обработки и передачи информации с помощью компьютерных и других технических средств.

Информатика изучает свойства, структуру и функции информационных систем, а также происходящие в них информационные процессы. Под информационной системой понимают систему, организующую, хранящую и преобразующую информацию. Подавляющее большинство современных информационных систем являются автоматизированными.

Информатика тесно связана с кибернетикой, но не тождественна ей. Кибернетика изучает общие закономерности процессов управления сложными системами в разных областях человеческой деятельности независимо от наличия или отсутствия компьютеров. Информатика же изучает общие свойства только конкретных информационных систем.

Информатику можно рассматривать как науку, как технологию и как индустрию.

Информатика как наука объединяет группу дисциплин, в которых изучаются различные аспекты свойств информации в информационных процессах, а также применение алгоритмических, математических и программных средств для ее обработки с помощью компьютеров.

Информатика как технология включает в себя систему процедур компьютерного преобразования информации с целью ее формирования, хранения, обработки, распространения и использования.

Основными чертами современной информационной технологии являются:

- дружественный программный и аппаратный интерфейс;
- интерактивный (диалоговый) режим решения задач;
- сквозная информационная поддержка всех этапов решения задачи на основе интегрированной базы данных;

- возможность коллективного решения задач на основе информационных сетей и систем телекоммуникаций;
- безбумажная технология, при которой основным носителем информации является не бумажный, а электронный документ.

Информатика как индустрия – это инфраструктурная отрасль народного хозяйства, обеспечивающая все другие отрасли необходимыми информационными ресурсами. Индустрия информатики включает в себя предприятия, производящие вычислительную технику и ее элементы; вычислительные центры различного типа и назначения (индивидуальные, кустовые, коллективного пользования и др.); предприятия, осуществляющие производство программных средств и проектирование информационных систем; организации, накапливающие, распространяющие и обслуживающие фонды алгоритмов и программ; станции технического обслуживания вычислительной техники.

Роль информатики в современных условиях постоянно возрастает. Деятельность как отдельных людей, так и целых организаций практически полностью зависит от их информированности и способности эффективно использовать имеющуюся информацию. Внедрение компьютеров, современных средств переработки и передачи информации в различные области индустрии послужило началом процесса, называемого информатизацией общества. Современное материальное производство и другие сферы деятельности не могут существовать без информационного обслуживания и переработки огромного количества информации. Информатизация на основе внедрения компьютерных и телекоммуникационных технологий является реакцией общества на потребность в существенном увеличении производительности труда в информационном секторе общественного производства, где сосредоточено более половины трудоспособного населения.

Результатом процесса информатизации является создание информационного общества, где манипулируют не материальными объектами, а идеями, образами, интеллектом, знаниями. Для каждой страны ее движение от индустриального этапа развития к информационному определяется степенью информатизации общества.

ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ

1.1. Понятия «информация», «данные», формы адекватности информации. Меры информации

Информация – это сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состояниях, которые уменьшают имеющуюся о них степень неопределенности, неполноты знаний.

Информатика рассматривает информацию как связанные между собой сведения, изменяющие наши представления о явлении или объекте окружающего мира. С этой точки зрения информацию можно рассматривать как совокупность знаний о фактических данных и зависимостях между ними.

В процессе обработки информация может менять структуру и форму. Признаком структуры являются элементы информации и их взаимосвязь. Формы представления информации могут быть различными. Основными из них являются: символьная (основана на использовании различных символов), текстовая (текст – это символы, расположенные в определенном порядке), графическая (различные виды изображений), звуковая.

В повседневной практике такие понятия, как информация и данные, часто рассматриваются как синонимы. На самом деле между ними имеются различия. Данными называется информация, представленная в удобном для обработки виде. Они могут быть представлены в виде текста, графики, аудиовизуального ряда. Представление данных называется языком информатики, представляющим собой совокупность символов, соглашений и правил, используемых для общения, отображения, передачи информации в электронном виде.

Одной из важнейших характеристик информации является ее адекватность. Адекватность информации – это уровень соответствия образа, создаваемого с помощью информации, реальному объекту, процессу, явлению. От степени адекватности информации зависит правильность принятия решения.

Адекватность информации может выражаться в трех формах: синтаксической, семантической и прагматической. Каждая из этих форм имеет свою меру.

Синтаксическая адекватность отображает формально-структурные характеристики информации, не затрагивая ее смыслового содержания. На синтаксическом уровне учитываются тип носителя и способ представления информации, скорость ее передачи и обработки, размеры кодов представления информации, надежность и точность преобразования этих кодов и т. д. Информацию, рассматриваемую с таких позиций, обычно называют данными.

Синтаксическая мера информации оперирует с обезличенной информацией, не выражающей смыслового отношения к объекту. На этом уровне объем данных в сообщении измеряется количеством символов в этом сообщении. В современных ЭВМ минимальной единицей измерения данных является бит – один двоичный разряд.

В информатике, в основном, используется величина, называемая байтом (byte) и равная 8 битам.¹ И если бит позволяет выбрать один вариант из двух возможных, то байт, соответственно, 1 из 256 (2^8). В большинстве современных ЭВМ при кодировании каждому символу соответствует своя последовательность из восьми нулей и единиц, т. е. байт. Соответствие байтов и символов задается с помощью таблицы, в которой для каждого кода указывается свой символ. Так, например, в широко распространенной кодировке Koi8-R буква «М» имеет код 11101101, буква «И» – код 11101001, а пробел – код 00100000.

Наряду с байтами для измерения количества информации используются более крупные, производные единицы:

1 Кб (килобайт) = 2^{10} б (байт) = 1024 б;

1 Мб (мегабайт) = 2^{10} Кб = 1024 Кб;

1 Гб (гигабайт) = 2^{10} Мб = 1024 Мб;

1 Тб (терабайт) = 2^{10} Гб = 1024 Гб;

1 Пб (петабайт) = 2^{10} Тб = 1024 Тб².

¹ Ряд ЭВМ 1950-х и 1960-х годов (БЭСМ-6, М-220) использовали 6-битовые символы в 48-битовых или 60-битовых машинных словах. В некоторых моделях ЭВМ производства Burroughs Computer Corporation (англ.) (ныне Unisys (англ.)) размер символа был равен 9 битам. В советской ЭВМ Минск-32 использовался 7-битный байт. 8-битные байты были приняты в System/360, вероятно, из-за использования BCD-формата представления чисел: одна десятичная цифра (0-9) требует 4 бита (тетраду) для хранения; один 8-битный байт может представлять две десятичные цифры. 6-битные байты могут хранить только по одной десятичной цифре, два бита остаются незадействованными. Постепенно 8-битные байты стали стандартом де-факто, и с начала 1970-х гг. в большинстве компьютеров байты состоят из 8 битов и размер машинного слова кратен 8 битам.

² Согласно предложению МЭК, название «петабайт» общепринято, но формально неверно, так как приставка *пета-*, означает умножение на 10^{15} , а не 2^{50} . Правильной для

Семантическая адекватность определяет степень соответствия образа объекта самому объекту. Здесь учитывается смысловое содержание информации. На этом уровне анализируются сведения, отражаемые информацией, рассматриваются смысловые связи. Таким образом, семантическая адекватность проявляется при наличии единства информации и пользователя. Эта форма служит для формирования понятий и представлений, выявления смысла, содержания информации и ее обобщения.

Семантическая мера информации используется для измерения смыслового содержания информации. Наибольшее распространение здесь получила тезаурусная мера, связывающая семантические свойства информации со способностью пользователя принимать поступившее сообщение. Тезаурус – это совокупность сведений, которыми располагает пользователь или система. Максимальное количество семантической информации потребитель получает при согласовании ее смыслового содержания со своим тезаурусом, когда поступающая информация понятна пользователю и несет ему ранее неизвестные сведения. С семантической мерой количества информации связан коэффициент содержательности, определяемый как отношение количества семантической информации к общему объему данных.

Прагматическая адекватность отражает соответствие информации цели управления, реализуемой на ее основе. Прагматические свойства информации проявляются при наличии единства информации, пользователя и цели управления. На этом уровне анализируются потребительские свойства информации, связанные с практическим использованием информации, с соответствием ее целевой функции деятельности системы.

Прагматическая мера информации определяет ее полезность, ценность для процесса управления. Обычно ценность информации измеряется в тех же единицах, что и целевая функция управления системой.

Вопросы и задания для самоконтроля

1. Дайте определение информатики как науки, технологии, индустрии. Определите предмет информатики.
2. Какова роль информатики в современном обществе?
3. Дайте определение понятия «информация».

²⁵⁰ является двоичная приставка *неби-*. Таким образом, если используется система СИ, то $1 \text{ ПБ} = 10^{15} \text{ Б}$, а если стандарт МЭК 60027-2, то $1 \text{ ПБ} = 2^{50} \text{ Б}$.

4. Дайте определение понятия «данные».
5. Дайте определение понятия «знания».
6. Какие формы представления информации вы знаете?
7. Перечислите формы адекватности информации и дайте их характеристику.
8. Назовите единицы измерения информации.
9. Дайте определение понятия «экономическая информация».
10. Перечислите показатели качества информации и дайте их характеристику.

1.2. Информационные системы.

Структура и классификация информационных систем

Информационная система – это взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации, используемой в достижении цели управления. В современных условиях основным техническим средством обработки информации является персональный компьютер. Большинство современных информационных систем преобразуют не информацию, а данные. Поэтому часто их называют системами обработки данных.

По степени механизации процедур преобразования информации системы обработки данных делятся на системы ручной обработки, механизированные, автоматизированные и системы автоматической обработки данных.

Важнейшими принципами построения эффективных информационных систем являются следующие:

Принцип интеграции, заключающийся в том, что обрабатываемые данные, однажды введенные в систему, многократно используются для решения большого числа задач.

Принцип системности, заключающийся в обработке данных в различных аспектах, чтобы получить информацию, необходимую для принятия решений на всех уровнях управления.

Принцип комплексности, заключающийся в механизации и автоматизации процедур преобразования данных на всех этапах функционирования информационной системы.

Информационные системы также классифицируются:

- по функциональному назначению – производственные, коммерческие, финансовые, маркетинговые и др.;

- объектам управления – информационные системы автоматизированного проектирования, управления технологическими процессами, управления предприятием (офис, фирма, корпорация, организация) и т. п.;
- характеру использования результатной информации – информационно-поисковые, предназначенные для сбора, хранения и выдачи информации по запросу пользователя; информационно-советующие, предлагающие пользователю определенные рекомендации для принятия решений (системы поддержки принятия решений); информационно-управляющие, результатная информация которых непосредственно участвует в формировании управляющих воздействий.

Структуру информационных систем составляет совокупность отдельных ее частей, называемых подсистемами.

Функциональные подсистемы реализуют и поддерживают модели, методы и алгоритмы получения управляющей информации. Состав функциональных подсистем весьма разнообразен и зависит от предметной области использования информационной системы, специфики хозяйственной деятельности объекта, управления.

В состав обеспечивающих подсистем обычно входят:

1. Информационное обеспечение – методы и средства построения информационной базы системы, включающие системы классификации и кодирования информации, унифицированные системы документов, схемы информационных потоков, принципы и методы создания баз данных.

2. Техническое обеспечение – комплекс технических средств, задействованных в технологическом процессе преобразования информации в системе. В первую очередь это вычислительные машины, периферийное оборудование, аппаратура и каналы передачи данных.

3. Программное обеспечение – совокупность программ регулярного применения, необходимых для решения функциональных задач, и программ, позволяющих наиболее эффективно использовать вычислительную технику, обеспечивая пользователям наибольшие удобства в работе.

4. Математическое обеспечение – совокупность математических методов, моделей и алгоритмов обработки информации, используемых в системе.

5. Лингвистическое обеспечение – совокупность языковых средств, используемых в системе с целью повышения качества ее разработки и облегчения общения человека с машиной.

Организационные подсистемы по существу относятся также к обеспечивающим подсистемам, но направлены в первую очередь на обеспечение эффективной работы персонала, и поэтому они могут быть выделены отдельно. К ним относятся:

1. Кадровое обеспечение – состав специалистов, участвующих в создании и работе системы, штатное расписание и функциональные обязанности.

2. Эргономическое обеспечение – совокупность методов и средств, используемых при разработке и функционировании информационной системы, создающих оптимальные условия для деятельности персонала, для быстрейшего освоения системы.

3. Правовое обеспечение – совокупность правовых норм, регламентирующих создание и функционирование информационной системы, порядок получения, преобразования и использования информации.

4. Организационное обеспечение – комплекс решений, регламентирующих процессы создания и функционирования как системы в целом, так и ее персонала.

Вопросы и задания для самоконтроля

1. Дайте определение понятия «информационная система».
2. Каковы важнейшие принципы построения информационных систем?
3. Охарактеризуйте функциональные подсистемы.
4. Охарактеризуйте организационные подсистемы.

1.3. Понятие «информационные технологии».

Виды информационных технологий

Технология (от греч. Τέχνη – искусство, мастерство, умение; др.-греч. Λόγος – мысль, причина; методика, способ производства) – комплекс организационных мер, операций и приемов, направленных на изготовление, обслуживание, ремонт и/или эксплуатацию изделия с номинальным качеством и оптимальными затратами и обусловленных текущим уровнем развития науки, техники и общества в целом.

Информационная технология – это процесс, в котором используется совокупность средств и методов сбора, обработки и передачи данных для получения информации нового качества о состоянии обь-

екта, процесса или явления. Цель информационной технологии – производство информации для ее анализа человеком и принятия на его основе решения по выполнению какого-либо действия.

В качестве инструментария информационной технологии используются распространенные виды программных продуктов: текстовые процессоры, издательские системы, электронные таблицы, системы управления базами данных, электронные календари, информационные системы функционального назначения.

К основным видам информационных технологий относятся следующие:

1. Информационная технология обработки данных предназначена для решения хорошо структурированных задач, алгоритмы решения которых хорошо известны и для решения которых имеются все необходимые входные данные. Эта технология применяется на уровне исполнительской деятельности персонала невысокой квалификации в целях автоматизации некоторых рутинных, постоянно повторяющихся операций управленческого труда.

2. Информационная технология управления предназначена для информационного обслуживания всех работников предприятий, связанных с принятием управленческих решений. Здесь информация обычно представляется в виде регулярных или специальных управленческих отчетов и содержит сведения о прошлом, настоящем и возможном будущем предприятия.

3. Информационная технология автоматизированного офиса призвана дополнить существующую систему связи персонала предприятия. Автоматизация офиса предполагает организацию и поддержку коммуникационных процессов как внутри фирмы, так и с внешней средой на базе компьютерных сетей и других современных средств передачи и работы с информацией.

4. Информационная технология поддержки принятия решений предназначена для выработки управленческого решения, происходящей в результате итерационного процесса, в котором участвуют система поддержки принятия решений (вычислительное звено и объект управления) и человек (управляющее звено, задающее входные данные и оценивающее полученный результат).

5. Информационная технология экспертных систем основана на использовании искусственного интеллекта. Экспертные системы дают возможность менеджерам получать консультации экспертов по любым проблемам, о которых в этих системах накоплены знания.

Вопросы и задания для самоконтроля

1. Дайте определение понятия «информационная технология».
2. Каковы основные виды информационных технологий?

1.4. Архитектура персонального компьютера.

Назначение основных узлов.

Функциональные характеристики компьютера

Компьютер – это многофункциональное электронное устройство, предназначенное для накопления, обработки и передачи информации. Под архитектурой персонального компьютера понимается его логическая организация, структура и ресурсы, т. е. средства вычислительной системы, которые могут быть выделены процессу обработки данных на определенный интервал времени.

В основу построения большинства компьютеров положены принципы, сформулированные Джоном фон Нейманом.

1. Принцип программного управления – программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности.

2. Принцип однородности памяти – программы и иные данные хранятся в одной и той же памяти; над командами можно выполнять те же действия, что и над данными!

3. Принцип адресности – основная память структурно состоит из пронумерованных ячеек.

Компьютеры, построенные на этих принципах, имеют классическую архитектуру.

Архитектура компьютера определяет принцип действия, информационные связи и взаимное соединение основных логических узлов компьютера, к которым относятся: центральный процессор; основная память; внешняя память; периферийные устройства.

Конструктивно персональные компьютеры выполнены в виде центрального системного блока, к которому через специальные разъемы присоединяются другие устройства. В состав системного блока входят все основные узлы компьютера:

- системная плата;
- блок питания;
- накопитель на жестком магнитном диске;
- накопитель на гибком магнитном диске;

- накопитель на оптическом диске;
- разъемы для дополнительных устройств.

На системной (материнской) плате в свою очередь размещаются:

- микропроцессор;
- математический сопроцессор;
- генератор тактовых импульсов;
- микросхемы памяти;
- контроллеры внешних устройств;
- звуковая и видеокарты;
- таймер.

Архитектура современных персональных компьютеров основана на магистрально-модульном принципе. Модульный принцип позволяет пользователю самому комплектовать нужную ему конфигурацию компьютера и производить при необходимости ее модернизацию. Модульная организация системы опирается на магистральный принцип обмена информацией. Все контроллеры устройств взаимодействуют с микропроцессором и оперативной памятью через системную магистраль передачи данных, называемую системной шиной. Системная шина выполняется в виде печатного мостика на материнской плате.

Микропроцессор – это центральный блок персонального компьютера, предназначенный для управления работой всех блоков машины и для выполнения арифметических и логических операций над информацией.

Системная шина является основной интерфейсной системой компьютера, обеспечивающей сопряжение и связь всех его устройств между собой. Системная шина обеспечивает три направления передачи информации: между микропроцессором и основной памятью; между микропроцессором и портами ввода-вывода внешних устройств; между основной памятью и портами ввода-вывода внешних устройств.

Порты ввода-вывода всех устройств через соответствующие разъемы (слоты) подключаются к шине либо непосредственно, либо через специальные контроллеры (адаптеры).

Основная память предназначена для хранения и оперативного обмена информацией с прочими блоками компьютера.

Внешняя память используется для долговременного хранения информации, которая может быть в дальнейшем использована для решения задач. Генератор тактовых импульсов генерирует последова-

тельность электрических символов, частота которых задает тактовую частоту компьютера. Промежуток времени между соседними импульсами определяет такт работы машины.

Источник питания – это блок, содержащий системы автономного и сетевого питания компьютера.

Таймер – это внутримашинные электронные часы, обеспечивающие автоматический съём текущего момента времени. Таймер подключается к автономному источнику питания и при отключении компьютера от сети продолжает работать.

Внешние устройства компьютера обеспечивают взаимодействие машины с окружающей средой: пользователями, объектами управления и другими компьютерами.

Основными функциональными характеристиками персонального компьютера (ПК) являются:

1. Производительность, быстродействие, тактовая частота. Производительность современных ЭВМ измеряют обычно в миллиардах операций в секунду (гигагерц).

2. Разрядность микропроцессора и кодовых шин интерфейса. Разрядность – это максимальное количество разрядов двоичного числа, над которым одновременно может выполняться машинная операция, в том числе и операция передачи информации; чем больше разрядность, тем при прочих равных условиях будет больше и производительность ПК.

3. Типы системного и локальных интерфейсов. Разные типы интерфейсов обеспечивают разные скорости передачи информации между узлами машины, позволяют подключать разное количество внешних устройств и различные их виды.

4. Емкость оперативной памяти. Емкость оперативной памяти измеряется обычно в Мбайтах. Многие современные прикладные программы с оперативной памятью, имеющей емкость меньше 16 Мбайт, просто не работают либо работают, но очень медленно.

5. Емкость накопителя на жестких магнитных дисках (винчестера). Емкость винчестера измеряется обычно в Гбайтах.

6. Тип и емкость накопителей на гибких магнитных дисках. Сейчас применяются накопители на гибких магнитных дисках, использующие дискеты диаметром 3,5 дюйма, имеющие стандартную емкость 1,44 Мб.

7. Наличие, виды и емкость кэш-памяти. Кэш-память – это буферная, недоступная для пользователя быстродействующая память,

автоматически используемая компьютером для ускорения операций с информацией, хранящейся в более медленно действующих запоминающих устройствах. Наличие кэш-памяти емкостью 256 Кбайт увеличивает производительность персонального компьютера примерно на 20 %.

8. Тип видеомонитора и видеоадаптера.
9. Наличие и тип накопителя на оптических дисках.
10. Наличие и виды мультимедийных аудио/видеосредств.
11. Имеющееся программное обеспечение и вид операционной системы.
12. Аппаратная и программная совместимость с другими типами ЭВМ. Данная совместимость означает возможность использования на компьютере соответственно тех же технических элементов и программного обеспечения, что и на других типах машин.
13. Возможность работы в вычислительной сети.
14. Надежность. Надежность – это способность системы выполнять полностью и правильно все заданные ей функции.
15. Стоимость, габариты, вес.

Вопросы и задания для самоконтроля

1. Дайте определение понятия «компьютер».
2. Перечислите основные принципы построения персональных компьютеров.
3. Перечислите основные узлы персонального компьютера.
4. Что такое системная шина?
5. Назовите основные функциональные характеристики персонального компьютера.
6. Перечислите основные устройства ввода и вывода информации.
7. Назовите виды компьютерной памяти.

1.5. Программное обеспечение компьютера.

Общая характеристика, состав и назначение основных видов программного обеспечения компьютера

Программное обеспечение (ПО) – это совокупность программ, позволяющих осуществить на компьютере автоматизированную обработку информации.

ПО делится на системное, прикладное и инструментальное.

Системное ПО – это совокупность программ для обеспечения работы компьютера и сетей ЭВМ.

Данный класс ПО делится на следующие виды: базовое, содержащее операционные системы (ОС), операционные оболочки и драйверы и сервисное, или утилиты, содержащие программы диагностики работоспособности компьютера, антивирусные программы, программы обслуживания дисков, программы архивирования данных, программы обслуживания сетей.

Операционные системы используются для управления выполнением пользовательских программ, планирования и управления вычислительными и другими ресурсами ЭВМ. Это комплекс программ и данных, предназначенных для увеличения пропускной способности ЭВМ, расширения сферы ее применения, автоматизации подготовки прикладных программ к выполнению. ОС являются обязательной составляющей ПО любого компьютера, без чего он не может функционировать должным образом. Наиболее популярными ОС для компьютеров класса IBM PC являются ОС семейства Windows и UNIX.

Операционные оболочки – это специальные программы, предназначенные для облегчения общения пользователя с командами ОС. Они имеют текстовый и графический варианты интерфейса конечного пользователя. Примерами являются Total Commander Far. Обычно операционные оболочки имеют набор функций, облегчающих пользователю работу с популярными сетевыми и внутренними сервисами, например, FTP-менеджер или редактор реестра.

Драйверы – это компьютерные программы, с помощью которых другая программа (обычно операционная система) получает доступ к аппаратному обеспечению некоторого устройства.

Пакеты прикладных программ (ПП) – это комплекс взаимосвязанных программ для решения задач определенного класса. Выделяются следующие виды ПП:

- проблемно-ориентированные. Используются для тех проблемных областей, в которых возможна типизация функций управления, структур данных и алгоритмов обработки. Например, это ППП автоматизации бухучета, финансовой деятельности, управления персоналом и т. д.;
- автоматизации проектирования. Используются в работе конструкторов и технологов, связанных с разработкой чертежей, схем, диаграмм;

- общего назначения. Включают текстовые и табличные процессоры, графические редакторы, системы управления базами данных (СУБД);
- офисные. Обеспечивают организационное управление деятельностью офиса. Включают органайзеры (записные и телефонные книжки, календари, презентации и т. д.), средства распознавания текста;
- системы искусственного интеллекта. Используют в работе некоторые принципы обработки информации, свойственные человеку. Включают информационные системы, поддерживающие диалог на естественном языке; экспертные системы, позволяющие давать рекомендации пользователю в различных ситуациях; интеллектуальные пакеты прикладных программ, позволяющие решать прикладные задачи без программирования.

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

Инструментарий технологии программирования делится на два класса инструментальных средств: для создания отдельных приложений (программ) и для создания информационных систем и технологий.

Вопросы и задания для самоконтроля

1. Перечислите и охарактеризуйте виды программного обеспечения ПК.
2. Какие виды прикладного программного обеспечения вы знаете?
3. Назовите виды системного программного обеспечения.

1.6. Архивация данных. Программы-архиваторы

Назначение программ-архиваторов заключается в экономии места на диске за счет сжатия (упаковки) одного или нескольких файлов в архивный файл. Программы-архиваторы используют для хранения в упакованном виде больших объемов информации, которая понадобится в будущем; переноса информации между компьютерами с помощью дискет или электронной почты; создания в сжатом виде ре-

зервных копий файлов; для защиты от компьютерных вирусов. В результате работы программ-архиваторов создаются архивные файлы (архивы).

В основе работы программ-архиваторов лежит процедура поиска и перекодирования одинаковых фрагментов содержимого файлов. Сжатие информации в файлах производится за счет устранения избыточности различными способами (за счет упрощения кодов, исключения постоянных битов, замены их повторяющейся последовательности коэффициентом повторения и т. д.). Существует множество алгоритмов сжатия данных. Современные архиваторы, как правило, одновременно используют несколько методов. Можно выделить некоторые основные.

1. Кодирование длин серий RLE (сокращение от run-length encoding – кодирование длин серий).

Последовательная серия одинаковых элементов данных заменяется на два символа: элемент и число его повторений. Широко используется как дополнительный, так и промежуточный метод. В качестве самостоятельного метода применяется, например, в графическом формате BMP.

2. Словарный метод LZ (сокращение от Lempel Ziv – имена авторов).

Используется словарь, состоящий из последовательностей данных или слов. При сжатии эти слова заменяются на их коды из словаря. Основным параметром словарного метода является размер словаря: чем больше словарь, тем больше эффективность. Однако для неоднородных данных чрезмерно большой размер может быть вреден, так как при резком изменении типа данных словарь будет заполнен неактуальными словами. Для эффективной работы данного метода при сжатии требуется дополнительная память. Приблизительно на порядок больше, чем нужно для исходных данных словаря. Существенным преимуществом словарного метода является простая и быстрая процедура распаковки. Дополнительная память при этом не требуется. Такая особенность особенно важна, если необходим оперативный доступ к данным.

3. Энтропийный метод (Huffman) – кодирование Хаффмена.

В этом методе элементы данных, которые встречаются чаще, кодируются при сжатии более коротким кодом, а более редкие элементы данных кодируются более длинным кодом. За счет того, что коротких кодов значительно больше, общий размер получается меньше исходного.

Сжиматься могут как один, так и несколько файлов, которые в сжатом виде помещаются в архив.

Архивный файл включает оглавление, содержащее следующую информацию о файлах, хранящихся в архиве: имя файла; сведения о каталоге, в котором он находился; дату и время последней модификации файла; размер файла на диске и в архиве; код циклического контроля для каждого файла, используемый для проверки целостности архива. В качестве расширений архивным файлам по умолчанию устанавливается имя соответствующей программы-архиватора.

Большинство программ-архиваторов позволяют создавать многотомные архивы. Многотомный архив – это последовательность архивных файлов, размер которых не превышает заданного размера тома. Обычно создается при необходимости переноса архива на гибких дисках, если размер архива превышает емкость дискеты. При создании таких архивов архиватор делает паузу для смены дискеты. Многотомный архив имеет общее имя, в расширении же каждого файла многотомного архива указывается номер тома.

Самораспаковывающийся архив – это определенным образом обработанный архивный файл, являющийся выполняемым файлом. Выполнение такого файла приводит к распаковке содержащихся в нем файлов.

Типовые функции программ-архиваторов следующие:

1. Помещение исходных файлов в архив.
2. Извлечение файлов из архива.
3. Удаление файлов из архива.
4. Просмотр оглавления архива.

Для каждого файла из архива в оглавлении архивного файла записывается код циклического контроля (CRC). При извлечении файла код циклического контроля для него вычисляется и сравнивается с тем, что записан в оглавлении архива. При их несовпадении выдается сообщение об ошибке.

В настоящее время наибольшее распространение получили программы-архиваторы, ориентированные на работу под управлением операционной системы Windows. Это winrar и Winzip, отличающиеся большой степенью сжатия, работой с длинными именами файлов и удобным интерфейсом.

Winrar разработан Евгением Рошалом на основе архиватора RAR, созданного им же в 1993г. для операционной системы MS DOS. Он поддерживает обработку многих архивных форматов и использует

оригинальный алгоритм упаковки, особенно эффективный для исполняемых и текстовых файлов. К важным дополнительным возможностям программы относятся: защита архива при помощи пароля; восстановление поврежденных архивов; создание многотомных и самораспаковывающихся архивов; сохранение комментариев к архивам. Пользовательский интерфейс winrar содержит основное меню, панель инструментов и рабочую область, в которой показаны все файлы текущей папки. При работе с winrar архивы воспринимаются как папки, содержимое которых можно просмотреть традиционными способами.

Архиватор формата ZIP (PKZIP) был первоначально создан для MS-DOS в 1989 году компанией Pkware.

Winzip был создан в начале 1990 г. как коммерческий графический интерфейс для PKZIP. Приблизительно в 1996 г. создатели Winzip включили код сжатия от проекта PKZIP, таким образом устранив потребность в присутствии консольной версии.

С версии 6.0 до версии 9.0 зарегистрированные пользователи могли загружать новейшие версии программного обеспечения, используя первоначальную регистрационную информацию и тем самым получая обновления бесплатно. Начиная с версии 10.0 система бесплатного обновления была отключена. Winzip доступна в стандартной и профессиональной версиях.

В мае 2006 г. корпорация Corel, известная своей линией продуктов Wordperfect и Coreldraw, объявила, что завершила приобретение Winzip Computing.

Вопросы и задания для самоконтроля

1. Дайте определение понятия «архивация».
2. Перечислите и охарактеризуйте основные алгоритмы сжатия данных.
3. Каковы типовые функции программ-архиваторов?
4. Охарактеризуйте современные программы-архиваторы.

1.7. Компьютерные вирусы. Антивирусные программы

Компьютерный вирус – это программа, способная создавать свои копии, внедрять их в различные объекты или ресурсы компьютерных систем, сетей и производить определенные действия без ведома пользователя.

Программа, внутри которой находится вирус, называется зараженной (инфицированной). Когда инфицированная программа начинает работу, то сначала управление получает вирус. Вирус заражает другие программы, а также выполняет запланированные деструктивные действия. Для маскировки вирус активизируется не всегда, а лишь при выполнении определенных условий (время, действие). После того, как вирус выполнит нужные ему действия, он передает управление той программе, в которой он находится.

Подобно настоящим вирусам, компьютерные вирусы прячутся, размножаются и ищут возможность перейти на другие ЭВМ.

Различные вирусы выполняют различные деструктивные действия:

- выводят на экран мешающие текстовые сообщения;
- создают звуковые эффекты;
- создают видеоэффекты;
- замедляют работу ЭВМ, постепенно уменьшают объем оперативной памяти;
- увеличивают износ оборудования;
- вызывают отказ отдельных устройств, зависание или перезагрузку компьютера и крах работы всей ЭВМ;
- имитируют повторяющиеся ошибки работы операционной системы;
- уничтожают FAT-таблицу, форматируют жесткий диск, стирают BIOS, стирают или изменяют установки в CMOS, стирают секторы на диске, уничтожают или искажают данные, стирают антивирусные программы;
- осуществляют научный, технический, промышленный и финансовый шпионаж;
- выводят из строя системы защиты информации, дают злоумышленникам тайный доступ к вычислительной машине;
- делают незаконные отчисления с каждой финансовой операции и т. д.

Главная опасность самовоспроизводящихся кодов заключается в том, что программы-вирусы начинают жить собственной жизнью, практически не зависящей от разработчика программы. Так же, как в цепной реакции в ядерном реакторе, запущенный процесс трудно остановить.

Основные симптомы вирусного заражения ЭВМ следующие: замедление работы некоторых программ; увеличение размеров файлов; появление не существовавших ранее файлов; уменьшение объема

доступной оперативной памяти; появление сбоев в работе операционной системы; запись информации на диски в моменты, когда этого не должно происходить.

Рассмотрим основные виды вирусов. Существуют различные классификации вирусов.

По среде обитания – это файловые, загрузочные, загрузочно-файловые вирусы. Файловые инфицируют исполняемые файлы, имеющие расширение *exe* и *com*. К этому же классу относятся и макровирусы, написанные с помощью макрокоманд. Они заражают рабочие файлы программ, поддерживающих макросы. Преимущественно это программы пакета Microsoft Office. Загрузочные внедряются в загрузочный сектор устройств внешней памяти или в сектор, содержащий программу загрузки системного диска. Некоторые вирусы записываются в свободные секторы диска, помечая их в файловой-таблице как нерабочие. Загрузочно-файловые интегрируют черты последних двух групп.

По способу заражения (активизации) – это резидентные и нерезидентные вирусы. Резидентный вирус логически можно разделить на две части – инсталлятор и резидентный модуль. При запуске инфицированной программы управление получает инсталлятор, который выполняет следующие действия:

- размещает резидентный модуль вируса в ОЗУ и выполняет операции, необходимые для того, чтобы последний хранился в ней постоянно;
- подменяет некоторые обработчики прерываний, чтобы резидентный модуль мог получать управление при возникновении определенных событий.

Нерезидентные вирусы не заражают оперативную память и проявляют свою активность лишь однократно, при запуске инфицированной программы.

По степени опасности вирусы могут быть неопасные, не повреждающие информацию и не влияющие на работу программ. Иногда они воспроизводят различные звуковые и видеоэффекты. Основной вред от них – это уменьшение свободного места на устройстве внешней памяти за счет размножения вируса. Опасные вирусы способны дестабилизировать работу операционной системы или отдельных программ; очень опасные могут повредить или уничтожить файлы на устройстве внешней памяти или повредить некоторые устройства компьютера.

По особенностям алгоритма вирусы делятся на компаньон-вирусы, паразитические, черви, невидимки (стелс-вирусы) и полиморфные вирусы.

Компаньон-вирусы не изменяют файлов. Алгоритм их работы состоит в том, что они создают для ехе-файлов новые файлы-спутники (дубликаты), имеющие то же имя, но с расширением com. (com-файл обнаруживается первым, а затем вирус запускает ехе-файл).

Паразитические при распространении своих копий обязательно изменяют содержимое дисковых секторов или файлов (все вирусы кроме компаньонов и червей).

Черви (репликаторы) аналогично компаньонам не изменяют файлы и секторы диска. Они проникают в компьютер по сети, вычисляют сетевые адреса других компьютеров и рассылают по этим адресам свои копии. Черви уменьшают пропускную способность сети, замедляют работу серверов.

Невидимки (стелс-вирусы) используют набор средств для маскировки своего присутствия в ЭВМ. Их трудно обнаружить, так как они перехватывают обращения ОС к пораженным файлам или секторам и подставляют незараженные участки файлов.

Полиморфные вирусы (мутанты) шифруют собственный код различными способами. Их трудно обнаружить, так как их копии практически не содержат полностью совпадающих участков кода.

Троянская программа маскируется под полезную или интересную программу, выполняя при запуске еще и функции вируса. Троянские программы предназначены в основном для кражи конфиденциальной информации или для удаленного администрирования. В отличие от вирусов, троянские программы не обладают свойством самовоспроизводства.

По целостности вирусы бывают монолитные, представляющие единый блок, и распределенные, когда программа разделена на части. Эти части содержат инструкции, которые указывают, как собрать их воедино, чтобы воссоздать вирус.

Антивирус – это программный продукт, выполняющий одну либо несколько из следующих функций: 1) защиту данных от разрушения; 2) обнаружение вирусов; 3) нейтрализацию вирусов; 4) контроль обмена данными через компьютерную сеть.

В состав антивирусной программы могут входить следующие модули:

Программы-детекторы рассчитаны на обнаружение конкретных, заранее известных программе вирусов и основаны на сравнении ха-

рактерной последовательности байтов (сигнатур), содержащихся в теле вируса, с байтами проверяемых программ. Программы-детекторы снабжаются блоками эвристического анализа. В этом режиме делается попытка обнаружить новые или неизвестные вирусы по характерным для всех вирусов кодовым последовательностям.

Программы-дезинфекторы (фаги) не только находят зараженные файлы, но и лечат их, удаляя из файла тело программы-вируса.

Программы-ревизоры анализируют текущее состояние файлов и системных областей диска и сравнивают его с информацией, сохраненной ранее. При этом проверяется состояние загрузочного сектора, файловой таблицы, а также длина файлов, их время создания, атрибуты, контрольные суммы. Программы-фильтры (мониторы) оповещают пользователя обо всех попытках какой-либо программы выполнить подозрительные действия. Фильтры контролируют обновление программных файлов и системной области диска, форматирование диска, резидентное размещение программ в ОЗУ.

Рассмотрим основные меры по защите ЭВМ от заражения вирусами.

Необходимо оснастить ЭВМ современными антивирусными программами и постоянно обновлять их версии.

При работе в сети обязательно должна быть установлена программа-фильтр.

Перед считыванием информации с носителей, записанных на других ЭВМ, следует всегда проверять их на наличие вирусов.

При переносе файлов в архивированном виде необходимо их проверять сразу же после разархивации.

При работе на других компьютерах необходимо защищать свои носители информации от записи.

Делать архивные копии ценной информации на других носителях.

Не оставлять подключенными носители при включении или перезагрузке ЭВМ, это может привести к заражению загрузочными вирусами.

Получив электронное письмо, к которому приложен исполняемый файл, не следует запускать этот файл без предварительной проверки.

Вопросы и задания для самоконтроля

1. Дайте определение понятия «компьютерный вирус».
2. Каковы основные признаки заражения компьютера вирусами?

3. Классифицируйте компьютерные вирусы по среде обитания.
4. Классифицируйте компьютерные вирусы по способу заражения.
5. Классифицируйте компьютерные вирусы по деструктивным возможностям.
6. Перечислите и охарактеризуйте основные модули антивирусных программ.

1.8. Компьютерные сети. Особенности построения. Назначение и классификация

Появление персональных компьютеров потребовало нового подхода к организации системы обработки данных и созданию новых информационных технологий. Возникла потребность перехода от использования отдельных ЭВМ в системах централизованной обработки данных к распределенной обработке данных.

Распределенная обработка данных – это обработка данных, выполняемая на независимых, но связанных между собой компьютерах, представляющих распределенную систему.

Компьютерная (вычислительная) сеть – это совокупность компьютеров и терминалов, соединенных с помощью каналов связи в единую систему, удовлетворяющую требованиям распределенной обработки данных.

Абонентами сети (т. е. объектами, генерирующими или потребляющими информацию в сети) могут быть отдельные компьютеры, комплексы ЭВМ, терминалы, промышленные роботы, станки с числовым программным управлением и т. д.

В зависимости от территориального расположения абонентов компьютерные сети делятся:

- на глобальные, объединяющие абонентов, расположенных в различных странах, на различных континентах. Глобальные вычислительные сети позволяют решить проблему объединения информационных ресурсов человечества и организации доступа к этим ресурсам;
- региональные, связывающие абонентов, расположенных на значительном расстоянии друг от друга. Они могут включать абонентов большого города, экономического региона, отдельной страны;

- локальные, объединяющие абонентов, расположенных в пределах небольшой территории. К классу локальных сетей относятся сети отдельных предприятий, фирм, офисов и т. д.

Объединение глобальных, региональных и локальных компьютерных сетей позволяет создавать многосетевые иерархии, обеспечивающие мощные средства обработки огромных информационных массивов и доступ к неограниченным информационным ресурсам.

В общем случае компьютерная сеть представляется совокупностью трех вложенных друг в друга подсистем: сети рабочих станций, сети серверов и базовой сети передачи данных.

Рабочая станция (клиентская машина, рабочее место, абонентский пункт, терминал) – это компьютер, за которым непосредственно работает абонент компьютерной сети. Сеть рабочих станций представлена совокупностью рабочих станций и средств связи, обеспечивающих взаимодействие рабочих станций с сервером и между собой.

Сервер – это компьютер, выполняющий общие задачи компьютерной сети и предоставляющий услуги рабочим станциям. Сеть серверов – это совокупность серверов и средств связи, обеспечивающих подключение серверов к базовой сети передачи данных.

Базовая сеть передачи данных – это совокупность средств передачи данных между серверами. Она состоит из каналов связи и узлов связи. Узел связи – это совокупность средств коммутации и передачи данных в одном пункте. Узел связи принимает данные, поступающие по каналам связи, и передает данные в каналы, ведущие к абонентам.

Базовыми требованиями, определяющими архитектуру компьютерных сетей, являются следующие:

- открытость – возможность включения дополнительных компьютеров, терминалов, узлов и линий связи без изменения технических и программных средств существующих компонентов;
- живучесть – сохранение работоспособности при изменении структуры;
- адаптивность – допустимость изменения типов компьютеров, терминалов, линий связи, операционных систем;
- эффективность – обеспечение требуемого качества обслуживания пользователей при минимальных затратах;
- безопасность информации – это способность сети обеспечить защиту информации от несанкционированного доступа.

Указанные требования обеспечиваются модульной организацией управления процессами в сети, реализуемой по многоуровневой схеме. Число уровней и распределение функций между ними существенно влияет на сложность программного обеспечения компьютеров, входящих в сеть, и на эффективность сети. Формальной процедуры выбора числа уровней не существует. Классической является семиуровневая схема.

Уровень 1 – физический – реализует управление каналом связи, что сводится к подключению и отключению канала связи и формированию сигналов, представивших передаваемые данные.

Уровень 2 – канальный – обеспечивает надежную передачу данных через физический канал, организованный на уровне 1.

Уровень 3 – сетевой – обеспечивает выбор маршрута передачи сообщений по линиям, связывающим узлы сети.

Уровни 1 – 3 организуют базовую сеть передачи данных как систему, обеспечивающую надежную передачу данных между абонентами сети.

Уровень 4 – транспортный – обеспечивает сопряжение абонентов сети с базовой сетью передачи данных.

Уровень 5 – сеансовый – организует сеансы связи на период взаимодействия процессов. На этом уровне по запросам процессов создаются порты для приема и передачи сообщений и организуются соединения – логические каналы.

Уровень 6 – представительный – осуществляет трансформацию различных языков, форматов данных и кодов для взаимодействия разнотипных компьютеров.

Уровень 7 – прикладной – обеспечивает поддержку прикладных процессов пользователей. Порядок реализации связей в сети регулируется протоколами. Протокол – это набор коммутационных правил и процедур по формированию и передаче данных в сети.

Базовые принципы организации компьютерной сети определяют ее основные характеристики:

- операционные возможности – перечень основных действий по обработке данных. Абоненты сети имеют возможность использовать память и процессоры многих компьютеров для хранения и обработки данных. Предоставляемая компьютерной сетью возможность параллельной обработки данных многими компьютерами и дублирования необходимых ресурсов

позволяет сократить время решения задач, повысить надежность системы и достоверность результатов;

- производительность – суммарная производительность компьютеров, участвующих в решении задачи пользователя;
- время доставки сообщений – определяется как статистическое среднее время от момента передачи сообщения в сеть до момента получения сообщения адресатом;
- стоимость предоставляемых услуг.

Локальная вычислительная сеть объединяет абонентов, находящихся на небольшом расстоянии друг от друга (в пределах 10 – 15 км). Обычно такие сети строятся в пределах одного предприятия или организации.

Информационные системы, построенные на базе локальных вычислительных сетей, обеспечивают решение задач хранения данных, обработки данных, организации доступа пользователей к данным, передачи данных и результатов их обработки пользователям.

Компьютерные сети реализуют распределенную обработку данных. Здесь обработка данных распределяется между двумя объектами: клиентом и сервером. В процессе обработки данных клиент формирует запрос к серверу на выполнение сложных процедур. Сервер выполняет запрос и результаты выполнения передает клиенту. Сервер обеспечивает хранение данных общего пользования, организует доступ к этим данным и передает их клиенту. Подобная модель вычислительной сети получила название архитектуры клиент – сервер.

По признаку распределения функций локальные компьютерные сети делятся на одноранговые и двухранговые (иерархические сети или сети с выделенным сервером).

В одноранговой сети компьютеры равноправны по отношению друг к другу. Каждый пользователь в сети решает сам, какие ресурсы своего компьютера он предоставит в общее пользование. Таким образом, компьютер выступает и в роли клиента, и в роли сервера. Одноранговое разделение ресурсов является вполне приемлемым для малых офисов с 5 – 10 пользователями, объединяя их в рабочую группу.

Двухранговая сеть организуется на основе сервера, на котором регистрируются пользователи сети.

Для современных компьютерных сетей типичной является смешанная сеть, объединяющая рабочие станции и серверы, причем часть рабочих станций образует одноранговые сети, а другая часть принадлежит двухранговым сетям.

Геометрическая схема соединения (конфигурация физического подключения) узлов сети называется топологией сети. Существует большое количество вариантов сетевых топологий, базовыми из которых являются шина, кольцо, звезда.

1. Шина. Канал связи (КС), объединяет узлы в сеть, образуя ломаную линию – шину. Любой узел может принимать информацию в любое время, а передавать – только тогда, когда шина свободна. Данные (сигналы) передаются компьютером на шину. Каждый компьютер проверяет их, определяя, кому адресована информация, и принимает данные, если они посланы ему, либо игнорирует. Если компьютеры расположены близко друг от друга, то организация КС с шинной топологией недорога и проста. Необходимо просто проложить кабель от одного компьютера к другому. Затухание сигнала с увеличением расстояния ограничивает длину шины и, следовательно, число компьютеров, подключенных к ней. Проблемы шинной топологии возникают, когда происходит разрыв (нарушение контактов) в любой точке шины; сетевой адаптер одного из компьютеров выходит из строя и начинает передавать на шину сигналы с помехами, тогда необходимо подключить новый компьютер.

2. Кольцо. Узлы объединены в сеть замкнутой кривой. Передача данных осуществляется только в одном направлении. Каждый узел помимо всего прочего реализует функции ретранслятора. Он принимает и передает сообщения, а воспринимает только обращенные к нему. Используя кольцевую топологию, можно присоединить к сети большое количество узлов, решив проблемы помех и затухания сигнала средствами сетевой платы каждого узла. Недостатки кольцевой организации: разрыв в любом месте кольца прекращает работу всей сети; время передачи сообщения определяется временем последовательного срабатывания каждого узла, находящегося между отправителем и получателем сообщения; из-за прохождения данных через каждый узел существует возможность непреднамеренного искажения информации.

3. Звезда. Узлы сети объединены с центром лучами. Вся информация передается через центр, что позволяет относительно просто выполнять поиск неисправностей и добавлять новые узлы без прерывания работы сети. Однако расходы на организацию каналов связи здесь обычно выше, чем у шины и кольца.

Комбинация базовых топологий – гибридная топология – обеспечивает получение широкого спектра решений, аккумулирующих достоинства и недостатки базовых.

Кроме проблем создания локальных вычислительных сетей имеется также проблема расширения (объединения) компьютерных сетей. Дело в том, что созданная на определенном этапе развития информационной системы вычислительная сеть со временем может перестать удовлетворять потребности всех пользователей. В то же время физические свойства сигнала, каналов передачи данных и конструктивные особенности сетевых компонент накладывают жесткие ограничения на количество узлов и геометрические размеры сети.

Для объединения локальных вычислительных сетей применяются следующие устройства.

Повторитель – устройство, обеспечивающее усиление и фильтрацию сигнала без изменения его информативности. По мере передвижения по линиям связи сигналы затухают. Для уменьшения влияния затухания используются повторители. Причем повторитель не только копирует или повторяет принимаемые сигналы, но и восстанавливает характеристики сигнала: усиливает сигнал и уменьшает помехи.

Мост – устройство, выполняющее функции повторителя для тех сигналов (сообщений), адреса которых удовлетворяют заранее наложенным ограничениям. Одной из проблем больших сетей является напряженный сетевой трафик (поток сообщений в сети). Эта проблема может решаться следующим образом. Компьютерная сеть делится на сегменты. Передача сообщений из сегмента в сегмент осуществляется только целенаправленно, если абонент одного сегмента передает сообщение абоненту другого сегмента. Мост является устройством, ограничивающим движение по сети и не позволяющим сообщениям попадать из одной сети в другую без подтверждения права на переход.

Мосты бывают локальные и удаленные.

Локальные мосты соединяют сети, расположенные на ограниченной территории в пределах уже существующей системы.

Удаленные мосты соединяют сети, разнесенные территориально, с использованием каналов связи и модемов.

Локальные мосты, в свою очередь, разделяются на внутренние и внешние.

Внутренние мосты обычно располагаются на одном компьютере и совмещают функцию моста с функцией абонентской ЭВМ. Расширение функций осуществляется путем установки дополнительной сетевой платы.

Внешние мосты предусматривают использование отдельного компьютера со специальным программным обеспечением.

Маршрутизатор – это устройство, соединяющее сети разного типа, но использующие одну операционную систему. Это, по сути, тот же мост, но имеющий свой сетевой адрес. Используя возможности адресации маршрутизаторов, узлы в сети могут посылать маршрутизатору сообщения, предназначенные для другой сети. Для поиска лучшего маршрута к любому адресату в сети используются таблицы маршрутизации. Эти таблицы могут быть статическими и динамическими.

Шлюз – специальный аппаратно-программный комплекс, предназначенный для обеспечения совместимости между сетями, использующими различные протоколы взаимодействия. Шлюз преобразует форму представления и форматы данных при передачи их из одного сегмента в другой. Шлюз осуществляет свои функции на уровне выше сетевого. Он не зависит от используемой передающей среды, но зависит от используемых протоколов обмена данными. Обычно шлюз выполняет преобразования между протоколами.

С помощью шлюзов можно подключить локальную вычислительную сеть к главному компьютеру, а также к глобальной вычислительной сети.

Вопросы и задания для самоконтроля

1. Дайте определение понятия «компьютерная сеть».
2. Назовите и охарактеризуйте основные топологии сетей.
3. Поясните назначение устройства «мост».
4. Каково назначение устройства «коммутатор» (Switch)?
5. Каково назначение устройства «шлюз»?
6. Дайте определение понятий «клиент» и «сервер».

1.9. Глобальная сеть Интернет.

Общая характеристика, особенности построения

Интернет представляет собой глобальную компьютерную сеть, соединяющую отдельные сети. Интернет обеспечивает обмен информацией между всеми компьютерами, которые входят в сети, подключенные к ней. Тип компьютера и используемая им операционная система значения не имеют.

Соединение сетей обладает громадными возможностями. Интернет предоставляет в распоряжение своих пользователей множество всевозможных ресурсов. Для того чтобы информация передавалась между компьютерами независимо от используемых линий связи, типа ЭВМ и программного обеспечения, разработаны специальные протоколы передачи данных. Они работают по принципу разбиения данных на блоки определенного размера (пакеты), которые последовательно отсылаются адресату. В Интернете используются два основных протокола: межсетевой протокол IP разделяет передаваемые данные на отдельные пакеты и снабжает их заголовками и указанием адреса получателя, а протокол управления передачей TCP отвечает за правильную доставку пакета. Так как эти протоколы взаимосвязаны, обычно говорят о протоколе TCP/IP.

Основные ячейки Интернета – локальные вычислительные сети. Это означает, что Интернет не просто устанавливает связь между отдельными компьютерами, а создает пути соединения для более крупных единиц – групп компьютеров. Если некоторая локальная сеть подключена к Интернету, то каждая рабочая станция этой сети также может подключаться к Интернету. Существуют также компьютеры, самостоятельно подключенные к Интернету. Они называются хост-компьютерами.

Каждый подключенный к сети компьютер имеет свой адрес, по которому его может найти абонент из любой точки света. К адресам станций предъявляются специальные требования: адрес должен иметь формат, позволяющий вести его обработку автоматически, и нести информацию о своем владельце. С этой целью для каждого компьютера устанавливаются два адреса: цифровой IP-адрес и доменный адрес. Первый из них более понятен компьютеру, второй – человеку. Оба эти адреса могут применяться равноправно.

Цифровой адрес имеет длину 32 бита. Он разделяется точками на 4 блока по 8 битов каждый, которые можно записать в виде десятичного числа, не превышающего значение 255. Адрес содержит полную информацию, необходимую для идентификации компьютера. Два блока определяют адрес сети, третий определяет адрес подсети и четвертый – адрес компьютера внутри заданной сети.

Доменный адрес определяет область, представляющую ряд хост-компьютеров. Этот адрес читается в обратном порядке: вначале указывается имя компьютера, а затем имя сети, в которой он находится. Для упрощения связи абонентов сети все ее адресное пространство

разбито на отдельные области – домены. В системе адресов Интернета приняты домены, представленные географическими регионами. Они имеют имя, состоящее из двух букв. Существуют домены, разделенные по тематическим признакам. Такие домены имеют трехбуквенное сокращенное название.

Компьютерное имя включает как минимум два уровня доменов. Уровни отделяются друг от друга точкой. Слева указывается домен верхнего уровня. Все имена, находящиеся слева, – поддомены общего домена.

Для обработки пути поиска в доменах имеются специальные серверы имен. Они преобразуют доменное имя в специальный цифровой адрес. Такие серверы называются DNS.

Использование технологий Интернета необязательно реализуется в рамках Всемирной информационной сети. Технологии, применяемые в глобальной сети, пригодны и для создания мощных корпоративных информационных систем и систем обеспечения коллективной работы. Интранет – это корпоративная сеть (сеть предприятия или офиса), использующая технологии и продукты Интернета для хранения данных, связи и доступа к информации.

История появления и развития Интернета. Своим зарождением Интернет обязан Министерству обороны США и его секретному исследованию, проводимому в 1969 г. с целью тестирования методов, позволяющих компьютерным сетям выжить во время военных действий с помощью динамической перемаршрутизации сообщений. Первой такой сетью была ARPAnet, объединившая три сети в Калифорнии с сетью в штате Юта по набору правил, названных интернет-протоколом (Internet Protocol или сокращенно IP).

В 1972 г. был открыт доступ для университетов и исследовательских организаций, в результате чего сеть стала объединять 50 университетов и исследовательских организаций, имевших контракты с Министерством обороны США.

В 1973 г. сеть выросла до международных масштабов, объединив сети, находящиеся в Англии и Норвегии. Десятилетие спустя IP был расширен за счет набора коммуникационных протоколов, поддерживающих как локальные, так и глобальные сети. Так появился TCP/IP. Вскоре после этого National Science Foundation (NSF) открыла NSFnet с целью связать 5 суперкомпьютерных центров. Одновременно с внедрением протокола TCP/IP новая сеть вскоре заменила ARPAnet в качестве «хребта» (backbone) Интернета.

Как же Интернет стал столь популярен и развит? Толчок к этому, а также к превращению его в среду для ведения бизнеса дало появление World Wide Web – Всемирной паутины (WWW, 3W, вэ-вэ-вэ, три даблью), системы гипертекста (hypertext), которая сделала путешествие по сети Интернет быстрым и интуитивно понятным.

А вот идея связывания документов через гипертекст впервые была предложена и продвигалась Тедом Нельсоном (Ted Nelson) в 1960 г., однако уровень существующих в то время компьютерных технологий не позволял воплотить ее в жизнь, хотя кто знает, чем бы всё закончилось, если бы эта идея нашла применение?!

Основы того, что мы сегодня понимаем под WWW, заложил в 1980-е гг. Тим Бернерс-Ли (Tim Berners-Lee) в процессе работ по созданию системы гипертекста в Европейской лаборатории физики элементарных частиц (European Laboratory for Particle Physics).

В результате этих работ в 1990 г. научному сообществу был представлен первый текстовый браузер (browser), позволяющий просматривать связанные гиперссылками (hyperlinks) текстовые файлы on-line. Доступ к этому браузеру широкой публике был предоставлен в 1991 г., однако распространение его вне научных кругов шло медленно.

Новым историческим этапом в развитии Интернет обязан выходу первой Unix-версии графического браузера Mosaic в 1993 г., разработанного в 1992 г. Марком Андресеном (Marc Andreessen), студентом, стажировавшимся в Национальном центре суперкомпьютерных приложений (National Center for Supercomputing Applications, NCSA) США.

С 1994 г. после выхода версий браузера Mosaic для операционных систем Windows и Macintosh, а вскоре вслед за этим браузеров Netscape Navigator и Microsoft Internet Explorer берет начало взрывообразное распространение популярности WWW и, как следствие Интернета, среди широкой публики сначала в США, а затем и по всему миру.

В 1995 г. NSF передала ответственность за Интернет в частный сектор, и с этого времени Интернет существует в том виде, каким мы знаем его сегодня.

Некоторые сервисы Интернета:

WWW (World Wide Web) – Всемирная паутина, гипермедиа-данные. Работает по протоколу HTTP – HyperText Transfer Protocol. Доступна конечному пользователю через программу просмотра гипермедиа – браузер;

FTP (File Transfer Protocol) – доступ к файловым архивам. Работает по протоколу FTP. Доступна конечному пользователю через браузер и специализированные программы;

E-mail (electronic mail) – электронная почта. Работает по протоколам SMTP (Simple Mail Transfer Protocol) и POP3(Post Office Protocol), которые соответственно отвечают за передачу и прием почты. Доступна конечному пользователю через браузер и специализированные программы;

News – эхо-конференции или публично доступная тематическая переписка. Работает по протоколу NNTP – News Network Transfer Protocol. Доступна конечному пользователю через браузер и специализированные программы;

IRC – (Internet Relay Chat). Работает по протоколу IRC. Доступна конечному пользователю через IRC-клиент;

Telnet – (TELEcommunication NETwork) – терминальный доступ к удаленным серверам. Работает по протоколу TCP. Используется для передачи данных с различной управляющей информацией. Для работы с TELNET используются специальные программы-терминалы. Обычно они содержатся в пакете программ операционной системы.

Как отдельный сервис можно выделить поисковые системы.

Поисковая система – веб-сайт, предоставляющий возможность поиска информации в Интернете. Большинство поисковых систем ищут информацию на сайтах Всемирной паутины, но существуют также системы, способные искать файлы на ftp-серверах, товары в интернет-магазинах, а также информацию в группах новостей.

Вопросы и задания для самоконтроля

1. Расскажите об истории возникновения и развития Интернета. Каков Интернет сегодня?
2. Перечислите и охарактеризуйте основные сервисы Интернета.
3. Расскажите об истории возникновения и развития сервиса WWW.
4. Поясните основные принципы функционирования Интернета.
5. Что такое домен?
6. Объясните понятия «IP-адрес», «доменный адрес».

ГЛАВА 2. MICROSOFT WORD

2.1. Знакомство с программой Microsoft Word 2007

Создание текстовых документов – наиболее распространенный способ использования компьютеров. Трудно представить сферу деятельности, в которой компьютер не применяется для подготовки и печати документов. Огромный спрос на удобные и универсальные приложения для создания документов побудил производителей программного обеспечения выпускать и развивать свои продукты. Microsoft Word – бесспорный лидер в данной области. Разработчики этой программы предусмотрели, казалось бы, все, с чем вы можете столкнуться на этапах создания документа.

Новый интерфейс программы Microsoft Word версии 2007 может показаться пользователям более ранних версий непривычным, но через некоторое время вы поймете и оцените, насколько удобно расположены элементы интерфейса.

Для запуска Microsoft Word 2007 выполните команду *Пуск -> Все программы -> Microsoft Office -> Microsoft Office Word 2007*.

Если вам когда-нибудь приходилось работать с предыдущими версиями Microsoft Word или просто видеть их, то при первом запуске программы вы сразу обратите внимание, что ее интерфейс изменился (рис. 2.1). Теперь в Word отсутствуют типичные для программных окон элементы – главное меню с раскрывающимися разделами, которые содержат ссылки на команды, и панели инструментов с кнопками наиболее часто используемых программ. Разработчики попытались предельно упростить интерфейс программы, сделав его максимально интуитивным.

Главное меню Word представлено в виде вкладок. При щелчке на любой из них содержимое вкладки открывается ниже. Оно представляет собой инструменты, расположенные на горизонтальной ленте. Сходные по действию инструменты объединены в группы. Название группы указывает на то, с какими объектами или параметрами работают ее кнопки.

Изначально состав вкладок главного меню фиксирован и изменить его вручную нельзя, однако в процессе работы он может пополняться новыми вкладками, которые будут открывать на ленте перечень инстру-

ментов для выполнения поставленной задачи. Так, при попытке нарисовать таблицу появятся две новые вкладки *Конструктор* и *Макет*, инструменты которых отвечают за создание и форматирование таблиц.

Все команды, которые раньше были запрятаны в многочисленные подменю главного меню, вынесены на ленту в виде кнопок, и ориентироваться в них стало гораздо проще. По внешнему виду лента напоминает панель инструментов, однако, в отличие от последней, изменить ее состав нельзя.

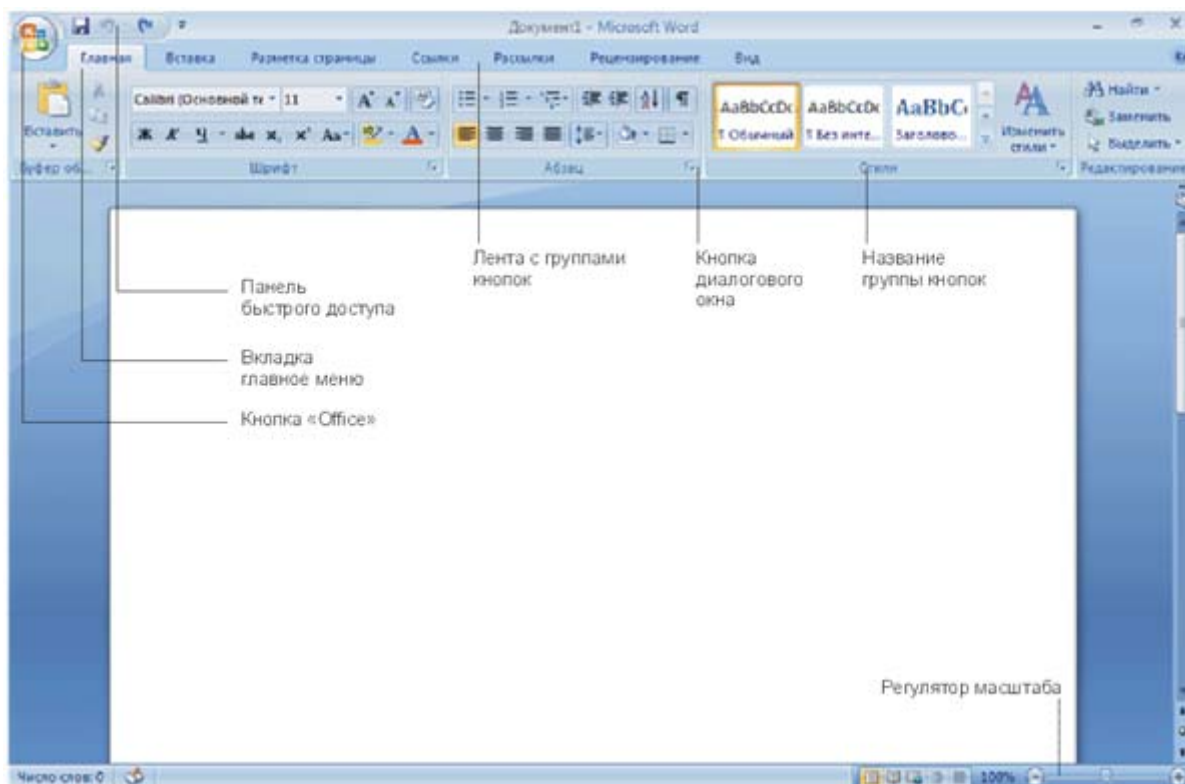


Рис. 2.1. Окно Microsoft Word 2007

В нижнем правом углу каждой группы находится кнопка диалогового окна. Она открывает стандартное окно, где в привычном для Windows виде представлены параметры, выведенные на ленту в виде кнопок, а также некоторые дополнительные параметры настройки, необходимость обращения к которым возникает достаточно редко.

Справа в заголовке окна располагается панель быстрого доступа. На нее вынесены кнопки наиболее часто используемых команд, таких как сохранение документа, отмена последнего действия и возврат к предыдущему действию. В отличие от ленты вы можете менять содержимое панели быстрого доступа, добавляя на нее новые кнопки и удаляя редко используемые. Для этого нужно щелкнуть на стрелке

справа от панели быстрого доступа, в раскрывшемся меню щелчками выбрать кнопки для добавления или удаления (уже имеющиеся на панели кнопки помечены флажками).

Практически в любом приложении для того, чтобы узнать назначение кнопки панели инструментов, достаточно привести на нее указатель мыши и задержать его до появления всплывающей подсказки. В Microsoft Word 2007 всплывающая подсказка содержит не только краткую информацию о назначении кнопки, но и подробное описание действия, которое можно выполнить с ее помощью, и конечно-го результата (рис. 2.2).

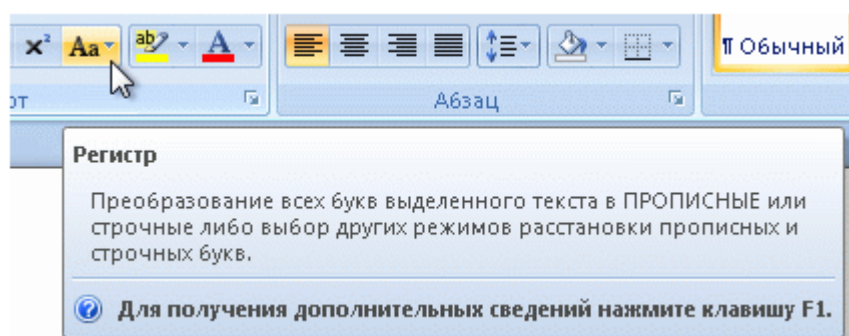


Рис. 2.2. Всплывающая подсказка появляется при наведении указателя мыши на кнопку

В верхнем левом углу окна Word имеется большая кнопка с логотипом Microsoft Office. Она открывает меню, состав которого аналогичен меню *Файл*. Здесь вы сможете найти основные команды, отвечающие за создание, открытие, сохранение и распечатку документа, некоторые дополнительные команды, список последних открытых документов, а также кнопку *Параметры Word*, открывающую окно настроек программы.

Для описания всех возможностей Microsoft Word понадобится не одна сотня страниц. Остановимся на основных моментах, которые должен знать любой пользователь, чтобы суметь правильно набрать и отформатировать текст, добавить к нему рисунки, таблицы, математические формулы и подготовить документ к распечатке.

Основные команды работы с документами находятся в меню, открываемом кнопкой *Office*.

При сохранении документа, созданного в Microsoft Word 2007, вы должны учитывать одну важную особенность: между Microsoft Word 2007 и предыдущими версиями программы нет обратной совместимости. Если вы сохраните файл в формате, используемом

Microsoft Word 2007 по умолчанию, то этот документ нельзя будет открыть на компьютере, на котором установлена одна из предыдущих версий Word без специальных конвертеров. Помните об этом при отправке документа коллегам и друзьям или, если собираетесь работать с ним на другом компьютере, используйте для сохранения формат, поддерживаемый предыдущими версиями Word. Напротив, документы, созданные в Word более ранних версий, без проблем открываются в Word 2007.

Для сохранения документа нажмите Кнопку «Office» и наведите указатель мыши на подменю *Сохранить как*. В развернувшемся меню выберите формат файла: если документ потребуется открыть на другом компьютере, на котором установлена одна из предыдущих версий Word, выполните команду *Документ Word 97–2003*; если перемещение файла за пределы данного компьютера не планируется, сохраните его в стандартном формате Word 2007, выбрав в поле *Тип файла* – *Документ Word*. В окне сохранения выберите папку, в которую будет помещен файл, введите имя и нажмите кнопку *Сохранить*.

Открыть любой документ Word можно стандартными способами:

- щелкните дважды на файле *Word*;
- запустите Microsoft Word 2007, нажмите кнопку *Office* и выполните команду *Открыть*. В появившемся окне перейдите к папке, в которой хранится документ, выделите его щелчком кнопки мыши и нажмите кнопку *Открыть*; щелкните на соответствующей ссылке в разделе *Последние документы* в меню кнопки *Office*, если документ ранее открывался.

Для создания чистого бланка документа выполните команду *Создать* в меню кнопки *Office*. В открывшемся окне щелкните кнопкой мыши на эскизе *Новый документ* и нажмите кнопку *Создать*. При запуске программы новый документ будет создан автоматически. Слева в верхней части страницы, находящейся прямо перед вами, мигает курсор, указывая на то, что вы можете сразу приступить к набору текста.

Вопросы и задания для самоконтроля

1. Назовите принципиальные отличия интерфейсов Microsoft Word 2007 и более ранних версий.
2. По каким принципам группируются команды в интерфейсе Microsoft Word 2007?
3. Опишите процедуры создания, открытия и сохранения документов в Microsoft Word 2007.

2.2. Как не следует набирать текст


В данном параграфе, чтобы предостеречь вас от типичных ошибок, будут описаны заведомо неверные действия, которые совершают начинающие пользователи при наборе текста в Word.

До начала работы для удобства разверните окно Word во весь экран. С помощью регулятора масштаба, который находится в нижнем правом углу окна, вы можете увеличить изображение страницы. Имейте в виду, что увеличение масштаба никак не влияет на реальный размер шрифта. Он меняется другим способом, о котором будет рассказано далее.

Многим начинающим пользователям кажется, что символы на некоторых участках набираемого текста расположены очень близко друг к другу, поэтому они пытаются искусственно увеличить расстояние между ними, вставляя лишние пробелы. Не делайте этого, так как программа Word обладает достаточным «интеллектом», чтобы обеспечить равномерное расстояние между символами.

Пытаясь выровнять правую границу текста, многие искусственно переходят на новую строку помощью клавиши *Enter*. Этого делать не нужно, поскольку Word автоматически переносит слово на следующую строку при достижении конца текущей строки и делает это в соответствии с установленным типом выравнивания, изменить который можно с помощью специальных инструментов, а не методом подгона. Нажатие клавиши *Enter* приведет к тому, что следующий фрагмент текста, логически связанный с уже набранным, будет обособлен в отдельный абзац.

Абзац в Word – это фрагмент текста, который начинается с новой строки и заканчивается нажатием клавиши *Enter*. В дальнейшем при форматировании абзацев (задании определенных характеристик, таких как величина отступа первой строки, расстояние между строками, интервал между абзацами) логическая структура текста будет потеряна.

Абзацы легко увидеть, включив отображение непечатаемых знаков нажатием на ленте кнопки *Отобразить все знаки* в группе *Абзац*. При этом в тексте вы увидите символы пробелов в виде точек между словами и маркеры конца абзаца . Эти символы не выводятся на печать, но помогают сориентироваться в структуре документа при его создании. Помните: пробел – это такой же символ, как и любая буква, цифра или знак препинания, и с ним можно выполнять стандартные

операции редактирования – удалять лишние, добавлять пропущенный, установив курсор между слившимися словами.

Если вы все-таки обнаружили в своем тексте лишнее разделение на абзацы, объедините их. Для объединения двух абзацев установите курсор в начало нижнего абзаца и нажмите клавишу *Backspace*. Таким же образом можно удалить пустую строку. Для разделения набранного текста на абзацы установите курсор в начале того предложения, которое нужно перенести на новую строку, и нажмите *Enter*. Для выравнивания строки по центру или правому краю многие новички вводят перед ней большое количество пробелов. Регулирование положения текста с помощью пробелов – самая распространенная ошибка среди пользователей. Для выравнивания существуют специальные инструменты, о которых будет рассказано далее. Если вы уже набрали текст, который должен размещаться по центру, оставьте его у левого края и переходите к вводу следующего абзаца.

Отдельного упоминания заслуживает ввод знаков препинания. Не следует обособлять пробелами с обеих сторон точки, запятые, кавычки, скобки, двоеточия, точки с запятой, восклицательные и вопросительные знаки. Знак препинания должен прилегать к символу, после которого (или перед которым) он ставится. Так, чтобы поставить запятую после слова, наберите его, введите запятую и только после этого нажмите *Пробел*. Слова, содержащие дефис, должны набираться без использования пробела (например, «темно-красный»). Тире должно обособляться пробелами с обеих сторон. Для помещения в документ тире, которое имеет большую длину, чем дефис, нажмите сочетание клавиш *Ctrl+Alt+* на цифровом блоке.

Не пытайтесь вручную создать нумерованный список или пронумеровать страницы. Для этих целей в Word имеются специальные средства.

Не беспокойтесь о том, что текст, который нужно набрать, не поместится на единственной странице, которая представлена на экране. Word автоматически переведет вас на следующую страницу, когда количество созданных строк превысит норму одной страницы. Перемещаться между страницами документа можно с помощью колесика мыши, вертикальной полосы прокрутки и клавиш *Page Up* и *Page Down*.

На начальном этапе от вас требуется только набрать текст, разделяя его на абзацы помощью клавиши *Enter*. Не предпринимайте попыток изменить его внешний вид вручную – это не только ухудшит расположение текста в документе, но и приведет к серьезным ошибкам и сложностям при его форматировании с помощью специальных средств Word.

Вопросы и задания для самоконтроля

1. Назовите типичные ошибки начинающих пользователей Microsoft Word.
2. Для чего используется инструмент *Масштаб*?
3. Что такое *Абзац* в Word?
4. Для чего используются непечатаемые символы?

2.3. Редактирование текста

Удаление, копирование и вставка фрагментов текста в Word выполняются стандартными способами. Прежде чем выполнить стандартные операции редактирования, необходимо выделить нужный текстовый фрагмент.



Перемещать и копировать фрагменты текста в Word можно несколькими способами. Самый удобный из них – перетаскивание. Для перемещения текста таким образом щелкните кнопкой мыши на выделенном фрагменте и, удерживая ее нажатой, перемещайте указатель мыши в нужном направлении. Обратите внимание – он изменит свой вид, а при движении за ним будет следовать вертикальная пунктирная черта, которая служит ориентиром для вставки текста в новое место. Расположите эту черту между словами, где должен быть вставлен фрагмент, и отпустите кнопку. Для копирования выделенного фрагмента эти же действия следует проводить, удерживая нажатой клавишу *Ctrl*.

Классический способ перемещения фрагментов текста связан с использованием команд контекстного меню. Щелкните правой кнопкой мыши на выделенном тексте и выполните команду *Вырезать* (при перемещении) или *Копировать* (при копировании) для помещения текста в буфер обмена. Затем щелкните правой кнопкой мыши между символами, где он должен быть помещен, и выполните команду *Вставить*. В отличие от буфера обмена Windows, в который можно поместить только один объект, буфер обмена приложений Microsoft Office позволяет накапливать до 24 текстовых фрагментов: когда вы в процессе работы копируете или вырезаете из текста элементы, последние 24 из них фиксируются в буфере обмена Microsoft Office, чтобы вы при необходимости могли задействовать любой из них для вставки без предварительных операций копирования или вырезания. Увидеть содержимое буфера обмена можно, щелкнув на кнопке диалогового окна в группе *Буфер обмена*. Слева откроется од-

ноименная панель со списком текстовых фрагментов, помещенных в буфер обмена. Чтобы вставить любой из них в документ, установите курсор в нужном месте, щелкните на стрелке справа от фрагмента в списке и выполните команду *Вставить*.

Буфер обмена Microsoft Office – единый для всех приложений пакета Microsoft Office. Объекты, помещенные в буфер обмена в одном окне, доступны на панели *Буфер обмена* другого окна любого приложения Microsoft Office.

Если вы хотите набрать новый фрагмент и добавить его в уже существующий, установите щелчком кнопки мыши курсор между теми предложениями или словами, где он должен быть вставлен, и наберите текст.

Word запоминает не только 24 последних фрагмента, помещенных в буфер обмена, но и все действия, которые вы выполняли на рабочем листе с начала работы с программой, поэтому при совершении ошибки (например при случайном удалении текста или неудачной операции форматирования) не нужно огорчаться. Отмените последнее действие, нажав сочетание клавиш Ctrl+Z или кнопку  *Отменить* на *Панели быстрого доступа*. Стрелка данной кнопки раскрывает список всех действий, которые были выполнены с самого начала работы с документом. Переместившись к действию в списке, можно отменить его и все действия, которые были выполнены после него. Если вы хотите восстановить отмененное действие, воспользуйтесь соседней кнопкой  *Вернуть*. С помощью кнопки *Отменить* можно отказаться только от действия, выполненного с текстом, а не от нажатия какой-либо кнопки на ленте.

Проверка правописания. Набирая текст, вы, безусловно, обратили внимание, что слова, содержащие ошибку, подчеркиваются красной волнистой линией, а некоторые предложения или их фрагменты – зеленой. Это результат работы встроенной системы проверки правописания. При наборе текста она помогает избежать не только случайных опечаток, но и ошибок, связанных с незнанием правил русского языка. Красной волнистой линией в Word подчеркиваются слова, содержащие орфографические ошибки. Грамматически и стилистически некорректные предложения выделяются зеленой волнистой линией.

Word проверяет орфографию и грамматику в документе автоматически по мере ввода текста. Если слово подчеркнуто красной линией, значит, оно отсутствует в орфографическом словаре системы проверки правописания. Щелкните на нем правой кнопкой мыши, чтобы

выбрать в контекстном меню правильный вариант, предлагаемый для замены. В некоторых случаях таких вариантов может быть несколько – в зависимости от того, какую букву можно заменить или добавить для получения грамотно написанного аналога (рис. 2.3).

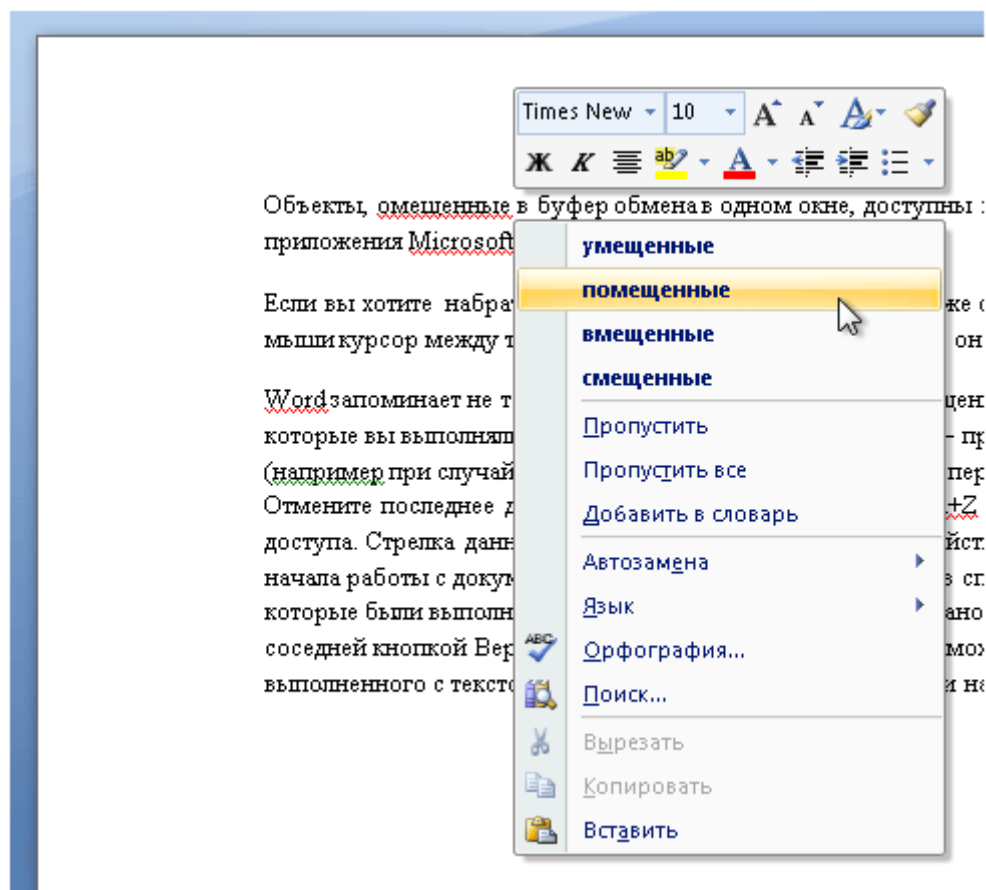


Рис. 2.3. Исправление ошибочного слова с помощью контекстного меню

Несмотря на лексическое богатство встроенного орфографического словаря, при наборе в Word текстов по узкой тематике система часто помечает специальные термины как ошибочные. В этом случае для замены каждого из них Word предлагает совершенно неприемлемые варианты либо информирует о том, что варианты для замены отсутствуют. Во избежание подобных ситуаций вы можете добавить незнакомое системе слово в словарь, если абсолютно уверены в правильности его написания, или указать системе, что в данном документе не нужно подчеркивать определенное слово. Щелкните правой кнопкой мыши на слове и в первом случае выполните команду *Добавить в словарь*, а во втором – команду *Пропустить все*.

Word предлагает варианты для замены незнакомого слова, если в нем допущена одна ошибка. При наличии в одном слове нескольких

ошибок система сообщит в контекстном меню о том, что вариантов для замены нет. Исправьте самостоятельно опечатки и буквы, в написании которых вы точно уверены, а затем еще раз обратитесь к контекстному меню для уточнения букв, написание которых вызывает у вас сомнение. Трактовку выявленных в тексте пунктуационных, синтаксических и стилистических ошибок, также ошибок, связанных с неправильным согласованием слов, можно прочесть в контекстном меню, щелкнув правой кнопкой мыши на подчеркнутом зеленой линией участке предложения. Если команда замечания доступна, вы можете внести исправление в предложение, просто щелкнув на ней (рис. 2.4).

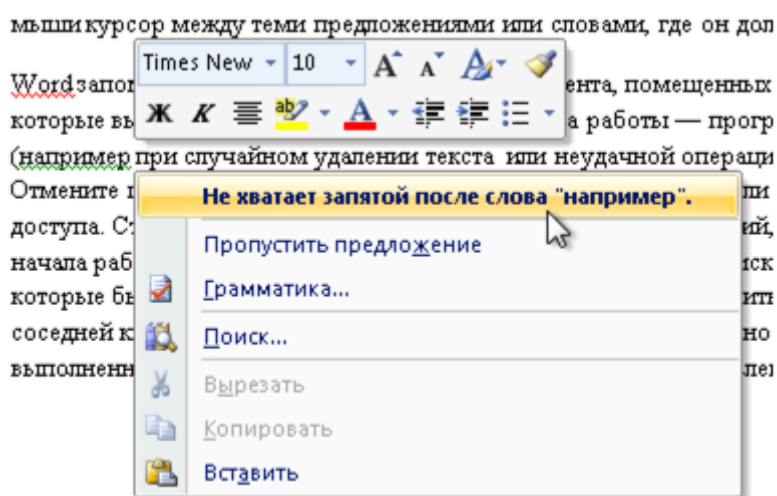


Рис. 2.4. Устранение пунктуационной ошибки

В русском языке существует огромное количество правил пунктуации. Многие из них довольно сложны и имеют исключения, поэтому правильная расстановка знаков препинания часто является непростой задачей даже для образованных людей. Проблемы с расстановкой знаков препинания возникают и у системы правописания Word. В некоторых случаях система не замечает грубейшие пунктуационные ошибки, в иных же ситуациях, наоборот, пытается исправить корректное предложение, поэтому, набирая тексты в Word, будьте предельно внимательны. Не следует полностью доверять трактовке Word неоднозначных по смыслу предложений. Помните о том, что у многих правил русского языка есть исключения, о существовании которых Word может быть неизвестно. Тем не менее в большинстве случаев система способна оказать помощь в грамотном написании текста.

Поиск и замена текста. Иногда в документе требуется найти определенный фрагмент. Сделать это, просматривая страницы, непро-

сто, особенно в случае больших текстов. В Word имеется специальная функция поиска, которая поможет вам быстро найти нужное слово или фразу. На вкладке *Главная* в группе *Редактирование* нажмите кнопку *Найти* или используйте сочетание клавиш *Ctrl+F*. В открывшемся окне (рис. 2.5) введите текст, который вы ищете, и нажмите кнопку *Найти далее*. В результате поиска Word переместится к найденному фрагменту и выделит его в тексте. Для продолжения поиска в оставшейся части документа нажмите кнопку *Найти далее* еще раз. Система выполняет поиск в тексте от текущего положения курсора; если поиск ведется не с начала, то при достижении конца документа вам будет предложено проверить первую часть текста (ту, которая расположена до курсора).

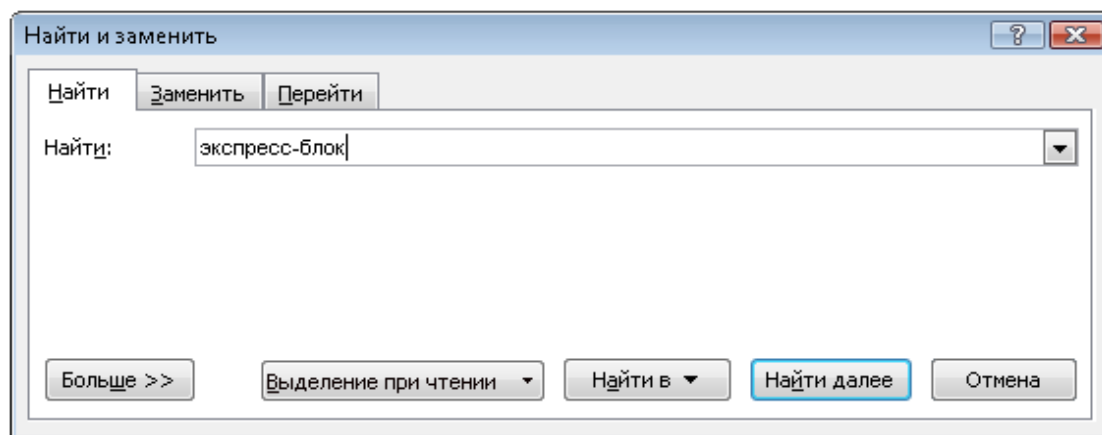


Рис. 2.5. Окно поиска текстового фрагмента

Часто требуется заменить во всем документе какое-либо слово или фразу. Просматривать каждую строчку в поисках ненужного фрагмента, особенно в больших текстах, неудобно. Лучше обратиться к специальной функции замены. В группе *Редактирование* вкладки *Главная* нажмите кнопку *Заменить*. В открывшемся окне наберите искомое слово или фразу в поле *Найти*, а также фрагмент, которым нужно ее заменить, в поле *Заменить на*, затем нажмите кнопку *Найти далее* (рис. 2.6).

Word переместится к первому найденному вхождению, выделит его. Убедитесь, что хотите изменить его на новое слово или фразу, и нажмите кнопку *Заменить*. После этого система выделит следующее вхождение для замены. Если, проверив его, вы решите отказаться от замены, нажмите кнопку *Найти далее*. К такому способу следует прибегать в том случае, если необходимо выявить исключения и проверить каждое вхождение. Если вы уверены в том, что все фрагменты

в тексте должны быть заменены, нажмите кнопку *Заменить все*. При этом Word выполнит автоматическую замену, проинформирует вас о количестве замененных фрагментов.

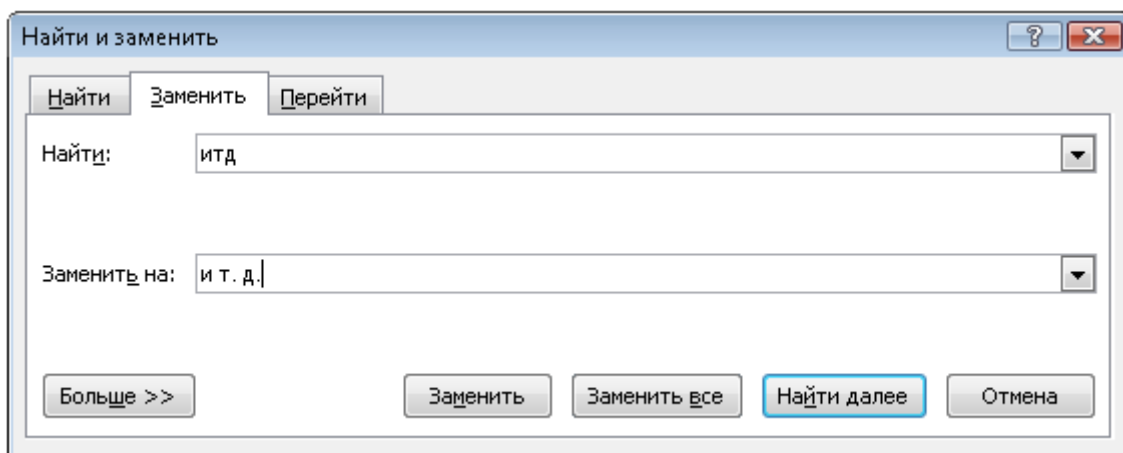


Рис. 2.6. Окно замены текстового фрагмента

Вопросы и задания для самоконтроля

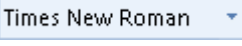
1. Что означают красные и зеленые волнистые линии, которыми подчеркиваются слова или предложения в тексте?
2. Как добавить неизвестное слово в словарь Word?
3. Всегда ли следует доверять системе проверки правописания в Word?
4. Перечислите способы удаления, копирования и вставки текстовых фрагментов.
5. Объясните принцип действия *Буфера обмена* Microsoft Office.
6. Объясните действия *Отменить* и *Вернуть*.
7. Объясните, как работают инструменты *Поиск* и *Замена* текста.

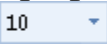
2.4. Форматирование текста




После набора текста его необходимо отформатировать, другими словами, оформить текст документа в соответствии с определенными правилами или по собственному желанию. Под форматированием текста понимается придание символам и абзацам определенных характеристик с помощью специальных инструментов. Прежде чем приступить к их рассмотрению, отметим одну важную особенность,


которая появилась в последней версии Word, – если вы задаете какой-либо параметр форматирования, выбирая его значение из раскрывающегося списка, то при наведении указателя мыши на любую команду в этом списке вы сразу же увидите результат ее действия в тексте. Для закрепления результата следует щелкнуть кнопкой мыши на соответствующем пункте. Перемещая указатель по списку, можно подбирать наилучшее значение, наблюдая за тем, как оно выглядит в тексте.




Форматирование символов. Существуют три важнейших свойства символов, которые вы можете изменять в Word, – шрифт, размер и начертание. Инструменты для работы с этими параметрами находятся в группе *Шрифт* на вкладке *Главная* ленты.

До применения форматирования к текстовому фрагменту его следует выделить. Если текст не выделен, а курсор находится в конце слова, то измененные параметры будут применены только к вновь набираемым символам. Когда курсор стоит внутри слова, изменения характеристик шрифта скажутся только на его буквах. В раскрывающемся списке Шрифт  можно выбрать новый шрифт для выделенного фрагмента. Шрифты отличаются друг от друга формой знаков, толщиной линии, которой вычерчен символ, шириной символов и другими характеристиками, среди которых можно варьировать две – размер и начертание. Обратите внимание на то, что не все шрифты могут быть применены к кириллическим знакам, а только те, которые справа в списке помечены буквами русского алфавита. К латинским буквам можно применить любой шрифт.

Размер шрифта можно изменить в раскрывающемся списке *Размер шрифта*  или ввести произвольное значение на клавиатуре. Именно с помощью этого инструмента, а не регулятора масштаба меняется реальный размер символов, который отразится не только на экране, но и при выводе документа на печать.

Для изменения начертания символов в группе *Шрифт* имеются следующие инструменты:  *Полужирный*;  *Курсив*;  *Подчеркнутый*. Щелкнув на стрелке данной кнопки, вы можете выбрать тип и цвет линии подчеркивания. Нажатие кнопки активирует инструмент, повторное ее нажатие деактивирует. При желании типы начертания можно комбинировать между собой, например сделать шрифт курсивным и подчеркнутым одновременно.

Дополнительное видоизменение символов можно провести при помощи следующих кнопок:  (*Зачеркнутый*) – зачеркивает текст в

выделенном фрагменте;  (*Подстрочный знак*) – уменьшает символы и опускает их ниже основной линии текста (создает нижний индекс);  (*Надстрочный знак*) – уменьшает символы и поднимает их выше основной линии текста (создает верхний индекс – степень);  (*Регистр*) – щелкнув на данной кнопке, вы можете выбрать вариант изменения регистра выделенного текста (перевода строчных букв в прописные или наоборот).

Результат применения всех описанных выше инструментов форматирования символов представлен на рис. 2.7.

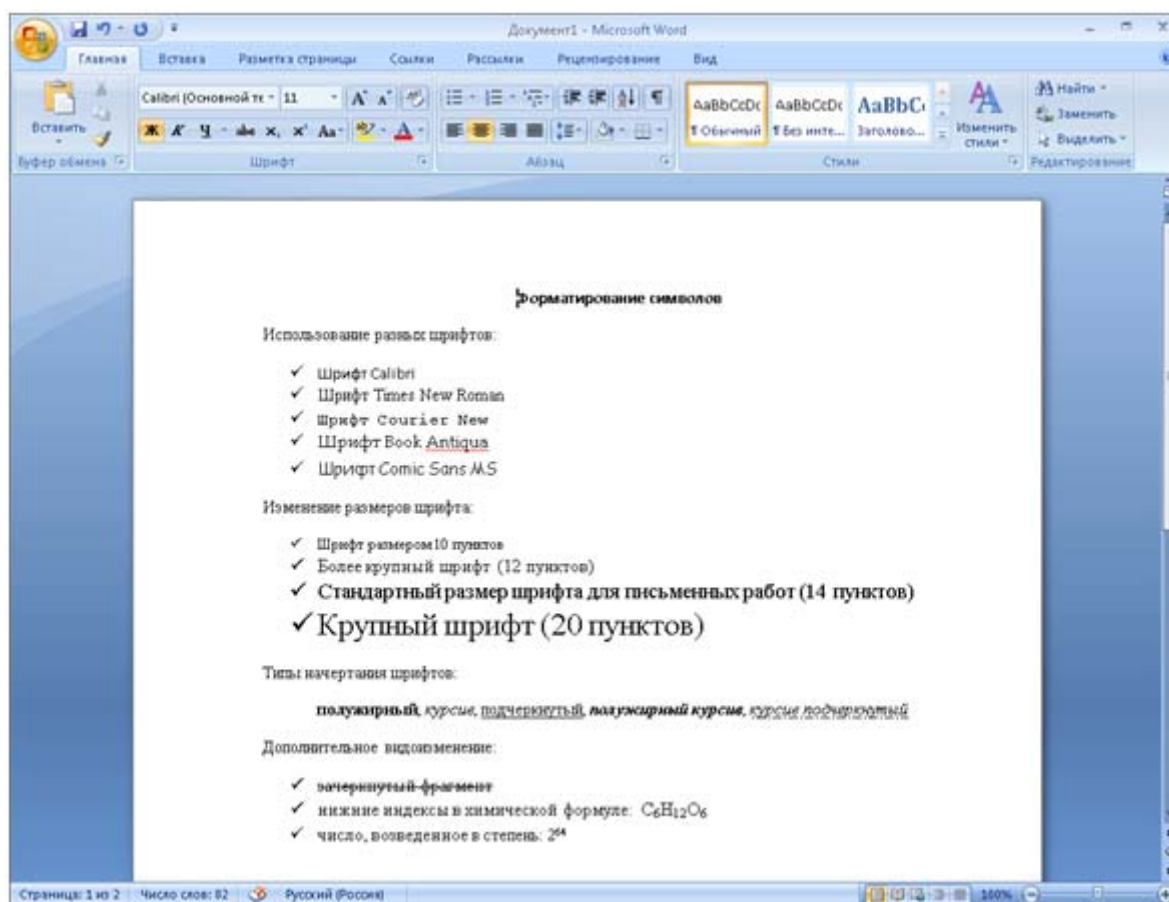




Рис. 2.7. Примеры форматирования фрагментов текста

При желании вы можете изменить цвет шрифта выделенного фрагмента, нажав в группе *Шрифт* кнопку  *Цвет текста*. Щелкнув на стрелке данной кнопки, вы откроете палитру, на которой можно выбрать другой цвет.

Привлечь внимание к определенному фрагменту в тексте можно, выделив его маркером точно так же, как вы бы делали это на бумаге. Нажмите кнопку  *Цвет выделения текста* в группе *Шрифт*,

чтобы активировать инструмент, щелкните слева от первого символа выделяемого фрагмента и, удерживая нажатой левую кнопку мыши, протащите указатель вдоль текста. Когда выделение охватит последний нужный символ, отпустите кнопку. При этом выделенный текст будет закрашен. Щелкните на кнопке *Цвет выделения текста* еще раз, чтобы деактивировать ее. Цвет маркера можно изменить на палитре, открываемой щелчком на стрелке данной кнопки.


Форматирование абзацев. Абзац – это фрагмент текста, находящийся между двумя маркерами конца абзаца, которые можно визуализировать нажатием кнопки *Отобразить все знаки* в группе *Абзац*. Абзацем в Word называется не только фрагмент текста, но и пустая строка, образованная нажатием клавиши *Enter* (то есть содержащая только маркер конца абзаца). Основными характеристиками абзаца являются выравнивание, интервал между строками и величина отступа первой строки. Их можно варьировать с помощью инструментов, расположенных в группе *Абзац* вкладки *Главная*.

Для форматирования одного абзаца его можно не выделять – достаточно просто установить курсор в любую его часть. Для одновременного изменения параметров нескольких абзацев их нужно выделить.

Рассмотрим инструменты, предназначенные для выполнения различных действий абзацами. Например, *выровнить текст*:

- *по левому краю*. Этот способ выравнивания выбран в Word по умолчанию и может быть использован при создании любых документов;
- *по центру*. Данный инструмент применяют для перемещения строки в центр страницы. Именно им, а не пробелом, следует пользоваться для создания заголовков;
- *по правому краю*. Смещает абзац к правому краю страницы. Данный тип форматирования может понадобиться при создании шапки заявления;
- *по ширине*. С помощью данного инструмента можно выровнить текст одновременно по левому и правому краю. Это достигается за счет увеличения промежутков между словами.

Если вы выровнили абзац определенным образом и решили набрать после него новый абзац, нажав клавишу *Enter*, имейте в виду, что он будет иметь такой же тип выравнивания. Переключиться на другой тип можно, нажав соответствующую кнопку в группе *Абзац*. Каждый абзац характеризуется величиной междустрочного интерва-


ла, а также интервала, добавляемого перед абзацем и после него. По умолчанию в Word используется интервал 1,15. Это значит, что расстояние между строками абзаца в 1,15 раза больше высоты строки. При необходимости величину междустрочного интервала можно изменить, щелкнув на кнопке  *Междустрочный интервал* и в раскрывшемся меню выбрав нужное значение. Если ни одно из них вас не устраивает, выполните команду *Другие варианты междустрочных интервалов*, в открывшемся окне (рис. 2.8) в области *Интервал* щелкните на кнопке раскрывающегося списка *Междустрочный* и выберите пункт *Множитель*. Справа в числовом поле *Значение* наберите с клавиатуры нужную величину интервала и нажмите *ОК*.


Между абзацами также предусмотрены определенные интервалы. По умолчанию перед абзацем интервал не добавляется, после него величина интервала составляет 10 пунктов. Интервалы между абзацами можно по желанию добавлять, удалять или изменять их величину. Щелкните на кнопке *Междустрочный интервал*. Из названий двух последних команд ясно, имеется ли интервал перед абзацем и после него: если название команды начинается со слова *Добавить*, это значит, что интервал отсутствует и его можно добавить, выполнив данную команду; когда интервал установлен, соответствующая ему команда начинается со слова *Удалить*, и ее выполнением вы можете удалить интервал.

Задать точное значение интервала между абзацами можно в числовых полях *Перед* и *После* в группе *Интервал* окна настройки параметров абзаца (рис. 2.8). Вызвать его можно нажатием кнопки диалогового окна в правом нижнем углу группы *Абзац* на ленте.

Вы можете самостоятельно изменять отступы абзаца от левого и правого поля и отступ первой строки абзаца. Делают это с помощью специальных маркеров абзаца, которые находятся на линейке. Для визуализации линейки нажмите кнопку, которая находится над вертикальной полосой прокрутки (рис. 2.9).

Менять положение любого маркера можно протаскиванием. Вместе с ним будут меняться отступы абзаца относительно полей документа или отступ первой строки. При задании точной величины отступа ориентируйтесь на шкалу линейки, которая градуирована в сантиметрах. Обратите внимание, что линейка за границами полей документа окрашена другим цветом. Не следует перемещать в эту область маркеры отступов.

Для увеличения отступа абзаца по левому краю на 1,25 см можно использовать кнопку  *Увеличить отступ* в группе *Абзац*. Для

уменьшения отступа на такую же величину обратитесь к соседней кнопке  Уменьшить отступ.

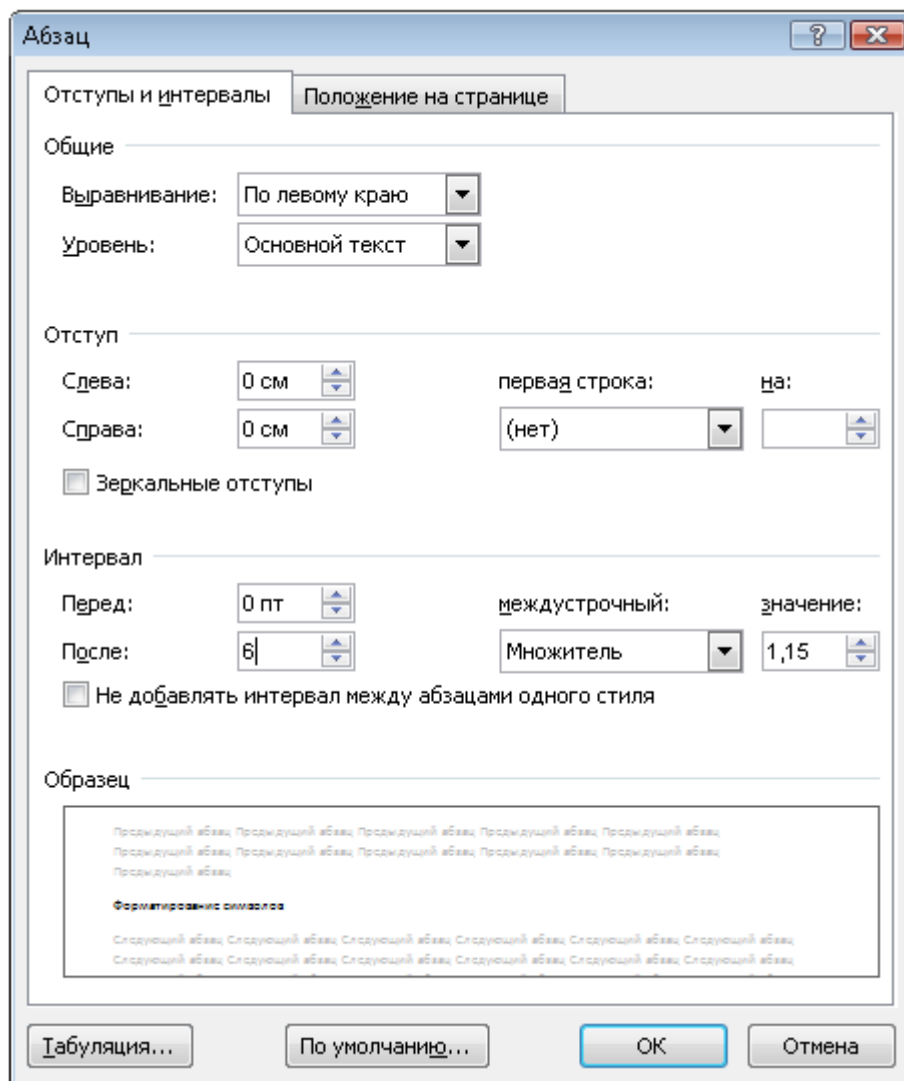





Рис. 2.8. Окно настройки параметров абзаца



Рис. 2.9. Линейка с маркерами отступов

Узнать, какие параметры форматирования шрифта и абзаца установлены в некотором фрагменте, можно, установив на него курсор и посмотрев настройки кнопок форматирования на ленте.

Создание списков. Перечисление однотипных объектов или последовательности действий в документе удобно представлять в виде списка.

В группе *Абзац* имеются три кнопки для создания различных списков:  *Маркеры* – кнопка отвечает за создание маркированного списка;  *Нумерация* – с помощью данной кнопки вы можете создать нумерованный список;  *Многоуровневый список* – кнопка отвечает за создание списка, состоящего из нескольких уровней.

Каждый пункт будущего списка должен находиться в отдельном абзаце. Для создания списка выделите необходимые абзацы и щелкните на одной из рассмотренных кнопок в зависимости от того, какой список хотите получить. Для отмены формата списка выделите его пункты и повторно нажмите эту же кнопку.

Обратите внимание, каждая из кнопок, отвечающих за создание списка, имеет стрелку. Щелкнув на ней, вы развернете варианты представления списка, из которых можно выбрать подходящий (рис. 2.10).

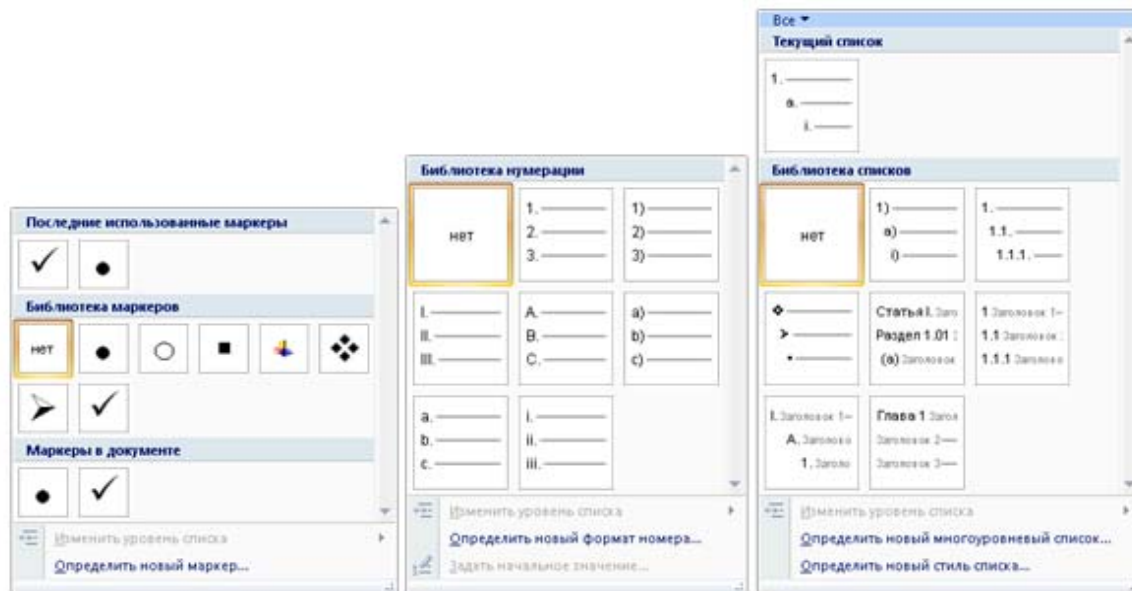


Рис. 2.10. Варианты представления маркированного, нумерованного и многоуровневого списков

Создание нумерованного списка можно начать и вручную. Для этого в начале абзаца наберите единицу с точкой, а затем нужный текст. При переходе к следующему абзацу Word автоматически пре-

вратит его во второй пункт списка, добавив двойку. Так будет происходить при каждом нажатии клавиши *Enter*. Указать системе, что данный абзац не должен входить в состав списка, можно, установив курсор на любом его участке и нажав кнопку *Нумерованный список* в группе *Абзац*.

Применение границ и заливки. В Word любой абзац или произвольный текстовый фрагмент можно окаймить рамкой, залить цветом. Тип рамки, а также отображение определенных границ (для абзаца) можно настроить по желанию.

В группе *Абзац* есть кнопка, отвечающая за настройку границ текста. По умолчанию активен инструмент создания нижней границы. Установите курсор в абзац, к которому будет применена граница, и нажмите эту кнопку. Щелкнув на стрелке данной кнопки, вы раскроете список инструментов, отвечающих за создание различных границ. В процессе работы вам может понадобиться часто создавать определенную границу – выберите щелчком в списке соответствующий ей инструмент. При этом он заменит собой инструмент, присвоенный кнопке настройки границ текста, и вы сможете быстро обратиться к нему, нажав данную кнопку. Для тонкой настройки всех параметров границы щелкните кнопкой мыши на стрелке кнопки создания границ и в появившемся списке выберите пункт *Границы и заливка*. В результате откроется одноименное окно (рис. 2.11), в котором можно выполнять следующие действия:

- варьировать тип рамки, щелкнув на нужном значке слева в списке *Тип* (наиболее эффектно объемная рамка и рамка, отбрасывающая тень);
- задавать тип, цвет и ширину линии рамки в соответствующих списках;
- отключать отображение некоторых границ рамки с помощью кнопок, расположенных области *Образец*;
- нажав кнопку *Горизонтальная линия*, выбрать тип линии, которую нужно вставить после выбранного абзаца. Обратите внимание, окно *Границы и заливка* имеет несколько вкладок. Параметры, настроенные на вкладке *Граница*, будут применены к выделенному абзацу или произвольному текстовому фрагменту. Вкладка *Страница* содержит аналогичные параметры, однако они отвечают за создание рамки, окаймляющей текущую страницу или каждую страницу документа целиком, что можно указать в раскрывающемся списке *Применить к*.

На вкладке *Заливка* в раскрывающихся списках можно выбрать цвет заливки абзаца или другого фрагмента и фоновый узор (или интенсивность цвета заливки).

Для быстрого выбора цвета заливки щелкните на стрелке кнопки *Заливка* в группе *Абзац*, выделите нужный фрагмент, а затем нажмите данную кнопку. На рис. 2.12 представлены три абзаца, помещенные в рамки различных типов и имеющие различные фоновые цвета.

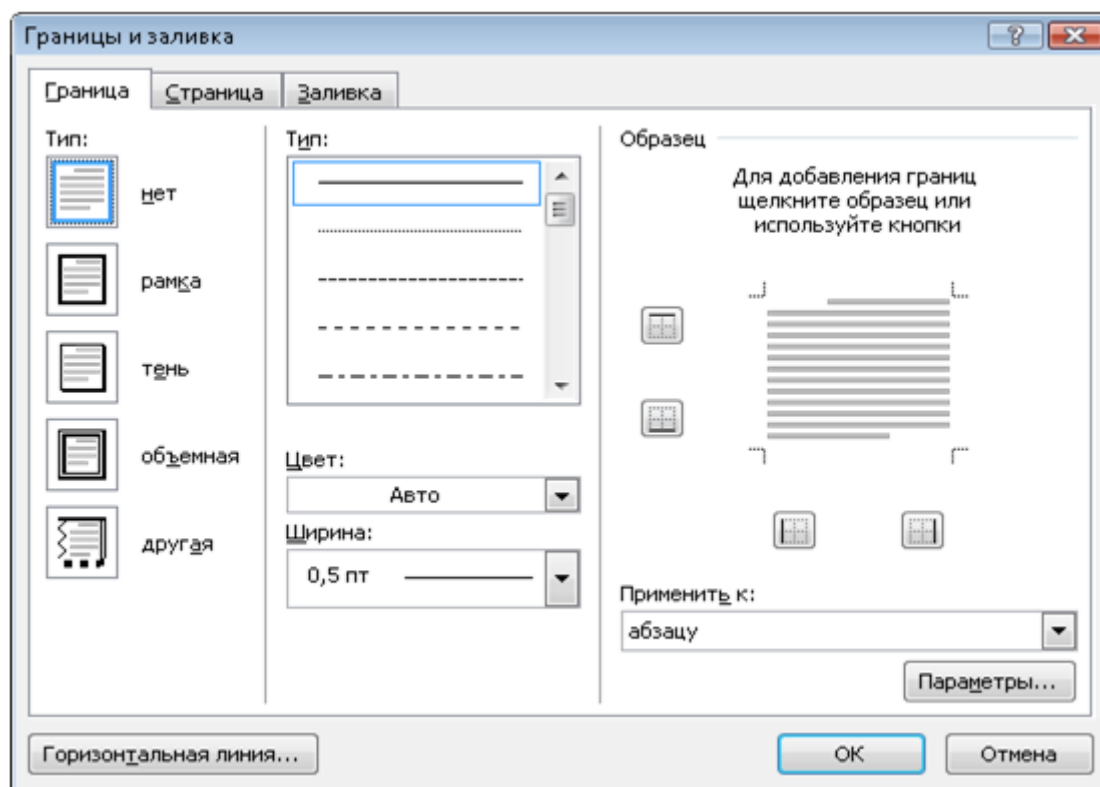



Рис. 2.11. Окно настройки параметров границы и заливки

Этот абзац окаймлен рамкой с тенью. Ширина линии составляет 1,5 пункта. Для заливки выбран желтый цвет.

Этот абзац окаймлен объемной рамкой, для которой выбран определенный тип линии. Ширина линии составляет 3 пункта. Для заливки выбран красный цвет.

В этом абзаце отображается только верхняя и нижняя граница рамки, для создания которых была выбрана двойная волнистая линия. В качестве фона использовался светло-зеленый цвет.

Рис. 2.12. Примеры абзацев, заключенных в рамки разных типов

Для отказа от всех параметров форматирования шрифта и абзаца для выбранного фрагмента нажмите кнопку  *Очистить формат* в

Использование стилей. Предположим, к каждому абзацу в документе вам нужно применить определенные параметры форматирования: выбрать определенный тип выравнивания, отступы абзаца и первой строки, шрифт, его размер, цвет и начертание, наличие границы. Установка всех этих параметров вручную отнимет у вас немало времени. Процесс форматирования текста можно значительно ускорить, применяя к различным фрагментам определенные стили.

Для применения к тексту определенного стиля выделите фрагмент, щелкните на кнопке *Дополнительные параметры* в группе *Стиль*, в открывшейся коллекции наведите указатель на кнопку понравившегося стиля и посмотрите, как он будет выглядеть в тексте (рис. 2.13). Подобрав стиль, щелкните на его кнопке для закрепления результата.

Вы можете заменить выбранную по умолчанию коллекцию стилей, а также цветовую гамму, набор шрифтов, используемые в текущей коллекции, новой. Для этого нажмите большую кнопку *Изменить стили* в группе *Стили*, зайдите в соответствующее подменю и выполните нужную команду.

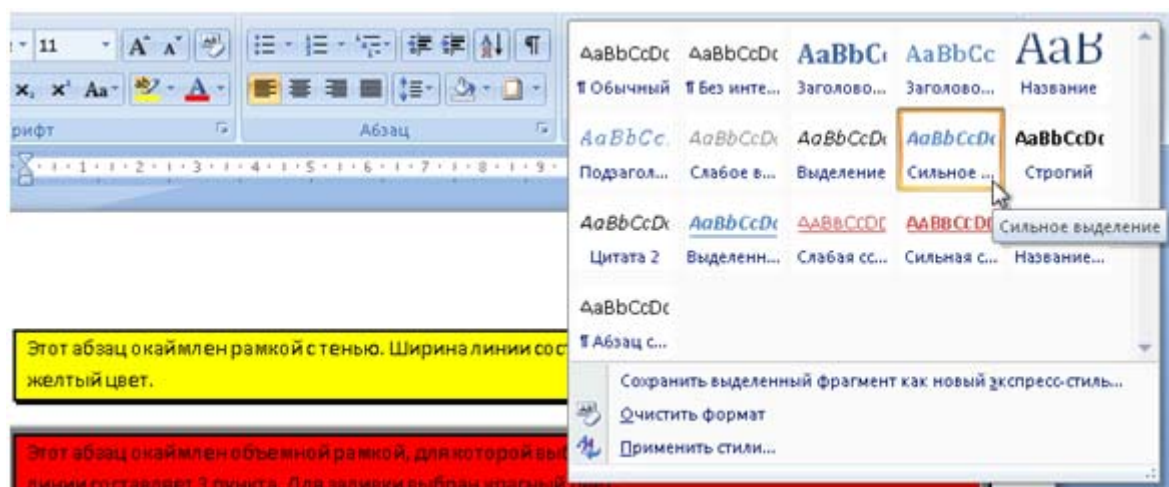


Рис. 2.13. Выбор стиля для выделенного фрагмента

Если в коллекциях Word не нашлось стиля с нужными параметрами, вы можете создать собственный стиль. Для этого выполните такую последовательность действий:

1. Выделите текстовый фрагмент и отформатируйте его с помощью инструментов, расположенных в группах *Шрифт* и *Абзац* вкладки *Главная*.

2. Щелкните на кнопке *Дополнительные параметры* в группе *Стили* и выполните команду *Сохранить выделенный фрагмент как новый экспресс-стиль*.

3. В открывшемся окне задайте имя нового стиля и нажмите *ОК*. В результате кнопка нового стиля будет добавлена в текущую коллекцию стилей.

Имейте в виду, что при закрытии документа созданный стиль не будет сохранен в коллекции; если вы планируете использовать его в дальнейшем при работе с другими документами, следует указать системе, что данный стиль должен присутствовать в коллекции постоянно. Для этого щелкните правой кнопкой мыши на кнопке созданного стиля в коллекции, выполните команду *Изменить контекстное меню* и в открывшемся окне установите находящийся внизу переключатель в положение *в новых документах*, использующих этот шаблон. В этом же окне вы можете внести изменения в параметры форматирования, применяемые в данном стиле. При необходимости изменить стиль, примененный ко многим текстовым фрагментам в документе, не нужно делать это вручную, перебирая каждый абзац по отдельности. Щелкните правой кнопкой мыши на кнопке стиля, который нужно изменить, и выполните команду *Выделить все вхождения*. При этом все фрагменты, созданные с использованием данного стиля, будут выделены в тексте. Останется только щелкнуть на кнопке другого стиля, чтобы применить его к выделенному тексту. Применять к тексту инструменты форматирования можно, не только обращаясь к ленте вкладки *Главная*. Конечно, вы уже заметили, что при выделении текстового фрагмента справа над ним появляется полупрозрачная панель. При наведении на нее указателя мыши можно видеть основные инструменты форматирования, дублирующие некоторые кнопки групп *Шрифт* и *Абзац*. Здесь же имеется и кнопка выбора стилей. Данная панель появляется и при вызове контекстного меню выделенного фрагмента.

Вставка специальных символов. Иногда возникает необходимость добавить в текст символ, которому не соответствует ни одна клавиша на клавиатуре. Это могут быть буквы греческого алфавита, раз-

личные математические символы, знаки валют и др. Для выбора нужного символа и помещения его в документ перейдите на вкладку *Вставка* и в группе *Символы* нажмите кнопку *Символ*. В открывшемся меню вы можете выбрать один из наиболее часто используемых символов.

При отсутствии среди них нужного выполните команду *Другие символы* этого же меню, в открывшемся окне (рис. 2.14) выберите щелчком кнопки мыши символ и нажмите кнопку *Вставить*. Знак будет вставлен в то место в документе, где находился курсор.

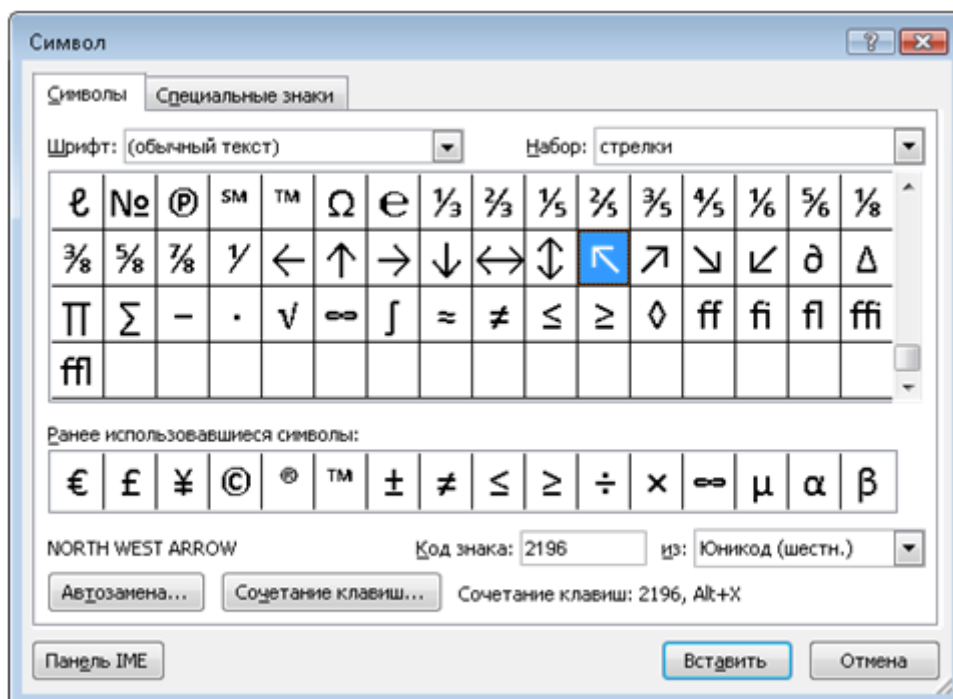


Рис. 2.14. Окно выбора специальных символов

Добавление таблицы. С помощью специальных средств Word вы можете добавлять в документ таблицы со структурой любой сложности и применять к ячейкам таблиц изученные ранее приемы форматирования, а также приемы, доступные только при работе с таблицами.

Вставка рисунков и надписей. При необходимости проиллюстрировать документ вы можете обратиться к специальным инструментам Word, чтобы самостоятельно создать схему или рисунок или вставить готовое изображение из коллекций Word или из внешнего файла.

Кнопки, открывающие вкладки для работы с различными типами графических объектов, находятся в группе *Иллюстрации* на вкладке *Вставка*.

В Word имеется большое количество фигур (линий, простейших геометрических фигур, фигурных стрелок), с помощью которых можно

акцентировать внимание на важных моментах и сделать текст более привлекательным. Открыть коллекцию фигур можно, нажав кнопку *Фигуры* в группе *Иллюстрации* вкладки *Вставка*. Выберите щелчком кнопки мыши понравившийся элемент. Для вставки фигуры в документ щелкните кнопкой мыши на нужном месте страницы и, удерживая ее нажатой, протяните мышь вправо вниз для создания фигуры нужного размера.

Представить в виде графической схемы список, какую-либо последовательность действий, иерархию объектов, циклический процесс или показать связь между элементами можно при помощи коллекции *SmartArt*, открываемой одноименной кнопкой в группе *Иллюстрации* на вкладке *Вставка*. В открывшемся окне (рис. 2.15) выберите в списке тип схемы, а затем щелкните кнопкой мыши на понравившемся эскизе (при этом он отобразится справа в увеличенном виде) и нажмите *ОК* для вставки заготовки схемы в документ.

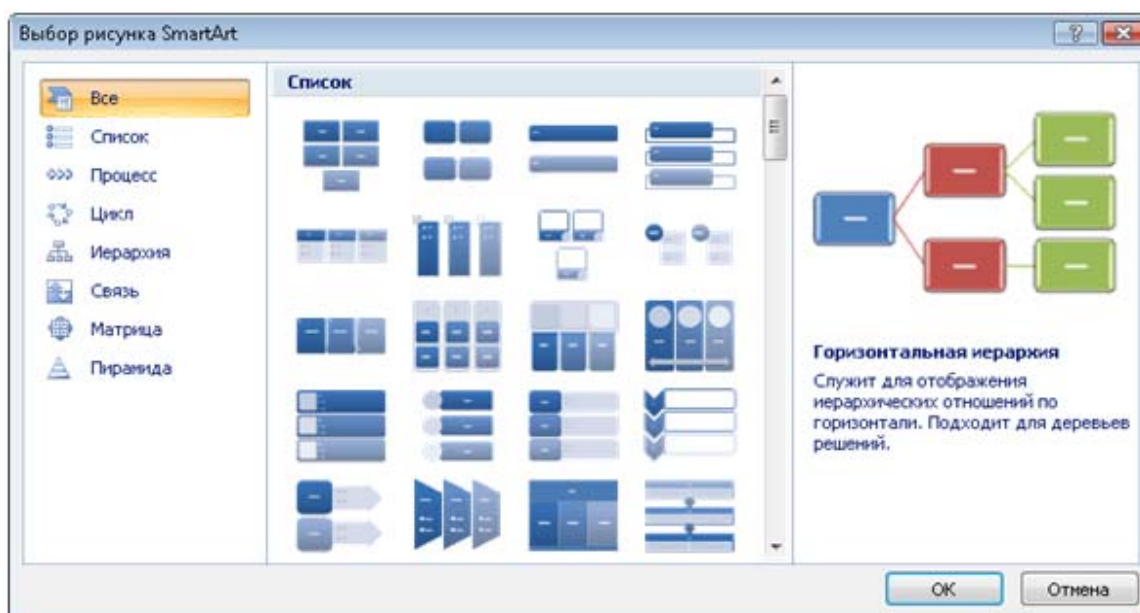


Рис. 2.16. Коллекция заготовок графических схем

Параметры фигур и надписей графической схемы настраиваются с помощью инструментов вкладок *Конструктор* и *Формат* группы вкладок *Работа с рисунками SmartArt*. Кнопка *Диаграмма* в группе *Иллюстрации* на вкладке *Вставка* позволяет вставить в документ диаграмму. Ряды данных для диаграммы нужно задавать в среде редактора электронных таблиц Microsoft Excel, который запускается автоматически при нажатии данной кнопки. Microsoft Excel посвящена гл. 3 данного учебного пособия, поэтому вопрос создания диаграмм будет рассмотрен далее.

Вставка математических формул. С помощью специальных средств Word вы можете поместить в документ математическую формулу любой сложности. Для этого установите курсор в нужное место в тексте и на вкладке *Вставка* в группе *Символы* нажмите кнопку *Формула*. В результате в документе появится область ввода, а также будет открыта новая вкладка ленты *Конструктор* (группа *Работа с формулами*). Главный принцип использования инструментов данной вкладки состоит том, что сначала следует вводить математический оператор, а потом добавлять в его поля числа.

Вкладка *Конструктор*, содержащая инструменты создания формул, а также процесс набора формулы выглядит как представлено на рис. 2.16.

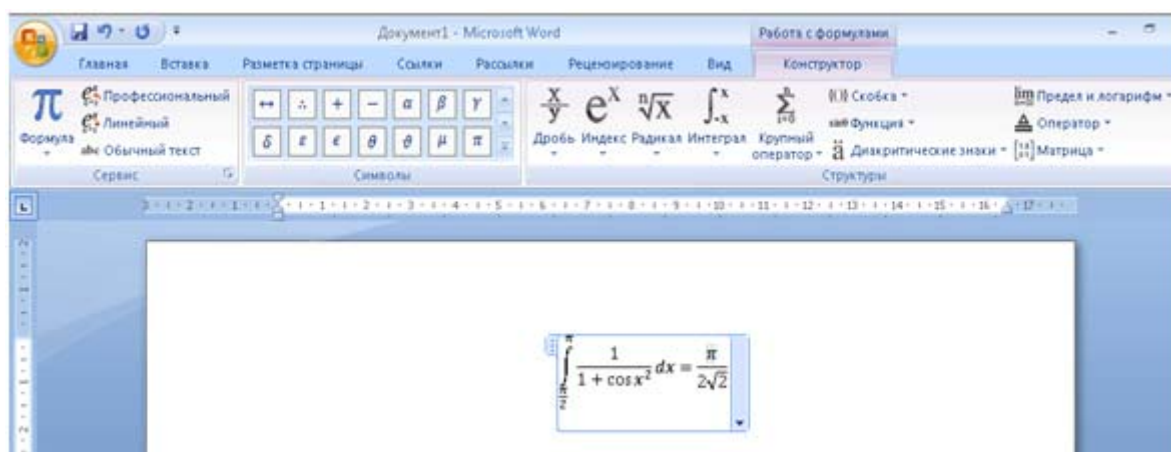


Рис. 2.16. Инструменты работы с математическими формулами и процесс создания формулы

Созданную формулу можно сохранить в библиотеке Word, чтобы впоследствии иметь возможность вставлять ее в другие документы. Щелкните на стрелке справа в области ввода, выполните команду *Сохранить как новую формулу*. В открывшемся окне наберите имя выражения и нажмите кнопку *ОК*. В дальнейшем, чтобы добавить сохраненное выражение в документ, щелкните на стрелке кнопки *Формула* в группе *Символы* вкладки *Вставка* в списке эскизов щелчком кнопки мыши выберите нужный.

Добавление сноски. Иногда возникает необходимость пояснить значение термина в тексте или уточнить некоторые детали относительно него. Это принято оформлять в виде сноски – текста, набранного мелким шрифтом под чертой в конце страницы, на который ведет ссылка (число рядом с поясняемым словом). Вставить пояснение можно следующим способом: установите курсор в конце поясняемого

слова, перейдите на вкладку *Ссылки* и в группе *Сноски* нажмите кнопку *Вставить сноску*.

В результате справа от слова надстрочным шрифтом появится номер сноски, а курсор будет перемещен в конец страницы, где нужно набрать пояснительную запись (рис. 2.17). По окончании редактирования сноски щелкните дважды на свободном участке рабочей области, чтобы вернуться к основному тексту.

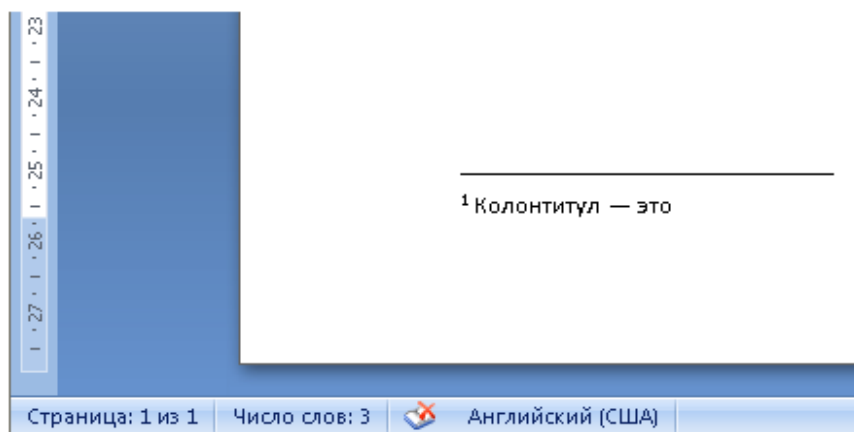


Рис. 2.17. Вставка сноски в документ

Настройка параметров страницы и печать документа. Перед выводом документа на печать необходимо установить нужные параметры страницы, добавить колонтитулы и убедиться, перейдя в режим предварительного просмотра, что документ не содержит ошибок в оформлении и готов к распечатке.

По окончании оформления документа переключитесь в режим предварительного просмотра, чтобы увидеть, как будет выглядеть документ при распечатке. В меню *Кнопки Office* выполните команду *Печать Предварительный просмотр*. В результате документ будет открыт в окне предварительного просмотра в том виде, в котором он будет выведен на печать. Указатель мыши примет вид лупы. Щелкнув на странице, вы сможете увеличить ее для более детального просмотра. Повторный щелчок приводит к уменьшению масштаба. Замеченные ошибки в настройке параметров страницы исправьте с помощью уже знакомых вам инструментов, расположенных в группе *Параметры страницы* на единственной вкладке *Предварительный просмотр*. Для запуска процесса печати нажмите кнопку *Печать* (первую на ленте), в открывшемся окне измените при необходимости параметры печати (например, количество копий документа или выборку страниц) и нажмите кнопку *ОК*.

ГЛАВА 3. ПРОГРАММА MICROSOFT EXCEL 2007

3.1. Знакомство с программой Microsoft Excel 2007

Для запуска Excel выполните команду *Пуск -> Все программы -> Microsoft Office -> Microsoft Office*.

Поскольку Excel является программой, входящей, как и Word, в состав пакета Microsoft Office, интерфейс этих программ во многом схож.

Главное меню также представлено в виде вкладок, на ленте которых находятся группы инструментов, предназначенных для форматирования ячеек и обработки данных. Некоторые из них вам знакомы из опыта работы в программе Word, большинство же являются совершенно новыми.

Рассмотрим структуру окна Excel (рис. 3.1). Электронная таблица состоит из ячеек, которые образуют строки и столбцы. Файл электронной таблицы называется *книгой* (см. заголовок окна). По умолчанию новый файл Excel (книга) имеет три электронные таблицы – три *листа* (так принято называть рабочие области в Excel). Переключаться между листами можно с помощью ярлыков в нижней части окна.

В каждую ячейку можно ввести данные трех типов: текст, число, которое может быть представлено в разных форматах, и формулу. По формуле выполняется расчет, результат которого отображается в содержащей ее ячейке.

Строка формул – это уникальный элемент интерфейса Excel, расположенный под лентой. Слева в строке выводится адрес активной ячейки (той, которая выделена черной рамкой), справа – ее содержимое, которое можно редактировать. Кроме того, строка формул содержит кнопку вызова *Мастера функций*, которые используются для создания математических выражений.

Прежде чем переходить к вопросу проведения расчетов в Excel, рассмотрим правила ввода редактирования данных.

Ввод и редактирование данных в ячейках. Данные всегда вводятся в активную ячейку, на которой находится черная рамка. При первом запуске программы Excel по умолчанию активна ячейка A1 (рис. 3.1). Для активизации другой ячейки необходимо поместить на нее рамку выделения. Это можно сделать, щелкнув на ней кнопкой

мыши или переместив рамку к нужной ячейке с помощью клавиш управления курсором. Выбрав ячейку, наберите в ней текст, число или формулу (о вводе сложных формул с использованием встроенных функций Excel будет рассказано в отдельном параграфе).

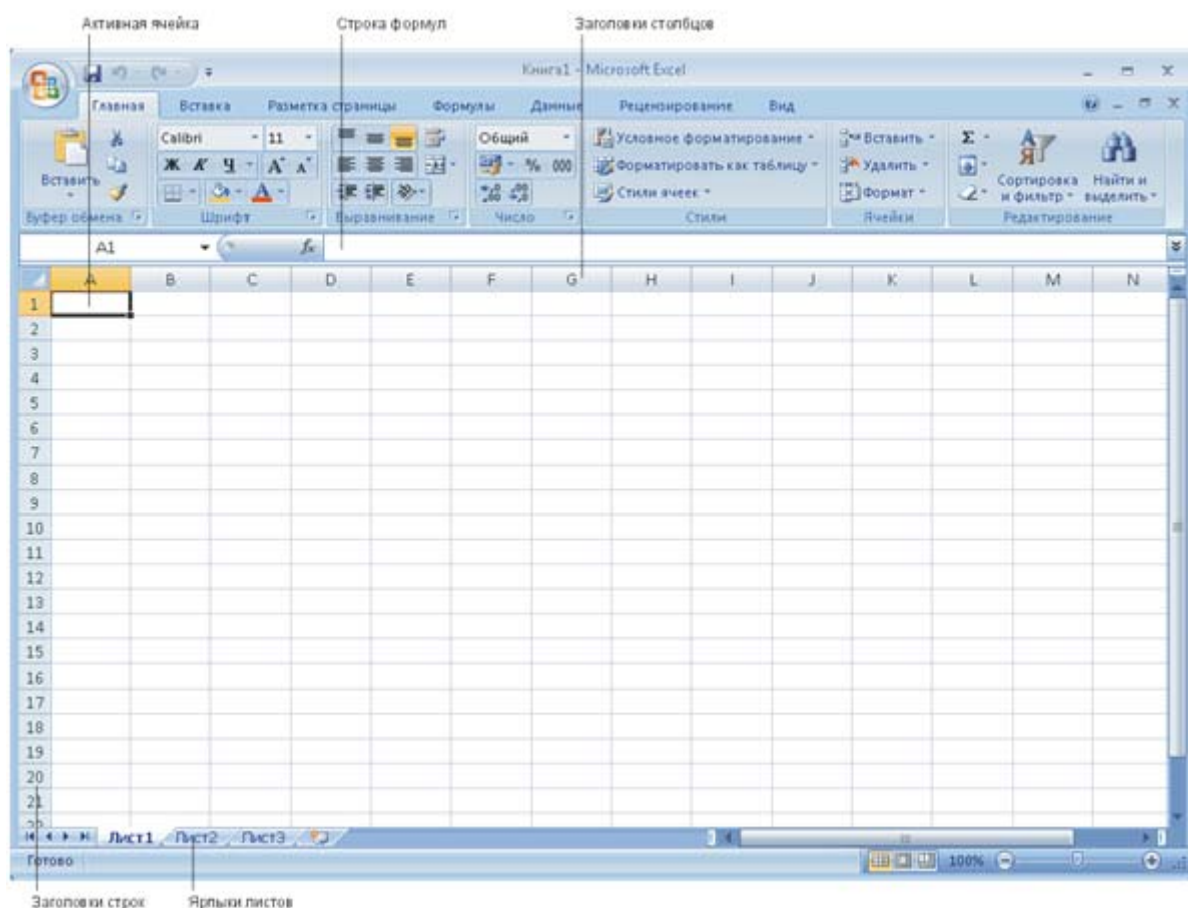


Рис. 3.1. Окно Microsoft Excel

Простейший пример таблицы, созданный в Excel, представлен на рис. 3.2.

При вводе десятичных дробей используйте запятую. Числа, содержащие точку, дефис или слеш, Excel воспринимает как даты. Так, если вы наберете в ячейке 1.5, 1/5 или 1–5, система распознает эту запись как первое мая текущего года, трансформировав ее в 01.май. Полную дату (в формате «число.месяц.год» – 01.05.2011) можно увидеть в строке формул, выделив ячейку. Для ввода даты, содержащей другой год, наберите последовательно через точку, дефис или слеш число, месяц и год. Двоеточие используется для ввода времени. Так, если вы наберете в ячейке 3:15, Excel распознает эту запись как время 3:15:00.

	А	В	С	Д
1	Список покупок	Цена	Количество	Стоимость
2	Шампунь	62,5	6	375
3	Зубная паста	37	3	111
4	Мыло	25	8	200
5	Гель для душа	50	3	150
6	Дезодорант	87,5	4	350
7	Стиральный порошок	38	4	152
8	Пена для ванны	63	2	126
9	Освежитель воздуха	34	6	204
10	Итого			1668

Рис. 3.2. Простейший пример таблицы, созданной в Excel

Для завершения ввода и перемещения к следующей нижней ячейке нажмите клавишу *Enter* или же используйте мышь или клавиши управления курсором для перехода к другим ячейкам.

В случае если ширина вводимого текста превышает ширину ячейки, он будет накладываться на пустые ячейки справа, но не заполнять их. Если же в ячейках, находящихся справа, имеются данные, то набираемый текст не будет пересекаться с ними. При удалении с ячейки рамки выделения текст будет «обрезан» по ширине, но увидеть его полностью можно в строке формул, «щелкнув» на ячейке. Однако существует простой способ избавиться от наложения – изменить ширину столбца с этой ячейкой. Для этого наведите указатель мыши на правую границу заголовка столбца и, когда он примет вид разнонаправленных стрелок, нажмите левую кнопку мыши и протаскивайте границу вправо до тех пор, пока не отобразится весь текст. Именно таким образом была увеличена ширина первого столбца на рис. 3.2. Для задания точной ширины столбца следите за значением во всплывающей подсказке при перетаскивании границы.

Визуализировать текст, не уместяющийся по ширине ячейки, можно и другим способом – переносом по словам за счет увеличения высоты строки. Выделите щелчком кнопки мыши проблемную ячейку и на вкладке *Главная* в группе *Выравнивание* нажмите кнопку *Перенос текста*. При этом высота строки, в которой находится ячейка, будет увеличена так, чтобы ее скрытое содержимое полностью отобразилось. Для переноса текста по словам высоту ячейки можно менять и вручную: протаскиванием за границу заголовка, как в случае столбца.

Для ввода простых формул, содержащих только арифметические знаки (+, -, *, /), выполните следующие действия:

- выделите щелчком ячейку, в которую нужно поместить формулу;
- введите знак равенства = (это нужно делать всегда при наборе формул);
- введите адреса ячеек, значения которых будут принимать участие в расчете. Для этого щелкните на первой из них. При этом ячейка будет выделена бегущей рамкой, ее адрес появится в ячейке ввода (рис. 3.3);
- наберите с клавиатуры арифметический знак и щелчком выделите вторую ячейку для вставки ее адреса (рис. 3.3) или же наберите адрес с клавиатуры, переключившись на английскую раскладку;
- нажмите на клавиатуре *Enter* для завершения ввода.

В итоге в ячейке отобразится результат вычисления.

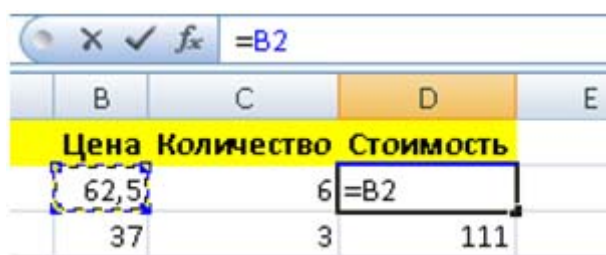


Рис. 3.3. Ввод простейшей формулы

Вы можете комбинировать в одной формуле несколько арифметических операций. При необходимости используйте скобки, как и в случае стандартной записи математических выражений. Например, если нужно сложить значения двух ячеек, затем результат поделить на число, находящееся в третьей ячейке, в виде формулы это будет выглядеть следующим образом: $= (B2 + C2) / D2$. При вводе формулы адреса ячеек указывайте щелчками или набирайте вручную.

Для удаления содержимого ячейки выделите ее щелчком и нажмите клавишу *Delete*; если нужно набрать в заполненной ячейке новые данные, предыдущие удалять необязательно – просто выделите ее и начните ввод. Старые данные будут автоматически заменены.

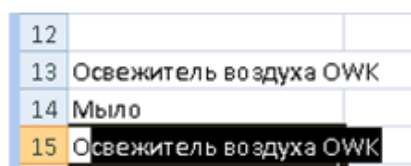
Ячейка может содержать большой текстовый фрагмент или сложную формулу, полностью удалять которые для внесения изменений нерационально. Следует дважды щелкнуть кнопкой мыши на

ячейке, установить курсор в нужное место для редактирования, внести необходимые изменения и нажать клавишу *Enter*.

Вы можете отказаться от редактирования ячейки, даже если уже начали выполнять его. Для этого просто нажмите клавишу *Esc*. При этом в ячейке будут восстановлены исходные данные. Для отмены уже совершенного действия нажмите стандартное сочетание клавиш *Ctrl+Z* или кнопку *Отменить* на *Панели быстрого доступа*.

При изменении значений в ячейках, на которые ссылается формула, результат вычислений в ячейке, содержащей формулу, будет автоматически пересчитан.

Автозавершение. Часто при заполнении таблицы приходится набирать один и тот же текст. Имеющаяся в Excel функция автозавершения помогает значительно ускорить этот процесс: если система определит, что набираемая часть текста совпадает с тем, который был введен ранее в другой ячейке, она подставит недостающую часть и выделит ее черным цветом (рис. 3.4).



12		
13	Освежитель воздуха OWK	
14	Мыло	
15	Освежитель воздуха OWK	

Рис. 3.4. Автозавершение при вводе текста

Можно согласиться с предложенным вариантом и перейти к заполнению следующей ячейки, нажав клавишу *Enter*, или же продолжить набирать нужный текст, не обращая внимания на выделение при совпадении первых нескольких букв.

Автозаполнение. Механизм автозаполнения удобно применять, когда в ячейки требуется ввести какую-либо последовательность данных.


Автозаполнение можно применять при вводе времени, дат, дней недели, месяцев, а также комбинаций текста с числом. Для этого достаточно ввести только первое значение. Принцип заполнения остальных ячеек Excel определит самостоятельно, увеличивая текущее значение на единицу (рис. 3.5). При необходимости ввести эти же величины с определенным интервалом поступите описанным выше образом, введя два первых значения, чтобы Excel определил разницу между ними.

При работе с ячейками важно разделять понятия: «содержимое ячейки» и «формат ячейки». Содержимое – это введенные данные. К формату относится не только форматирование, примененное к ячейке (выравнивание содержимого, параметры шрифта данных, заливка,

граница), но и формат данных в случае, когда ячейки содержат числа. О числовых форматах приемах форматирования ячеек будет рассказано ниже, а сейчас рассмотрим вопрос копирования форматов с помощью автозаполнения.

12	<i>Время</i>	<i>Дата</i>	<i>День недели</i>	<i>Месяц</i>	<i>Название работы</i>
13	19:15	03.02.2007	Понедельник	Январь	Лабораторная работа №1
14	20:15	04.02.2007	Вторник	Февраль	Лабораторная работа №2
15	21:15	05.02.2007	Среда	Март	Лабораторная работа №3
16	22:15	06.02.2007	Четверг	Апрель	Лабораторная работа №4
17	23:15	07.02.2007	Пятница	Май	Лабораторная работа №5
18	0:15	08.02.2007	Суббота	Июнь	Лабораторная работа №6
19	1:15	09.02.2007	Воскресенье	Июль	Лабораторная работа №7
20	2:15	10.02.2007	Понедельник	Август	Лабораторная работа №8
21	3:15	11.02.2007	Вторник	Сентябрь	Лабораторная работа №9
22					
23					

Рис. 3.5. Автозаполнение ячеек с различными типами данных

Предположим, вы отформатировали ячейку, задали определенный формат числа и хотите распространить формат данной ячейки на ряд других без вставки содержимого. Для этого выделите ячейку и выполните операцию автозаполнения протаскиванием рамки за нижний маркер. После того как вы отпустите кнопку мыши, в правом нижнем углу ряда появится кнопка  *Параметры автозаполнения*. Щелкнув на ней, вы откроете меню, в котором можно выбрать способ заполнения ячеек (рис. 3.6).

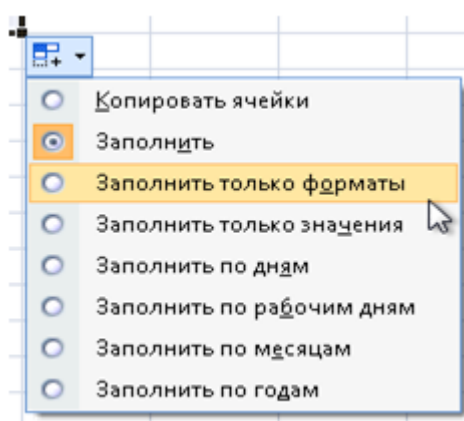


Рис. 3.6. Выбор способа автозаполнения

В данном случае для копирования формата следует выбрать пункт *Заполнить только форматы*.

При необходимости применить автозаполнение только к содержимому ячеек без сохранения формата выполните команду *Заполнить только значения*. По умолчанию копируются содержимое ячеек (с созданием последовательности, если это возможно) и их формат.

Форматирование ячеек. К таблицам, созданным в Excel, можно применять все те же приемы форматирования ячеек, что и в Word, а также некоторые специфические для Excel приемы.

Принципы форматирования содержимого ячеек Excel ничем не отличаются от рассмотренных ранее (для текстовых таблиц в Word). Кнопки задания определенных параметров шрифтов, заливки ячеек и визуализации границ (которые по умолчанию скрыты) находятся в группе *Шрифт* на вкладке *Главная*.

Стиль таблицы как единого целого определяет оформление заголовка, границы и заливку ячеек. После ввода всех данных в таблицу можете приступить к ее форматированию.

Для чего выделите весь диапазон таблицы и нажмите кнопку *Форматировать как таблицу* в группе *Стили* на вкладке *Главная*. Выбрав в коллекции подходящий стиль, щелкните на его эскизе. В открывшемся окне необходимо установить флажок *Таблица с заголовками*, если вы уже ввели текст заголовков. В противном случае поверх выделенного диапазона будет вставлена строка заголовка с именами столбцов по умолчанию *Столбец 1*, *Столбец 2* и т. д., которые нужно будет переименовать.

	А	В	С	Д
1	Список покупок ▼	Цена ▼	Количество ▼	Стоимость ▼
2	Шампунь	62,5	6	375
3	Зубная паста	37	3	111
4	Мыло	25	8	200
5	Гель для душа	50	3	150
6	Дезодорант	87,5	4	350
7	Стиральный порошок	38	4	152
8	Пена для ванны	63	2	126
9	Освежитель воздуха	34	6	204
10	Итого			1668

Рис. 3.7. Использование встроенного стиля таблицы

Вы можете сначала задать стиль таблицы, а затем начать заполнять ее. Для этого охватите рамкой примерную область расположения таблицы, обратитесь к кнопке *Форматировать как таблицу*, выберите подходящий стиль и в открывшемся окне просто нажмите *ОК*. В

нижнем правом углу вставленной заготовки можно видеть маленький треугольник. Для изменения размеров заготовки таблицы наведите на него указатель мыши и протащите границу по горизонтали или вертикали, чтобы увеличить или уменьшить количество оформленных ячеек. При необходимости увеличить и число строк, и число столбцов протяните сначала нижнюю границу вниз, а затем правую вправо. Определившись с размерами, заполните форму. При желании вы можете комбинировать стиль таблицы со стилями отдельных ячеек в ее составе. На рис. 3.7 приведен пример таблицы, оформленной с использованием стиля из коллекции Excel.

Формат данных в ячейке. Выше уже упоминалось о том, что числа в Excel могут отображаться в различных форматах.

Расскажем о том, какие существуют форматы чисел и как задать для ячейки определенный числовой формат.

По умолчанию в Excel установлен общий формат ячеек. Это означает, что при вводе данных система распознает их и выравнивает в ячейке определенным образом, а при особых условиях ввода автоматически меняет числовой формат. Некоторые варианты автоматического форматирования ячеек уже были описаны выше. Вспомним эти случаи, а также рассмотрим примеры других ситуаций автоматической смены числового формата:

- по умолчанию десятичную часть дробных чисел следует набирать через запятую; при вводе чисел через точку, слеш или тире в ячейке устанавливается формат даты, данные отображаются в виде даты;
- при использовании двоеточия при наборе чисел для них автоматически устанавливается формат времени;
- для ввода числа в процентном формате добавьте после него знак %;
- представить число в денежном формате (в рублях) можно, введя после него *р.* В денежном формате к числу добавляется знак денежной единицы, а каждые три цифры числа отделяются друг от друга пробелами для лучшего восприятия;
- отделив пробелами регистры числа, например 36 258, 2 739, вы переведете его в числовой формат. Он аналогичен денежному, за исключением того, что на экран не выводится знак денежной единицы.

Excel оперирует с числами с точностью до 15 знака после запятой, однако в ячейках по умолчанию выводится только два десятичных зна-

ка (при необходимости эту настройку можно изменить). Полностью число можно увидеть в строке формул, выделив ячейку. Изменить формат числа можно при помощи не только описанных выше приемов набора чисел, но и специальных инструментов. За числовые форматы ячеек отвечают параметры, находящиеся в группе *Число* на вкладке *Главная*. В раскрывающемся списке *Числовой формат* вы можете выбрать формат чисел для выделенной ячейки или диапазона. С большинством форматов вы уже ознакомились. Необходимо лишь сделать замечание относительно процентного формата. При его выборе число, находящееся в ячейке, будет умножено на 100 и к нему добавится знак %.

Новыми для вас являются экспоненциальный и текстовый форматы. Рассмотрим на конкретных примерах экспоненциальный формат числа. Любое число можно представить в виде десятичной дроби, умноженной на 10 в степени, равной количеству знаков после запятой. Так, число 1 230 можно записать в виде 1,23 10³, число 0,000 15 как 1,5 10⁻⁴. Другими словами, в числе выделяется мантисса (дробная часть), порядок записывается в виде показателя степени. Точно так же поступает и Excel, используя следующие правила оформления. После мантиссы ставится разделитель *E*, а затем записывается показатель степени с обязательным указанием знака (+ для положительного показателя, – для отрицательного). Так, число 1 230 в экспоненциальном формате будет выглядеть как 1,23E+03, а число 0,000 15 как 1,5E-04. Если в мантиссе содержится более двух знаков после запятой, они будут скрыты (Excel проводит автоматическое округление для отображения, но не реальное округление числа).

Текстовый формат полезен тогда, когда требуется, чтобы введенное число распознавалось системой как текст и не принимало участие в расчетах. При выборе текстового формата число в ячейке будет выровнено по левому краю, как и текст.

С помощью кнопок *Увеличить разрядность* и *Уменьшить разрядность* увеличивают или уменьшают количество знаков, отображаемых после запятой.

3.2. Проведение расчетов в программе Microsoft Excel 2007

Основное назначение Excel – выполнение расчетов с данными. Обработка данных происходит в ячейках, содержащих формулы. Правила ввода простейших формул вы уже изучили в начале главы. В

данном параграфе будут рассмотрены общие принципы создания формул любой сложности и приведены примеры типичных расчетов в Excel.

Правила ввода формул. Ввод любой формулы всегда нужно начинать со знака равенства =. В формуле могут находиться:

- знаки арифметических действий: +, -, *, /, ^ (знак возведения числа в степень), знак %;
- числа, строки (они берутся в кавычки);
- ссылки на ячейки и диапазоны ячеек (как на текущем листе, так и на других листах книги) для определения порядка вычислений, скобки;
- встроенные функции.

В Excel имеется большое количество функций, с помощью которых можно проводить вычисления и другие действия, относящиеся к разным областям знаний. При использовании встроенной функции после знака = следует ввести ее имя, а затем в скобках аргументы функции – данные, которые используются в расчетах. Аргументами функции могут быть числа, ссылки на ячейки или диапазоны ячеек, а также другие встроенные функции (они называются вложенными). Рассмотрим конкретные примеры:

- =A2+B2 – сложение значений двух ячеек;
- =A1*0,8 – умножение числа из ячейки A1 на 0,8;
- =D1^2+1 – возведение числа из ячейки D1 в квадрат и прибавление единицы к результату;
- =СУММ(A1: A5) – суммирование значений из диапазона ячеек A1: A5. Это пример использования встроенной функции. Здесь СУММ – имя функции, A1:A5 – диапазон ячеек, ее единственный аргумент, заключенный в скобки;
- =МУМНОЖ(B1:B2;B7:C7) – вычисление произведения матриц B1:B2 и B7:C7. Эта функция имеет два аргумента, которые являются массивами данных из выделенных диапазонов.

При наличии у функции нескольких аргументов они отделяются друг от друга точкой с запятой. В качестве аргументов функций вы можете использовать ссылки на ячейки и диапазоны на текущем и других листах. В последнем случае перед адресом ячейки или диапазона следует ввести название листа в кавычках и поставить разделитель "!". Например, «Лист1»! B2, «Лист 3»! A1: C4.

Параметры вставки в документ встроенных функций находятся на вкладке *Формулы* в группе *Библиотека функций*. Функции распре-

делены по категориям в зависимости от типов задач, для решения которых они нужны. О назначении той или иной функции можно прочесть во всплывающей подсказке, которая появляется при наведении указателя мыши на имя функции в меню (рис. 3.8).

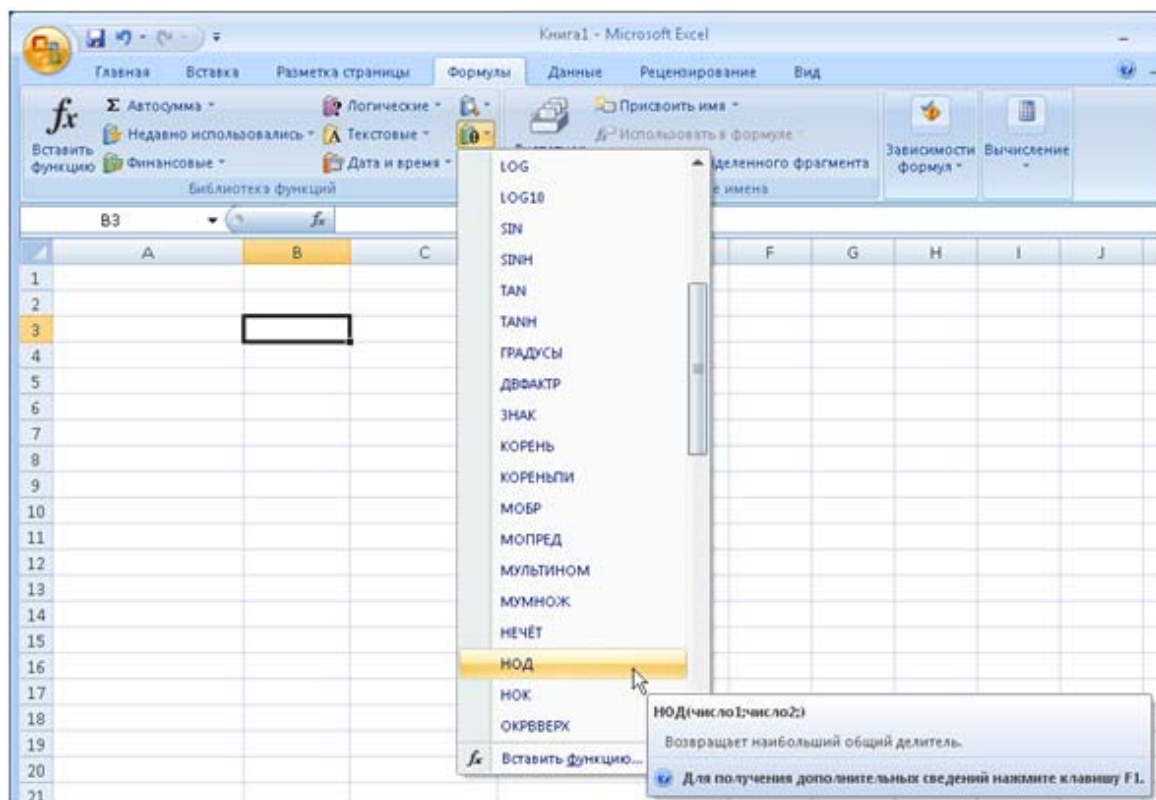




Рис. 3.8. Просмотр назначения функции

Если вы хотите просмотреть полный список встроенных функций Excel, нажмите кнопку *Вставить функцию* , которая находится в строке формул. В открывшемся окне *Мастера функций* выберите в раскрывающемся списке *Категория* пункт *Полный алфавитный перечень* и в списке ниже щелчком выделите имя функции, чтобы прочитать о выполняемых ею действиях.

Название встроенной функции можно ввести с клавиатуры (что крайне нежелательно ввиду высокой вероятности ошибки), вставить из соответствующего меню кнопок, расположенных в группе *Библиотека функций* на вкладке *Формулы*, или же из окна *Мастера функций*. Часто применяемые на практике функции вынесены в меню кнопки, которая находится в группе *Редактирование* на вкладке *Главная*. Рассмотрим задачи, связанные с их использованием.

Простейшие расчеты. Функция суммирования данных  является самой востребованной, именно поэтому задействовать ее в

Excel проще всего. Щелкнув на стрелке кнопки, вы раскроете список команд, вызывающих функции, которые можно задействовать так же быстро, как и функцию суммирования. Схема действий при их использовании не отличается от последовательности шагов для функции суммирования. Ниже приводится краткое описание функций, вызываемых командами кнопки.

Среднее – вызывает функцию =СРЗНАЧ(), с помощью которой можно подсчитать арифметическое среднее диапазона ячеек (просуммировать все данные, а затем разделить на их количество).

Число – вызывает функцию =СЧЕТ(), которая определяет количество ячеек в выделенном диапазоне.

Максимум – вызывает функцию =МАКС(), с помощью которой можно определить самое большое число в выделенном диапазоне.

Минимум – вызывает функцию =МИН() для поиска самого маленького значения в выделенном диапазоне.

Результат работы некоторых из перечисленных функций можно видеть, не обращая непосредственно к ним самим. Выделите интересующий вас диапазон и посмотрите вниз на строку состояния Excel. Слева от регулятора масштаба появятся значения суммы, количества ячеек в диапазоне и среднего арифметического (рис. 3.9).

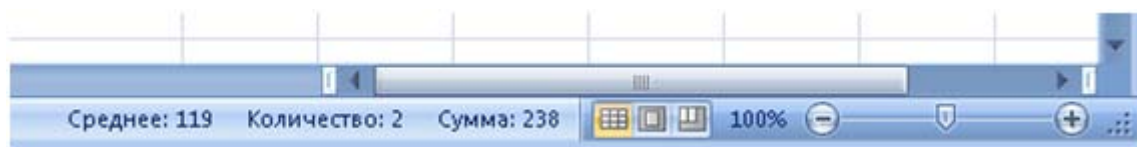


Рис. 3.9. Результаты вычислений в строке состояния для выделенного диапазона

Относительная и абсолютная адресация. Адреса ячеек и диапазонов в Excel могут быть относительными и абсолютными. До сих пор рассказывалось об относительных ссылках на ячейки и диапазоны, которые состоят только из номера строки и буквы столбца, например B2 или D4:D8. Преимущество относительной адресации состоит в том, что при копировании ячеек с использованием автозаполнения, ссылки в скопированных формулах меняются автоматически (относятся ячейкам текущей, а не исходной строки), поэтому нет необходимости набирать вручную каждую формулу. Однако на практике встречаются ситуации, когда адрес ячейки или диапазона ячеек необходимо зафиксировать, чтобы он не изменялся при копировании или автозаполнении ячеек. Для этого необходимо добавить перед но-

мером строки и буквой столбца знак \$. Так, если сделать адрес ячейки B2 абсолютным, он будет выглядеть как \$B\$2. Кроме того, можно зафиксировать в ссылке адрес только столбца (\$B2) или только строки (B\$2). Это называется смешанной адресацией. Для быстрого изменения адресации в готовой формуле дважды щелкните на ней, установите курсор на нужную ссылку и последовательно нажимайте клавишу F4 для изменения типа адреса. Знак \$ можно добавлять в формулы вручную с клавиатуры.

Ошибки в формулах. При работе с формулами в Excel нередко возникают ошибки, связанные не только с правильностью написания формулы, но и с корректным определением адресов ячеек диапазонов с данными. Проведем краткий обзор способов выявления и устранения ошибок.

При обнаружении ошибки в синтаксисе формулы (например, если между аргументами функции будет отсутствовать разделитель «;», будут пропущены или поставлены лишние скобки, обнаружится меньше аргументов, чем того требуется для данной функции) Excel выдаст сообщение об ошибке. Текст сообщения будет различным в зависимости от того, удалось ли Excel определить источник ошибки; когда система не может определить источник ошибки, появляется сообщение представленное на рис. 3.10.

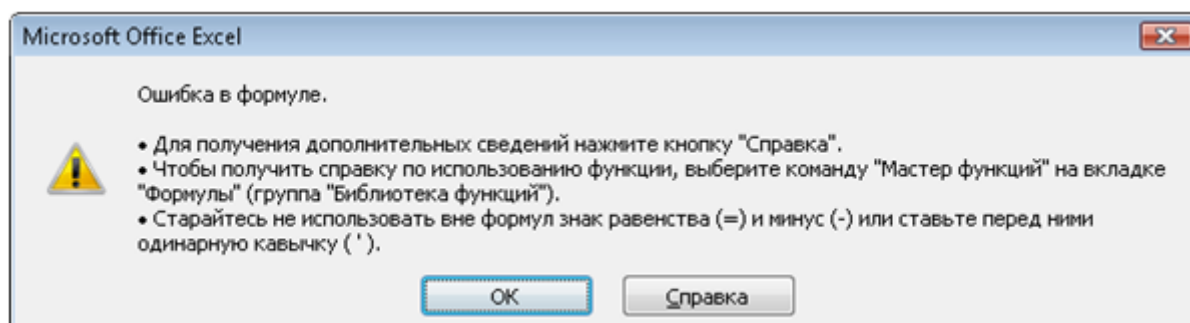


Рис. 3.10. Сообщение об ошибке, не распознанной Excel

В этом случае вернитесь к ячейке с формулой, перепроверьте ее и исправьте ошибку. Иногда системе удастся определить, какие изменения следует внести в формулу, чтобы она стала синтаксически правильной. При этом появляется сообщение, указанное на рис. 3.11.

Проверьте, удовлетворяет ли предлагаемое системой исправление условиям расчета, в зависимости от сделанного вывода согласитесь с автоматическим исправлением, нажав *Да*, или же исправьте ошибку вручную.

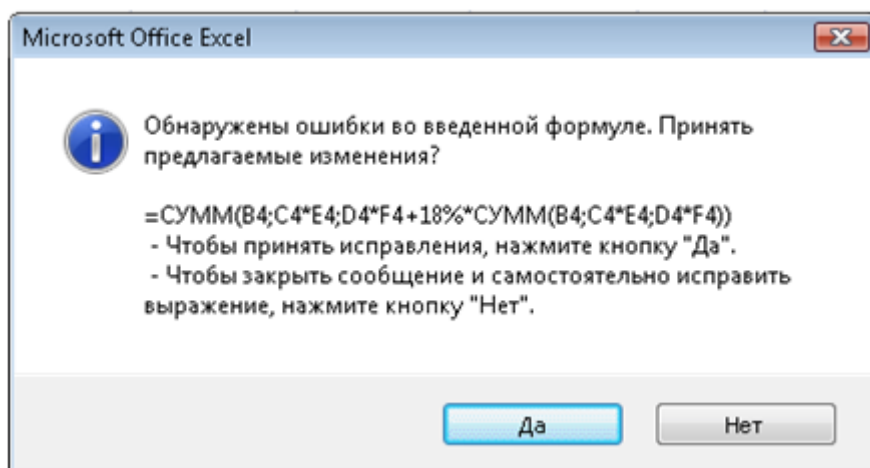


Рис. 3.11. Сообщение о распознанной ошибке

Иногда после введения формулы в ячейке вместо результата появляется текстовое сообщение об ошибке. Это связано с тем, что при проведении вычислений система столкнулась с каким-либо противоречием. Вот список наиболее часто встречающихся сообщений в ячейках:

1. #ЗНАЧ! – ошибка в типе данных, используемых в формуле. Возможно, в одной из ячеек диапазона находится текст;
2. #ИМЯ? – ошибка в имени функции или адресах ячеек и диапазонов, присутствующих в формуле;
3. #ССЫЛКА! – удалены или перемещены ячейки или диапазоны, на которые ссылается формула;
4. #ДЕЛ/0! – при расчете происходит деление на ноль;
5. ##### – данные не умецаются по ширине в ячейку; увеличьте ширину столбца протаскиванием границы заголовка.

Самая опасная ошибка – ввод неправильных адресов ячеек и диапазонов в формулу. Система определяет только математические и синтаксические ошибки, но предугадать, данные каких именно ячеек должны присутствовать в формуле, не может. За этим должны внимательно следить вы.

Проще всего проверить правильность указания адресов ячеек и диапазонов в формуле можно следующим образом. Щелкните дважды на ячейке, содержащей формулу. При этом ячейки и диапазоны, входящие в нее, будут выделены в таблице рамками с маркерами, цвет которых соответствует цвету ссылки в формуле (рис. 3.12).

При указании неправильного адреса перетащите рамку на нужную ячейку (или измените ее размеры протаскиванием маркера при увеличении или уменьшении диапазона).

В	С
Скидка	10%
Цена, руб.	Цена со скидкой, руб.
205	184,5
280	=B5-\$C\$1*B5
130	117

Рис. 3.12. Выделение цветными рамками ячеек, ссылка на которые имеется в формуле

Построение графиков и диаграмм. В Excel имеются средства для создания графиков и диаграмм, с помощью которых вы сможете в наглядной форме представить зависимости и тенденции, отраженные в числовых данных. Кнопки построения графиков и диаграмм находятся в группе *Диаграммы* на вкладке *Вставка*.

Выбирая тип графического представления данных (график, гистограмму, диаграмму того или иного вида), руководствуйтесь тем, какую именно информацию нужно отобразить. Для выявления изменения какого-либо параметра с течением времени или зависимости между двумя величинами следует построить график. Для отображения долей или процентного содержания принято использовать круговую диаграмму. Сравнительный анализ данных удобно представлять в виде гистограммы или линейчатой диаграммы.

Рассмотрим принцип создания графиков и диаграмм в Excel. В первую очередь вам необходимо создать таблицу, данные которой будут использоваться при построении зависимости. Таблица должна иметь стандартную структуру – данные следует поместить в один или несколько столбцов (в зависимости от типа задачи). Для каждого столбца создайте текстовый заголовок. Впоследствии он будет автоматически вставлен в легенду графика.

Word и Excel полностью совместимы: объекты, созданные в одной из этих программ, можно скопировать в документ другого приложения. Так, чтобы перенести из Excel в документ Word любой график или таблицу, достаточно просто выделить ее и выполнить команду *Копировать* контекстного меню, затем перейти в Word, щелкнуть правой кнопкой мыши на месте размещения объекта и выполнить команду *Вставить*.

Сортировка, фильтрация и поиск. Excel часто используется для создания списков, каждая строка которых содержит информацию,

относящуюся к одному объекту. Обратимся к рис. 3.18. Представленный в нем прайс-лист является типичным списком. В списке имеются шапка (заголовки столбцов) и столбцы, содержащие однотипные данные в соответствии с заголовком. В свою очередь, каждая строка представляет собой характеристику объекта, название которого, как правило, присутствует в первом столбце таблицы.

На практике возникают ситуации, когда необходимо отсортировать список по возрастанию или убыванию параметра в одном из его столбцов. Например, прайс-лист на рис. 3.18 можно отсортировать по возрастанию или убыванию цены товара или по названию товара, выстроив его по алфавиту.

	А	В	С
1	Прайс-лист	Скидка	10%
2			
3	<i>Мыши</i>	<i>Цена, руб.</i>	<i>Цена со скидкой, руб.</i>
4	Мышь A4-Tech PS/2 белая/синяя/красная/черная	205	
5	Мышь BenQ M101 mini optical (PS/2, USB)	280	
6	Мышь Genius NetScroll Eye OPTICAL 400 dpi PS/2	130	
7	Мышь Genius NetScroll+Traveler IRIS OPTICAL 400dpi PS/2	445	
8	Мышь Genius Wireless NetScroll+Traveler SE OPTICAL 800dpi PS/2+USB silver	755	
9	Мышь Krauler ML-X450DB professional 3D laser mouse deep blue 6 buttons USB/PS2	550	
10	Мышь Krauler ML-X430SB professional 3D laser mouse silver black 5 buttons USB/PS2	500	
11	Мышь Krauler ML-XL10SB mini laser mouse metal black 3 buttons USB/PS2	500	
12	Мышь Oklik 303M Optical (USB+PS/2) red/silver/white	250	
13	Мышь Logitech S-96/S-96 black Optical PS/2	210	
14	Мышь Logitech Pilot Wheel Optical black (USB+PS/2)	310	
15	Мышь Logitech Cordless R-45 (USB+PS/2) беспроводная	210	
16	Мышь Logitech V100 Optical for Notebook (USB)	620	
17	Трекбол Logitech Cordless Track Men Wheel USB/PS2	1345	
18	Трекбол Logitech Cordless Track Men USB/PS2	1400	
19			

Рис. 3.18. Прайс-лист для расчета цены с учетом скидки

Поиск и замена данных в ячейках Excel осуществляется по таким же принципам, что и поиск текстовых фрагментов Word. Над книгами Excel можно выполнять все стандартные операции: открытие, сохранение, распечатку, за которые отвечают команды *Кнопки Office* в верхнем левом углу окна программы. В Excel их действие абсолютно аналогично.

ГЛАВА 4. ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ VISUAL BASIC FOR APPLICATION

4.1. Основы программирования на языке VBA

Visual Basic For Application (VBA – Visual Basic для приложений) реализован в четырех важнейших компонентах пакета Microsoft Office. VBA представляет собой полностью унифицированный язык программирования и почти полностью унифицированную среду разработки.

VBA включает в себя основные конструкции языка Visual Basic для Windows, встроен в MS Office и приложения Office. VBA является общим для всех приложений MS Office. В VBA применяется объектно-ориентированный подход к разработке приложений. Каждое приложение – Word, Excel, Access, PowerPoint и др. – дополняет средства VBA новыми функциональными возможностями с помощью объектов, свойств и методов, присущих этому приложению. Как объектно-ориентированный язык программирования VBA использует следующие основные понятия: объект, свойство, метод, событие, класс и семейство объектов. Введем определения.

Объект – это объединение данных с кодом, предназначенным для их обработки. Объекты применяются в программировании, потому что они представляют предметы, о внутренней организации которых и о том, как они сделаны, программисту не обязательно даже знать, но он может ими пользоваться и управлять.

Свойства – это параметры объекта, которые ему присущи. Они характеризуют объект. Все свойства объекта имеют свои имена. Различные объекты характеризуются различными свойствами, например, объект «линия» характеризуется длиной и цветом, а объект «окружность» радиусом и цветом. Синтаксис задания свойства объекту имеет вид:

ОБЪЕКТ. СВОЙСТВО = Значение.

Пример:

.....
'изменить имя текущего листа
ActiveSheets . Name = "Январь"

.....
Метод – это программа, которая воздействует на объекты и их параметры. С помощью методов можно, например, закрыть объект, удалить его или показать на экране. Синтаксис записи такой:

ОБЪЕКТ.МЕТОД

Пример:

.....
‘удалить лист
Worksheets(“Обзор”) . Delete
.....

Событие – это действие, распознаваемое объектом, для которого можно запрограммировать отклик (например, щелчок мышью по строке меню или нажатие клавиши). При наступлении события происходит активизация метода, т. е. начинают выполняться связанные с объектом программы и функции.

Класс – это шаблон, на основе которого во время выполнения программы создается объект. Класс определяет свойства и методы объекта. Если взять пример из жизни, то в качестве класса объектов можно принять школьный класс, где объекты – это учащиеся, каждый со своими, присущими только ему свойствами. Также каждому объекту в классе присуще одно или несколько одинаковых свойств, например, возраст, по которому и определяется принадлежность объекта к классу. Объект, который создается на основе класса, называют *экземпляром* класса.

Итак, все приложения Microsoft Office имеют макроязык и средство записи макросов. Все макрокоманды VBA включаются в проекты, которые могут принадлежать различным документам. При создании проекта используется один из трех методов:

- запись макроса при помощи макрорекодера, с последующим его редактированием;
- создание макроса в редакторе VBA;
- создание проекта в редакторе VBA.

Код VBA. Код VBA хранится в проектах. Проекты содержат модули, а модули включают процедуры и функции. Процедуры и функции содержат описания и инструкции на языке VBA.

Каждое приложение MS Office хранит проекты по-своему: Word сохраняет проекты в шаблонах документа (DOT) и в документах

(DOC); Excel сохраняет проекты в рабочих книгах (XLS); Access сохраняет проекты в файлах базы данных (MDB); PowerPoint сохраняет проекты в презентациях (PPT).

Для программирования на языке VBA нет необходимости в обязательном порядке устанавливать какую-либо версию транслятора VB (Visual Basic). Вполне достаточно, если на компьютере будет установлен пакет Microsoft Office 97 и выше. Для создания программы, которая называется макросом, необходимо загрузить Microsoft Word или Microsoft Excel. Далее: *Главное меню-Сервис-Макрос-Редактор Visual Basic*. В окне проекта первой строкой записывается оператор объявления процедуры (программы):

Sub имя() – это стандартный заголовок макроса. Завершающей строкой процедуры является строка End Sub

Для запуска программы из редактора используется клавиша F5; для запуска программы из документа – alt+F8.

Элементы языка VBA. При написании кода VBA следует знать некоторые правила или соглашения, которых следует придерживаться, чтобы избежать ошибок выполнения кода. Первое соглашение – это соглашение по именам. Именовывать нам придется переменные, константы, процедуры, функции, объекты и т. д., поэтому определим правила, которым будем следовать:

- длина имен не должна превышать 255 символов;
- имена должны начинаться с буквы и могут содержать комбинацию букв, цифр и символов;
- имена не должны совпадать со стандартными именами языка Visual Basic;
- имена не должны включать точек, пробелов и следующих символов: ! * / \ @ # \$ %. Символ пробела можно заменить символом подчеркивания _.

Регистр букв в именах не имеет значения, т. е. следующие имена: PRICE, Price и price – эквивалентны.

Примеры правильной записи имен:

H123, Price, Sale

Неправильно записанные имена:

21days – начинается с цифры;

Price Sale – содержит символ пробела;

Dim – совпадает со служебным словом языка.

Дадим также несколько определений элементам VBA.

Ключевые слова – это слово или символ, которые являются элементом языка программирования Visual Basic. Это может быть инструкция, имя функции, оператор, например: SUB, DIM, CONST.

Выражение – это комбинация ключевых слов, операций, констант. Результатом выражения может быть строка, логическое значение или число. Примеры выражений: $X+3$; $Z<>0$; 11 .

Операторы – это специальные языковые конструкции, которые используются для объединения простых выражений в более сложные. Операторы VBA выполняют арифметические, логические операции, а также операции сравнения и конкатенации (сцепления строк).

Типы данных языка VBA. Язык Visual Basic имеет 11 встроенных типов данных (табл 4.1.).

Тип данных определяет, каким образом задаваемая величина хранится в памяти компьютера; сколько разрядов может содержать присваиваемое ей значение; насколько оно может быть велико и может ли иметь дробную часть. Рассмотрим типы данных VBA более подробно.

Boolean использует для хранения значения 2 байта памяти, как и тип *Integer*, хотя может принимать значения либо 0 (ложь – False), либо 1 (True – правда). Для хранения этой информации достаточно 1 бита, а занимает 16 битов.

Integer предназначен для хранения смещенного целого числа, т. е. с указанием знака числа “+” или “-”. Из 16 битов 1 используется для хранения знака, а остальные используются для значащих цифр.

Long используется для хранения смещенного целого числа с двойным разрешением по сравнению с *Integer* и, соответственно, занимает в два раза больше памяти. Из 32 занимаемых в памяти битов 1 предназначается для хранения знака, 31 – для хранения самого числа.

Single используется для вычислений с плавающей точкой. Переменные данного типа обеспечивают точность до 6-го знака после десятичной точки.

Double предназначен для хранения действительного числа с плавающей точкой двойной точности по отношению к *Single*. Для хранения требует вдвое больше памяти, но и диапазон значений расширяется в 8 раз.

Currency предназначен для хранения действительного числа с фиксированной точкой, которая всегда располагается между 4 и 5 цифрами справа. Этот тип специально предназначен для получения минимальных ошибок округления и чаще всего используется в бухгалтерских расчетах.

Таблица 4.1

Основные типы данных языка VBA

Тип данных	Размер в байтах	Разрядность цифр	Описание	Диапазон значений
Boolean	2	1	Логический	True или False (да – нет, включен – отключен)
Integer	2	5	Целое	От (-32768) до 32768
Long	4	10	Длинное целое	От (-2147483648) до 2147483647
Single	4	7	Действительное одинарной точности	От (-3.402823E+38) до (-1.401298E-45) и от 1.401298E-45 до 3.402823E+38
Double	8	15	Действительное двойной точности	От $\pm 1.79769313486232E+308$ до $\pm 4.94065645841247E-324$
Currency	8	19	Денежный	От (-922337203685477.5808) до 922337203685477.5807
Date	8	-	Для дат и времени	От 01.01.100 до 31.12.9999
String	1	+1	Символьный	Длина строки – от 0 до 65535 символов
Object	4	-	Объект	Любой объект
Array	Определяется количеством и размером элементов		Массивы	-
Variant	Определяется записанными данными		Любой из встроенных типов данных	-

Пример: 0.0021 46.7280.

Date предназначен для хранения дат и времени с точностью до секунды. Машинной даты записываются следующим образом: целая часть – количество дней, прошедших от начала отсчета 01.01.1900, а дробная часть – время.

Пример: 05.03.1994 18.00 запишется как 34398.75. При работе с Excel следует помнить, что в нем диапазон используемых дат – от 01.01.1900 до 31.12.2078. Сам VBA позволяет оперировать датами в диапазоне 01.01.100 – 31.12.9999.

String используется для хранения текста по одному байту на символ, добавляя к этому количеству еще один для обозначения конца строки.

Object предназначен для ссылок на любой объект VBA.

Array определяет индексный список величин одного и других типов – массив.

Variant используется по умолчанию. Он задается VBA, если вы сами не укажете этот тип. При этом производится проверка значения, вводимого в память, и его идентификация, поэтому резко возрастает и время выполнения программы, и объем памяти.

Переменные. Переменная – это область памяти, которая предназначена для временного хранения данных и которой присвоено имя. Содержимое переменной может быть изменено в процессе выполнения программы. Данные, хранимые в переменных, могут быть различного типа: числовые данные, строковые, логические и т. д. Тип данных определяет тип переменной.

Область определения переменной определяет, где можно использовать переменную. В VBA есть три уровня определения переменных.

1. Уровень процедуры. Переменные используются только в процедуре, в которой они описаны. Переменная описывается с помощью инструкции DIM, размещенной в теле процедуры.

Например:

Dim x As Single ‘переменная x описана как действительная одинарной точности;

Dim I As Integer ‘переменная I описана как целая.

2. Уровень модуля. Переменные используются только в процедурах модуля, в которых они описаны, но не в других модулях проекта. Переменная описывается с помощью инструкции DIM, размещенной в разделе описаний модуля.

3. Общие переменные. Переменные используются во всех модулях и процедурах проекта. Переменная описывается с помощью инструкции PUBLIC, размещенной в разделе описаний модуля.

Переменные уровня процедуры сохраняют свое значение лишь при выполнении процедуры, при выходе из процедуры значение переменной теряется. Если процедура вызывает другие процедуры, переменная сохраняет свое значение, пока не закончится выполнение и этих процедур. Если переменная уровня процедуры описана с помощью ключевого слова *Static*, она сохраняет свое значение до тех пор, пока программа выполняется в каком-либо модуле. По завершении работы всей программы переменная теряет свою область определения и свое значение. Ее время жизни совпадает с временем жизни переменной уровня модуля.

Если переменная не описана, то ей присваивается тип Variant при вводе имени переменной.

Константы. Константа – это именованная область памяти для хранения данных, изменение которых во время работы программы не допускается. Константа может иметь числовой, логический или строковый тип.

Константа описывается ключевым словом CONST, и при описании константы требуется присвоить ей значение. Попытка изменить в ходе выполнения программы значение константы вызывает ошибку.

Примеры описания констант:

.....

Const Pi as Double = 3.1415926 – описана константа Pi

Const Name = "Visual Basic" – описана константа строкового типа.

.....

Операции. Операции производятся над выражениями, переменными, константами, литералами.

Арифметические операции. Арифметические операции языка Visual Basic описаны в порядке убывания приоритета в табл. 4.2.

Правила записи арифметических выражений в языке Бейсик:

- Все арифметические операции записываются в одну строку. Если в числителе или знаменателе дроби стоит многочлен, его следует заключить в скобки.

- При записи десятичных дробей для отделения целой части от дробной используется десятичная точка.

- Знаки арифметических операций опускать нельзя.

Таблица 4.2

Арифметические и строковые операции языка VBA

Математические и строковые операции VBA в порядке убывания приоритета	Выполняемое действие
\wedge	Возведение в степень
-	Отрицание
*	Умножение
/	Деление
\	Целочисленное деление
Mod	Определение остатка от деления
+	Сложение
-	Вычитание
&	Объединение (конкатенация) строк

• Если в десятичной дроби нет целой части, то ноль перед десятичной точкой можно опустить.

Арифметические операции выполняются слева направо в порядке приоритета. Для изменения естественного приоритета операций применяются скобки (так же, как в арифметике). В Бейсике существуют стандартные функции, при применении которых не требуется никаких дополнительных библиотек или определений.

Логические операции. Логические выражения в математической логике оперируют логическими переменными. Логические переменные имеют в современных языках программирования тип *Boolean*, что означает – булева(я) или логическая. Логическая переменная в процессе выполнения программы может принимать только два значения – True или False. Определить логическую переменную можно, присвоив ей значение логического выражения.

Логическое выражение – выражение, в котором присутствуют операции отношения или знаки логических операций.

Операции отношения:

- = (равно),
- \neq (не равно) ,
- < (меньше),
- > (больше),
- <= (меньше или равно),
- >= (больше или равно),

Например: $a > b$ и $b \leq a + 2$.

Логические выражения могут быть связаны *знаками логических операций*:

- OR – логическое сложение (ИЛИ),
- AND – логическое умножение (И),
- NOT – логическое отрицание (НЕ),
- XOR – исключающее ИЛИ,
- EQV – логическая эквивалентность,
- IMP – логическая импликация.

Операция конкатенации. Операция используется для сцепления двух или нескольких строк, обозначается знаком «+». Результатом будет строка, состоящая из строк – операндов.

Например:

операнд1=”язык”

операнд2=”Visual Basic”

результатом операции

”язык” + ”Visual Basic”

будет строка

”язык Visual Basic”.

Математические функции VBA.

Математические функции языка VBA представлены в табл. 4.3.

Таблица 4.3

Математические функции языка VBA

Функция	Выполняемое действие
Atn(x)	Возвращение арктангенса угла в радианах $\arctg x$
Sin(x)	Возвращение синуса угла в радианах $\sin x$
Cos(x)	Возвращение косинуса угла в радианах $\cos x$
Tan(x)	Возвращение тангенса угла в радианах $tg x$
Exp(x)	Возвращение значения e^x , где x – значение переменной или введенное число
Log(x)	Возвращение натурального логарифма числа $\ln x$
Sqr(x)	Возвращение квадратного корень числа
Abs(x)	Возвращение абсолютного значения числа (по модулю) $ x $

Функции преобразования типов Str() и Val().

Str(*число*) – возвращает значение типа Variant (String), являющееся строковым представлением числа. Обязательный аргумент *число* может задаваться любым допустимым числовым выражением. При преобразовании числа в строку в начале строки обязательно резервируется позиция для знака числа. Если число является положительным, возвращенная строка будет содержать пробел на месте знака.

Заметим, что в качестве допустимого десятичного разделителя функция Str воспринимает только точку (.). При наличии другого десятичного разделителя (например, запятой) для преобразования чисел в строки следует использовать функцию CStr.

Val(*строка*) – возвращает числа, содержащиеся в строке, как числовое значение соответствующего типа.

Обязательный аргумент *строка* является любым допустимым строковым выражением. Функция Val прекращает чтение строки на первом символе, который она не может распознать в качестве части числа. В качестве разделителя целой и дробной части распознается *только* точка (.). Пробелы, символы табуляции и символы перевода строк удаляются из значения аргумента.

В следующем примере возвращается числовое значение 1615198:
Val(" 1615 198-я ул. Н.Й.").

Организация ввода/вывода информации. При создании макрокоманд требуется организовать обмен информацией между макросом и пользователем, между макросом и документом. Эта задача решается с помощью функций InputBox, MsgBox.

Функция InputBox – выводит на экран диалоговое окно, содержащее сообщение и поле ввода, устанавливает режим ожидания ввода текста пользователем или нажатия кнопки, а затем возвращает значение типа String, содержащее текст, введенный в поле. Синтаксис функции InputBox:

InputBox (сообщение [, заголовок] [, текст по умолчанию])

Сообщение – текст в диалоговом окне.

Заголовок – строковое выражение, отображаемое в строке заголовка диалогового окна. Если этот аргумент опущен, в строку заголовка помещается имя приложения.

Текст по умолчанию – строковое выражение, отображаемое в поле ввода, если пользователь не введет другую строку. Если этот аргумент опущен, поле ввода изображается пустым.

Если пользователь нажимает кнопку *OK* или клавишу *ENTER*, функция InputBox возвращает содержимое поля ввода.

Пример. Ввести значение $X=0.5$.

.....
X=Val(InputBox("Введите X"))

.....
В диалоговом окне ввести с клавиатуры 0.5.

Функция MsgBox позволяет выводить результаты счета программы в диалоговом окне.

Синтаксис:

MsgBox (сообщение [, кнопки] [, заголовок]).

Сообщение – строковое выражение, отображаемое в диалоговом окне.

Кнопки – числовое выражение, представляющее сумму значений, которые указывают число и тип отображаемых кнопок, тип используемого значка, основную кнопку. Значение по умолчанию этого аргумента равняется 0.

Заголовок – строковое выражение, отображаемое в строке заголовка диалогового окна. Если этот аргумент опущен, в строку заголовка помещается имя приложения.

Будем использовать в качестве аргумента функции только *сообщение*, представляющее собой конкатенацию (сцепление)(+) двух строк: комментария к результату, взятого в кавычки, и строки, полученной из числового выражения с помощью функции Str.

Пример. Вывести результат на экран монитора.

.....
MsgBox("Ответ=" + Str(y))

.....
Использование окна отладки. Окно отладки можно вывести на экран при помощи команды *Вид – Окно отладки* или нажав клавиши <Ctrl>+<G> .

Для вывода текущих значений в окно Проверка непосредственно из макроса необходимо ввести в нужном месте кода макроса команду *Debug с* единственным методом *Print*. Синтаксис

Debug.Print выражение.

В месте, где установлена эта команда, произойдет прерывание выполнения макроса, и в окне *Проверка* появится текущее значение переменной или выражения.

Пример.

.....
Debug.Print y

.....
Использование объектов Excel. Объект *Worksheet* – это лист рабочей книги. Он входит в коллекцию всех листов рабочей книги – семейство *Sheets*. Этот объект можно использовать для ввода/вывода данных. В этом случае к этому объекту можно применить метод *Select*, позволяющий выбирать конкретный лист.

Пример.

`Sheets (“Лист1”) . Select`

Если макрос написан для определенного листа, то удобно исходные данные поместить в соответствующие ячейки данного листа. В этом случае метод *Select* можно не использовать. При этом результаты счета будут помещены на этот же листе автоматически.

Поскольку одиночная ячейка не представлена собственным объектом, то доступ к ячейке можно осуществить с помощью свойства *Cells*, указывая номер строки и столбца. Для присвоения некоторой переменной *x* значения, находящегося в ячейке A2 используем оператор:

`X=Cells(2,1)`

Пример. Вывести значения переменных *s* и *p* во вторую и третью строки второй колонки (столбец B) третьего листа соответственно.

.....
`Sheets (“Лист3”).Select`

‘переходим на третий лист

`Cells(2,8)=s`

‘выводим s

`Cells(3,8)=p`

‘выводим p
.....

Часто используется свойство *Value*, возвращающее содержимое ячейки.

Пример.

`Cells(2,8).Value=s`

‘выводим s

4.2. Операторы языка VBA и реализация основных алгоритмических конструкций

В языке VBA используется множество инструкций или операторов. Они служат для выполнения каких-либо стандартных действий. Условно из них можно выделить группы.

Оператор комментариев. Данный оператор не выполняет в программе никаких действий и может содержать любой текст. Инст-

рукция комментариев имеет два вида: это инструкция, начинающаяся с ключевого слова REM, за которым следует произвольный текст или знак апострофа ('), который ставится в начале строки или после любого оператора, и за ним также можно писать любой текст.

Пример.

REM произвольный текст комментария

' произвольный текст комментария

A1=5 ' комментарий к строке оператора

Оператор присваивания. Присваивает выражение переменной или константе. Инструкции присвоения всегда включают знак равенства (=).

Синтаксис:

ПЕРЕМЕННАЯ = ВЫРАЖЕНИЕ

Вначале вычисляется выражение, а затем результат запоминается в переменной.

Выделяют следующие алгоритмические структуры:

- линейная, когда все действия выполняются последовательно (без условий) от начала до конца;
- разветвленная, когда на некотором шаге встречается условие, в зависимости от которого процесс решения может пойти по одной из нескольких ветвей;
- циклическая, когда некоторые фрагменты программы должны выполняться несколько раз.

В алгоритмическом языке установлены следующие геометрические фигуры для обозначения действий (рис. 4.1).

Программирование алгоритмов линейной структуры. Программа линейного вычислительного процесса представляет собой последовательность операторных строк, содержащих операторы присваивания, а также средства ввода и вывода.

При составлении линейных программ необходимо:

- выделить начальные данные;
- установить последовательность вычислений;
- определить исходные данные до момента их использования;
- желательно снабжать программу комментариями для удобства ее чтения.

Пример. Вычислить значение выражения для любых a и b.

Программа

Sub Lin()

Dim a As Double, x As Double

```

a = Val(InputBox("Введите a")) 'ввод значения переменной a
x = Val(InputBox("Введите x")) 'ввод значения переменной x
b = 1 / (x) ^ (1 / 4) 'вычисляем значение b'
c = Sin(a ^ 2 + b ^ 2) 'вычисляем значение C
MsgBox ("Ответ=" + Str(c))
End Sub

```

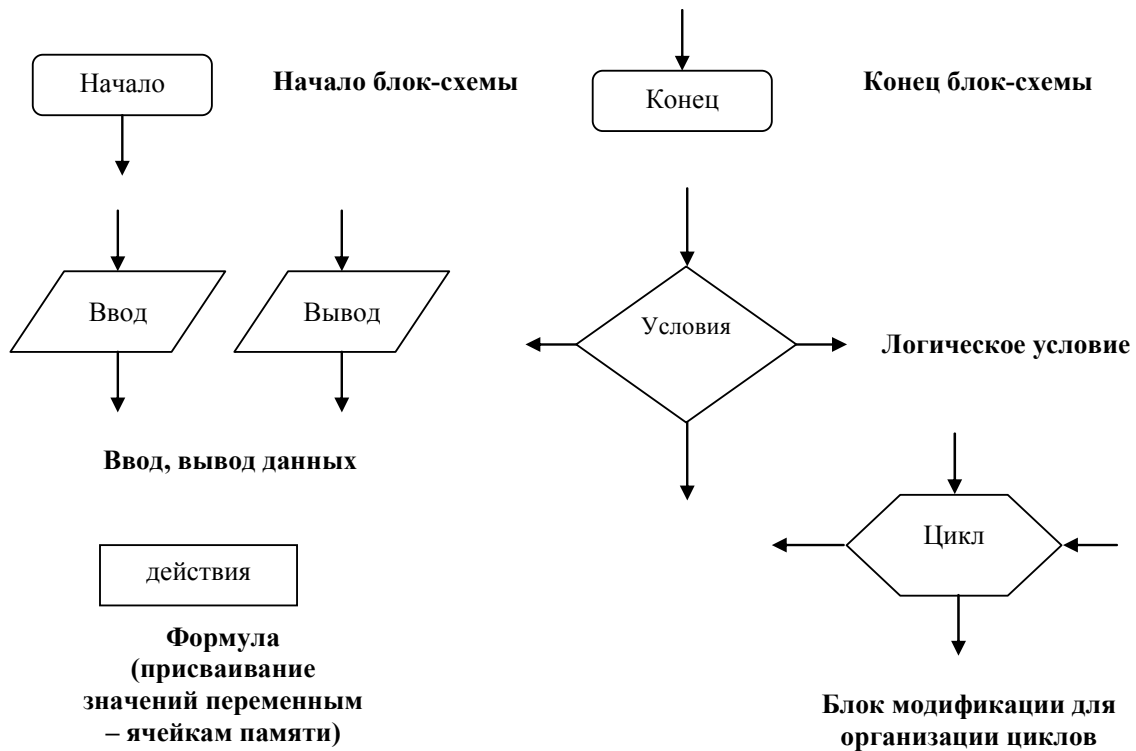


Рис. 4.1. Геометрические фигуры для обозначения действий

Пример. Составить программу для вычисления площади S треугольника ABC по заданным значениям сторон a, b, c .

Программа

```
Sub Geron()
```

```
Dim a, b, c, p, s As Double
```

```
a = Val(InputBox("Введите a")) 'ввод значения переменной a
```

```
b = Val(InputBox("Введите b")) 'ввод значения переменной b
```

```
c = Val(InputBox("Введите c")) 'ввод значения переменной c
```

```
'P – полупериметр, S – площадь
```

```
p = (a + b + c)
```

```
s = Sqr(p * (p - a) * (p - b) * (p - c))
```

```
Cells(1, 1) = "Площадь="
```

```
Cells(1, 2) = s
```

```
End Sub
```

Операторы управления порядком выполнения команд. Обычно выполнение макроса начинается с выполнения инструкций SUB или FUNCTION и продолжается в порядке следования строк с операторами до тех пор, пока не встретится инструкция END SUB или END FUNCTION.

Но часто при проектировании макросов необходимо выполнять различные варианты программы в зависимости от значения каких-либо переменных или параметров (условий), задаваемых пользователем во время выполнения программы. На этот случай в VBA предусмотрен ряд операторов, предназначенных для изменения порядка выполнения операторов в программе.

Оператор безусловного перехода *GoTo* позволяет изменить порядок выполнения команд. Задаёт безусловный переход на указанную строку внутри той процедуры, в которой она находится.

Синтаксис:

GoTo строка

Обязательный аргумент *строка* может быть любой меткой строки или номером строки.

Метка строки – это строка программы, в начале которой указывается имя метки и двоеточие (:). Метки должны иметь уникальные имена в пределах одной процедуры.

Заметим, что слишком большое число инструкций *GoTo* затрудняет чтение и отладку программ. По возможности следует использовать структурные управляющие инструкции (*Do...Loop*, *For...Next*, *If...Then...Else*, *Select Case*). Инструкция *GoTo* в VBA применяется, главным образом, для обработки ошибок.

Условный оператор *If... Then... Else* выполняет определённые инструкции или наборы инструкций в зависимости от значения заданных условий.

Синтаксис:

If *условие* Then [*оператор1*] [Else *оператор2*].

Различают условные операторы: *строчный* и *блочный*. Строчный оператор используется в том случае, когда при разветвлении программы необходимо на каждой ветке выполнить по одному оператору, а блочный необходим тогда, когда операторов несколько.

Обязательный параметр *условие* представляет собой логическое выражение, которое возвращает значение (истина) TRUE или (ложь) FALSE. Если *условие* равно TRUE, то выполняется *оператор1*, если *условие* равно FALSE, то выполняется *оператор2*. Далее выполняет-

ся, если иное не предусмотрено программой в *операторах 1* или *2*, инструкция, следующая по порядку за строкой с инструкцией *If...Then...Else*.

Часть оператора *If*, а именно ключевое слово *Else* и следующие за ним инструкции можно опустить, тогда при условии, равном *FALSE* выполняется оператор, следующий за строкой с инструкцией *If*.

Синтаксис:

If условие Then [оператор]

Заметим, что в квадратных скобках обозначены те элементы инструкции, которые не являются обязательными для записи.

В реальных случаях часто требуется обработка нескольких инструкций при выполнении условия. В этих случаях предпочтительнее блочный вид инструкции *If*.

Синтаксис:

If условие Then

[Блок операторов]

[Else

[Блок операторов]]

End If

Блочная структура *If* удобнее, так как каждый блок операторов может включать в себя произвольное число утверждений. Кроме того, допускается вложение структур *If*.

Оператор *Select Case* – это оператор выбора из многих возможных вариантов. Оператор *Select Case* выполняет одну из нескольких групп операторов в зависимости от значения выражения. В языках высокого уровня такой оператор называется переключателем.

Синтаксис:

Select Case выражение

[Case список условий-1

[операторы-1]]

[Case список условий-2

[операторы-2]]

...

[Case список условий-n

[операторы-n]]

[Case Else

[операторы_ else]]

End Select

Выражение – обязательный параметр, может быть любым выражением любого типа данных VBA, включая числовой, строковый или логический.

Список условий – обязательный при наличии инструкции CASE. Содержит результаты вычисления выражения, записанные в одном из видов:

1. Case КОНСТАНТА1, КОНСТАНТА2, КОНСТАНТА3, ...
2. Case Is ЗНАК_ОТНОШЕНИЯ КОНСТАНТА
3. Case КОНСТАНТА1 to КОНСТАНТА2

Выполняется оператор Select Case следующим образом. Сначала вычисляется значение выражения, стоящего после ключевых слов Select Case, затем производится проверка на совпадение значения выражения и одного из Case условий. В случае совпадения выполняются операторы, помещенные за этим Case условием. В случае если совпадения не обнаружено ни в одном из Case условий, то выполняются операторы, находящиеся после ключевых слов Case Else. Если список выражений в Case условии записан в первой форме, то необходимо, чтобы результат вычислений выражения равнялся одной константе из списка, если Case условие записано во второй форме, то необходимо, чтобы выполнялось условие отношения между значением выражения и константой. Если же в третьей форме, то необходимо выполнение соотношения

КОНСТАНТА1<=выражение<=КОНСТАНТА2.

Программирование алгоритмов разветвляющейся структуры. Алгоритм разветвляющейся структуры – это алгоритм такого процесса, в котором его реализация осуществляется по одному из заранее предусмотренных (возможных) направлений в зависимости от исходных условий или промежуточных результатов. Каждое отдельное направление обработки информации в таком процессе называется ветвью. Выбор ветви определяется проверкой выполнения логического условия, определяющего свойства исходных данных или промежуточных результатов. В каждом конкретном случае процесс обработки данных выполняется лишь по одной из ветвей. Для их реализации используются операторы управления порядком выполнения команд, рассмотренные выше.

Пример. Дано действительное число x. Вычислить y.

Программа.

Sub Raz2()

Dim x, y As Double

x = Val(InputBox("Введите x")) 'ввод значения переменной x


```

If x > 0 Then y = Sin(x) Else y = 2 * x
MsgBox ("Значение y=" + Str(y#))
End Sub

```

Пример. Дано действительное число x. Вычислить y.

Программа

```

Sub Raz3()
Dim x, y As Double
x = Val(InputBox("Введите x"))
If x < 0.1 Then y = Cos(x ^ 2) Else If x > 0.1 Then y = Exp(x) Else y
= x ^ 3 - 2
MsgBox ("Значение y=" + Str(y#))
End Sub

```

Пример. Даны три числа. Выяснить, существует ли треугольник с такими сторонами.

Программа

```

Sub Treug()
Dim a, b, c As Double
a = Val(InputBox("Введите сторону a"))
b = Val(InputBox("Введите сторону b"))
c = Val(InputBox("Введите сторону c"))
If (a + b) > c And (b + c) > a And (a + c) > b Then MsgBox ("Тре-
угольник существует") Else MsgBox ("Треугольник не существует")
'оператор печатать в одной строке
End Sub

```

Пример. Дано действительное число x. Вычислить z, y, h.

Программа

```

Sub Raz3()
Dim x, z, y, h As Double
x = Val(InputBox("Введите x"))
If x > 0.8 Then
z = 2 * Sin(x)
y = Log(x) + 4 * x
h = Cos(x)
Else
If x = 0.8 Then
z = Sqr(Sin(x))
y = Cos(x ^ 2) + x

```

```

h = 2 * x
Else
z = Abs(x - 2)
y = 2 + x ^ 2 * Sin(x)
h = 0
End If
End If
Cells(1, 1) = "x=": Cells(1, 2) = x
Cells(2, 1) = "z=": Cells(2, 2) = z
Cells(3, 1) = "y=": Cells(3, 2) = y
Cells(4, 1) = "h=": Cells(4, 2) = h
End Sub

```

Программирование алгоритмов циклической структуры.

Решение многих практических задач сводится к выполнению вычислений по одним и тем же зависимостям, но при разных значениях входящих в них величин. Такой вычислительный процесс называется циклическим, а многократно повторяющиеся участки этого процесса называются циклами.

Различают *регулярные* циклы с управляющим параметром (с известным числом повторений), условием окончания которого является достижение параметром цикла своего конечного значения; *итерационные* циклы, в которых условие повторения или окончания цикла задается по некоторому результату, например, пока не будет достигнута точность вычислений. Реализуются циклы с помощью специальных операторов цикла.

Оператор цикла *While... Wend*

Синтаксис:

While условие

[операторы]

Wend

Условие – обязательный элемент. Числовое выражение или строковое выражение, которое имеет значение True или False.

Операторы – необязательный элемент. Один или несколько операторов, выполняемых, пока условие имеет значение True.

Этот оператор называют оператором цикла с предусловием. Выполняется оператор *While... Wend* следующим образом. Если *условие* имеет значение *True*, выполняются все операторы до инструкции *Wend*. Затем управление возвращается инструкции *While* и вновь проверяется *условие*. Если *условие* по-прежнему имеет значение *True*,

процесс повторяется. Если оно не имеет значение *True*, выполнение возобновляется с инструкции, следующей за инструкцией *Wend*. В связи с этим элемент *условие* здесь является условием выполнения цикла. Циклы *While... Wend* могут иметь любую глубину вложенности.

Оператор цикла *Do...Loop* также используется для выполнения наборов операторов неопределенное число раз. Оператор имеет синтаксис:

```
Do  
[операторы]  
Loop Until [условие]
```

Условие – необязательный элемент. Числовое или строковое выражение, которое имеет значение *True* или *False*.

Операторы – один или несколько операторов, выполнение которых повторяется, пока условие не приобретет значение *True*.

Этот оператор называют оператором цикла с постусловием. Выполняется оператор *Do...Loop* следующим образом. Если *условие* имеет значение *False*, выполняются все операторы после инструкции *Do*. Затем управление передается инструкции *Until* и вновь проверяется *условие*. Если *условие* по-прежнему имеет значение *False*, процесс повторяется. Если оно имеет значение *True*, управление передается следующему за инструкцией *Loop Until* оператору. В связи с этим элемент *условие* здесь является условием выхода из цикла.

Часто при составлении макроса заранее известно количество повторений группы операторов, в таких случаях можно использовать инструкцию *For...Next*.

Оператор *For...Next* используется для выполнения наборов операторов указанное число раз. Циклы *For* используют в качестве счетчика переменной, значение которой увеличивается или уменьшается при каждом выполнении цикла на указанное значение.

Синтаксис:

```
For счетчик = начало To конец [step шаг]  
[операторы]  
Next [счетчик]
```

Счетчик – обязательный элемент. Это должна быть числовая переменная. Она не может иметь тип *Boolean* или быть элементом массива.

Начало – обязательный элемент, содержит начальное значение переменной *счетчик*.

Конец – обязательный элемент, содержит конечное значение переменной *счетчик*.

Шаг – элемент необязательный, это значение, на которое изменяется счетчик при каждом выполнении тела цикла. Если это значение не задано, по умолчанию *шаг* равен единице. *Шаг* может быть как положительным, так и отрицательным.

Операторы – необязательный элемент. Один или несколько операторов между *For* и *Next*, которые выполняются указанное число раз.

Инструкция *For...Next* работает следующим образом: начальное значение элемента *счетчик* сравнивается с конечным значением. Если *шаг* положителен и начальное значение меньше конечного или если шаг отрицателен и начальное значение больше конечного, то управление передается внутрь тела цикла. После выполнения всех операторов в теле цикла значение *шаг* добавляется к текущему значению переменной *счетчик*. После этого операторы тела цикла либо выполняются еще раз (на основе того же условия, которое привело к началу выполнения цикла), либо цикл завершается и выполнение продолжается с оператора, следующего за *Next*.

Допускается вложение циклов *For...Next* (один цикл *For...Next* располагается внутри другого). Счетчик каждого цикла должен иметь уникальное имя.

Пример. Составить таблицу значений функции $y=x^2$ на отрезке $[2,8]$ с шагом 0.5.

```
Программа  
Sub Tablica()  
Dim x, y As Double, i As Integer  
i = 1  
Cells(1, 1) = "X": Cells(1, 2) = "Y"  
For x = 2 To 8 Step 0.5  
y = x ^ 2  
i = i + 1  
Cells(i, 1) = x: Cells(i, 2) = y  
Next x  
End Sub
```

Пример. Вычислить сумму квадратов n чисел: 1, 4, 9, 16, 25, Число n задать произвольно при вводе.

```
Программа  
Sub sum()
```

```

Dim n, i As Integer, s As Double
n = Val(InputBox("Введите количество слагаемых n"))
s = 0
For i = 1 To n
    s = s + i ^ 2
Next i
MsgBox ("Сумма s=" + Str(s#))
End Sub

```

Пример. Вычислить произведение квадратов n чисел: 1, 4, 9, 16, 25, Число n задать произвольно при вводе.

Программа

```

Sub Proiz()
Dim n, i As Integer, p As Double
n = Val(InputBox("Введите количество слагаемых n"))
p = 1
For i = 1 To n
    p = p * i ^ 2
Next i
MsgBox ("Произведение p=" + Str(p#))
End Sub

```

Вложенные циклы. Внутри одного цикла могут находиться один или несколько других циклов. В этом случае охватывающий цикл называется *внешним*, а вложенные в него циклы называются *внутренними*. Правила организации как внешнего, так и внутренних циклов аналогичны правилам организации простого цикла. Параметры внешнего и внутреннего циклов изменяются не одновременно, т. е. при одном значении параметра внешнего цикла параметр внутреннего последовательно принимает все возможные значения. При организации вложенных циклов необходимо следить за тем, чтобы область действия внутреннего цикла не выходила за область действия внешнего цикла.

Пример. Составить таблицу значений функции на отрезке [1,4] с шагом h=0.5.

Программа

```

Sub VICircle()
Dim x, s, a, b, h As Double, i, n, k As Integer
a = Val(InputBox("Введите a"))
b = Val(InputBox("Введите b"))
h = Val(InputBox("Введите шаг h"))

```

```

n = Val(TextBox("Введите количество слагаемых n"))
k = 1
Cells(1, 1) = "X": Cells(1, 2) = "S"
For x = a To b Step h
    s = 0
    For i = 1 To n
        s = s + x / i
    Next i
    k = k + 1
    Cells(k, 1) = x: Cells(k, 2) = s
Next x
End Sub

```

Массивы. Массивом называется последовательность величин одного типа данных, обозначаемая одним именем. Чтобы получить доступ к нужному элементу массива, нужно указать имя массива и индекс этого элемента. Имя массива образуется так же, как имя переменной. Различают *одномерные* и *двумерные* массивы. Одномерный массив – это список переменных, двумерный массив – таблица, имеющая строки и столбцы. Элементы одномерного массива снабжаются одним индексом, заключенным в круглые скобки. Он определяет порядковый номер элемента в массиве. Элементы двумерного массива снабжаются двумя индексами, заключенными в квадратные скобки и разделенными запятой. Первый индекс – номер строки, второй – номер столбца, на пересечении которых расположен элемент в таблице (матрице). Например, $A(5)=3$, $B(2,3)=6$. Все используемые массивы должны быть описаны до их использования в программе.

Оператор описания *DIM* применяется для описания переменных и выделения для них памяти.

Синтаксис:

Dim имя переменной [(индексы)] [As тип]

Имя переменной – обязательный элемент, содержит имя переменной;

Индексы – необязательный элемент. Если индексы отсутствуют, то описана простая переменная;

Тип – необязательный элемент, это тип данных переменной. Если тип данных не указан, по умолчанию переменная получает тип Variant.

Пример.

Dim Num As Integer явно объявляет переменную целого типа.

Если необходимо описать массив, то с помощью индексов указывается размерность массива.

Пример.

```
Dim B(3,3) As Single
```

```
Dim A(11) As Integer
```

Первый оператор объявляет двумерный массив 3x3 матрицу, состоящую из действительных чисел. Вторым оператором объявляется одномерный массив (вектор) из 12 целых чисел, причем по умолчанию первый элемент массива будет A(0), а последний A(11). В этом случае говорят, что 0 – базовый индекс. Можно изменить базовый индекс, написав в области описания модуля инструкцию *Option base 1*. После этого индексы массивов будут нумероваться с единицы. Другим способом изменения базового индекса является использование ключевого слова *To* при объявлении массива.

Пример.

```
Dim B(1 To 3, 1 To 3) As Single
```

```
Dim A(1 To 12) As Integer
```

Массив в программе можно определить поэлементно.

Пример.

```
Dim B(1 To 2, 1 To 3) As Single
```

```
B(1,1)=2: B(1,2)=5: B(1,3)=4:
```

```
B(2,1)=7: B(2,2)=1: B(2,3)=3
```

Для определения одномерных массивов можно использовать функцию *Array*, преобразующую список элементов, разделенных запятыми, в вектор из этих значений, и присваивающую им тип *Variant*.

Пример.

```
Dim A As Variant
```

```
A= Array(10,40,60)
```

Здесь A(0)=10, A(1)=40, A(2)=60.

Удобным способом ввода/вывода массивов является использование рабочего листа Excel. Исходный массив считывается из соответствующих ячеек рабочего листа, выводится массив также в ячейки рабочего листа. В этом случае используется оператор цикла *For...Next* и свойство *Cells()* для указания конкретного местоположения элемента массива. Если макрос написан для того же рабочего

листа, где располагаются элементы массива, то указание на выбор этого рабочего листа можно опускать.

Рассмотрим примеры реализации некоторых алгоритмов обработки массивов.

Пример. Дан массив А из пяти чисел. Найти максимальный элемент массива и его индекс.

Программа

```
Sub Massiv()  
Dim a(5) As Single  
Dim max As Single, i, k As Integer  
'ввод массива  
For i = 1 To 5  
a(i) = Cells(1, i + 1)  
Next i  
max = a(1)  
k = 1          'индекс максимального элемента  
For i = 1 To 5  
If max < a(i) Then max = a(i): k = i  
Next i  
MsgBox ("Максимальный элемент=" + Str(max!))  
MsgBox ("Индекс максимального элемента k=" + Str(k))  
End Sub
```

Пример. Дан массив А из пяти чисел. Найти количество положительных элементов массива.

Программа

```
Sub Kol()  
Dim a(5) As Single  
Dim i, k As Integer  
For i = 1 To 5  
a(i) = Cells(1, i + 1)  
Next i  
k = 0  
For i = 1 To 5  
If a(i) > 0 Then k = k + 1  
Next i  
MsgBox ("Количество положительных элементов k=" + Str(k))  
End Sub
```


Пример. Дан вектор A(5). Найти элементы вектора B(5) по следующему правилу: $b_i = \sin(a_i), i=1, \dots, 5$.

Программа

```
Sub NewMassiv()  
Dim a(5), b(5) As Single  
Dim i As Integer  
'ввод массива a(5)  
For i = 1 To 5  
a(i) = Cells(1, i + 1)  
Next i  
For i = 1 To 5  
b(i) = Sin(a(i))  
Next i  
'вывод нового массива b(5)  
Cells(3, 1) = "Массив b(5)"  
For i = 1 To 5  
Cells(3, i + 1) = b(i)  
Next i  
End Sub
```

Пример. Найти скалярное произведение двух векторов A(5) и C(5).

Программа

```
Sub SkalProiz()  
Dim a(5), c(5) As Single  
Dim i As Integer, s As Single  
'ввод массива a(5)  
For i = 1 To 5  
a(i) = Cells(1, i + 1).Value  
Next i  
'ввод массива c(5)  
For i = 1 To 5  
c(i) = Cells(7, i + 1).Value  
Next i  
s = 0  
For i = 1 To 5  
s = s + a(i) * c(i)  
Next i  
MsgBox ("Скалярное произведение s=" + Str(s))  
End Sub
```

Пример. Дана матрица A(2,3). Найти минимальный элемент.

Программа

```
Sub MinMat()  
Dim a(2, 3) As Single  
Dim min As Single, i, j As Integer  
'ввод матрицы  
For i = 1 To 2  
For j = 1 To 3  
a(i, j) = Cells(i + 1, j)  
Next j  
Next i  
min = a(1, 1)  
For i = 1 To 2  
For j = 1 To 3  
If a(i, j) < min Then min = a(i, j)  
Next j  
Next i  
MsgBox ("Минимальный элемент min=" + Str(min))  
End Sub
```

Пример. Дана матрица A(2,3) и вектор B(3). Найти произведение A*B.

Программа

```
Sub UmnMatNaVec()  
Dim a(2, 3) As Single, b(3) As Single, c(2) As Single  
Dim s As Single, i, j As Integer  
'ввод матрицы  
For i = 1 To 2  
For j = 1 To 3  
a(i, j) = Cells(i + 1, j)  
Next j  
Next i  
'ввод массива  
For i = 1 To 3  
b(i) = Cells(4, i + 1)  
Next i  
For i = 1 To 2  
s = 0  
For j = 1 To 3  
s = s + a(i, j) * b(j)
```

```

Next j
c(i) = s
Next i
'Вывод нового массива b(5)
Cells(6, 1) = "Массив c(2)"
For i = 1 To 2
Cells(6, i + 1) = c(i)
Next i
End Sub

```

Подпрограммы и функции. При разработке макросов часто требуется выполнять одни и те же законченные действия в различных его частях. Чтобы избежать многократного набора кода для этих действий, целесообразно описать их в виде подпрограммы и обращаться к ней по мере необходимости. Различают подпрограммы-процедуры и подпрограммы-функции. Описываются подпрограммы при помощи инструкций *Sub...End Sub* – это подпрограмма-процедура, инструкций *Function...End Function* – это подпрограмма-функция.

Подпрограмма *Function* и подпрограмма *Sub* являются самостоятельной программой, которая может получать аргументы, выполнять последовательность операторов и изменять значения своих аргументов. Различие между процедурами и функциями состоит в том, что функция всегда возвращает значение и процедура *Function* может применяться в правой части выражения, как и любая другая встроенная функция, например, Sqr, Cos или Chr, а процедура *Sub* не может. Для работы подпрограммам иногда необходимо передать данные, которые носят названия параметров или аргументов подпрограммы. Параметры передаются в подпрограммы при ее вызове при помощи перечисления их после имени подпрограммы. При этом описание действий в подпрограмме осуществляется с использованием формальных параметров. Обращение к подпрограмме осуществляется с фактическими параметрами, которые должны соответствовать формальным по числу, типу и месту расположения.

Оператор *Sub...End Sub* описывает имя, аргументы и текст программы, составляющий тело процедуры *Sub*.

Синтаксис инструкции:

```

Sub имя [(список аргументов)]
    операторы
End Sub

```

Имя – обязательный элемент, это имя процедуры *Sub*, удовлетворяющее =: стандартным правилам именования переменных.

Список аргументов – необязательный элемент. Список переменных, представляющий параметры, которые передаются в процедуру Sub при ее вызове (формальные параметры). Имена переменных разделяются запятой.

Операторы – любая группа операторов, выполняемых в процедуре Sub, находятся между Sub и End Sub.

Не допускается определение процедуры Sub внутри другой процедуры Sub или Function.

Оператор вызова подпрограмм Call – позволяет обратиться к конкретной процедуре по имени процедуры.

Синтаксис оператора:

Call имя подпрограммы [параметры]

Имя подпрограммы – обязательный элемент, содержит имя вызываемой процедуры.

Параметры – необязательный элемент, это разделяемый запятыми список переменных, массивов или выражений, передаваемых в процедуру (фактические параметры).

Для передачи в процедуру полного массива следует воспользоваться именем массива с пустыми скобками.

Оператор Function...End Function описывает имя, аргументы и текст программы, составляющий тело подпрограммы – функции Function.

Синтаксис инструкции:

Function имя [(список аргументов)] [As тип]

операторы

[имя = выражение]

End Function

Имя – обязательный элемент. Содержит имя подпрограммы-функции Function, удовлетворяющее стандартным правилам именования переменных.

Список аргументов – необязательный элемент, это список переменных, представляющий параметры, которые передаются в подпрограмму Function при ее вызове (формальные параметры). Имена переменных разделяются запятой.

Тип – необязательный элемент. Тип данных значения, возвращаемого подпрограммой Function.

Операторы – элемент, содержащий любую группу операторов, выполняемых внутри процедуры Function.

Выражение – возвращаемое значение подпрограммой Function.

Использование подпрограммы Function (нестандартной функции) аналогично использованию стандартных функций. Обращение к ней можно записать, например, в правой части оператора присваивания, при этом указываются имя подпрограммы – функции и в круглых скобках фактические параметры т. е. *список аргументов*, заменяющих формальные параметры в операторе Function

Пример. Вычислить полярные координаты точки, если известны ее декартовы координаты. Связь полярных и декартовых координат определяется по формулам: $x=R \cos \alpha$; $y=R \sin \alpha$.

Программа

```
Sub PolKoord()  
Dim x As Double, y As Double  
Cells(1, 7) = "Полярные координаты"  
Cells(1, 7).Interior.ColorIndex = Int(Rnd * 10)  
x = Val(InputBox("Введите x"))  
y = Val(InputBox("Введите y"))  
Cells(2, 7) = "Радиус"  
Cells(2, 8) = Radius(x, y)  
Cells(3, 7) = "Угол"  
Cells(3, 8) = Ugol(x, y)  
End Sub  
  
Function Radius(x As Double, y As Double) As Double  
Radius = (x ^ 2 + y ^ 2) ^ (1 / 2)  
End Function  
  
Function Ugol(x As Double, y As Double) As Double  
Dim I1, I2, I3 As Boolean  
Ugol = Atn(y / x) * 180 / (3.14152654)  
I1 = x < 0 And y > 0  
I2 = x < 0 And y < 0  
I3 = x > 0 And y < 0  
If I1 Or I2 Then Ugol = Ugol + 180  
If I3 Then Ugol = Ugol + 360  
End Function
```

Пример. Решить систему уравнений методом Гаусса.

Программа

```
Sub Method()  
Dim a(3, 4), x(3), a1(3, 3)  
n = 3  
For i = 1 To n  
For j = 1 To n + 1
```

```

a(i, j) = Cells(i + 1, j)
Next j
Next i
For k = 1 To n - 1
Call max(a(), n, k)
Call gauss(a(), n, k)
Next k
Call obr(a(), n, x())
Cells(6, 1) = "Вектор x"
For i = 1 To n
Cells(6, i + 1) = x(i)
Next i
End Sub
Sub max(a(), n, k)
h = Abs(a(k, k))
For i = k To n
If Abs(a(i, k)) > h Then h = Abs(a(i, k)): l = i
Next i
For j = 1 To n + 1
s = a(k, j): a(k, j) = a(l, j): a(l, j) = s
Next j
End Sub
Sub gauss(a(), n, k)
For i = k + 1 To n
d = a(k, k)
w = a(i, k)
For j = k To n + 1
a(i, j) = a(i, j) - a(k, j) / d * w
Next j
Next i
End Sub
Sub obr(a(), n, x())
x(n) = a(n, n + 1) / a(n, n)
For k = n - 1 To 1 Step -1
s = 0
For j = k + 1 To n
s = s + a(k, j) * x(j)
Next j
x(k) = (a(k, n + 1) - s) / a(k, k)
Next k
End Sub

```

ЗАКЛЮЧЕНИЕ

Информатика – молодая научная дисциплина, изучающая вопросы, связанные с поиском, сбором, хранением, преобразованием и использованием информации в самых различных сферах человеческой деятельности. Информатика связана с вычислительной техникой, компьютерными системами и сетями, так как именно компьютеры позволяют порождать, хранить и автоматически перерабатывать информацию в таких количествах, что научный подход к информационным процессам становится одновременно необходимым и возможным.

В представленном учебном пособии были рассмотрены теоретические основы этой емкой дисциплины: были представлены основные понятия информатики, дано общее описание информационных технологий, рассмотрена архитектура персонального компьютера, а также современное программное обеспечение специального и общего назначения.

В современных условиях глобальной информатизации общества и производства все большее значение приобретают надежные и удобные средства для автоматизации работ, связанных с накоплением, обработкой и передачей информации. К таким средствам, в первую очередь, относятся программы для работы с текстами – текстовые редакторы и проектирования электронных таблиц – табличные процессоры.

Текстовые редакторы предназначены для ввода, обработки, хранения и печатания текстовой информации в удобном для пользователя виде. Они предоставляют пользователю все необходимые операции и сервисные возможности для эффективной обработки текстов: набор текста в интерактивном режиме, редактирование и форматирование текста, импорт/экспорт текстов из одного формата в другой, работа с разными шрифтами, со спецсимволами, с иллюстративным материалом. В пособии рассмотрен популярный текстовый процессор MS WORD, представляющий собой многофункциональную программу обработки текстов.

Другим важным на сегодняшний день приложением, рассмотренным в пособии, является MS EXCEL – табличный процессор, позволяющий выполнять такие функции, как создание и редактирование электронных таблиц и многотабличных документов (рабочих книг); оформление и печать электронных таблиц; применение средств деловой графики и решение экономических задач графическими методами и др.

Особое внимание в учебном пособии уделено языку программирования Visual Basic for Applications (VBA), встроенному в линейку продуктов Microsoft Office (включая MS WORD и MS EXCEL), а также во многие другие программные пакеты.

Данное пособие позволяет изучить теоретические основы информатики, освоить основные приемы работы в текстовом процессоре MS WORD и в табличном процессоре MS EXCEL, а также научиться программированию на Visual Basic for Applications.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Алексеева, И. В. Сборник задач и упражнений по курсу «Информатика» / И. В. Алексеева. – Обнинск: Обнин. ин-т атом. энергетики, 2007.
2. Власов, В. К. Элементы информатики / В. К. Власов, Л. Н. Королев; под ред. Л. Н. Королева. – М.: Наука, 2008.
3. Информатика / под ред. Н. В. Макаровой. – М.: Финансы и статистика, 2007. – 768 с.
4. Информатика: учеб. для вузов / под ред. С. В. Симоновича. – СПб.: Питер, 2008.
5. Кураков, Л. П. Информатика / Л. П. Кураков, Е. К. Лебедев. – М.: Вуз и школа, 2009. – 636 с.
6. Могилев, А. В. Информатика: учеб. пособие для вузов / А. В. Могилев, Н. И. Пак, Е. К. Хеннер; под ред. Е. К. Хеннера. – М.: Издат. центр «Академия», 2008.
7. Острейковский, В. А. Информатика / В. А. Острейковский. – М.: Высшая школа, 2007. – 512 с.
8. Першиков, В. И. Толковый словарь по информатике / В. И. Першиков, В. М. Савинков; 2-е изд. доп. – М.: Финансы и статистика, 2008.
9. Фигурнов, В. Э. IBM PC для пользователей / В. Э. Фигурнов. – М., 2007.
10. Якубайтис, Э. А. Информационные сети и системы: справоч. кн. / Э. А. Якубайтис. – М.: Финансы и статистика, 2008.
11. www.intuit.ru
12. www.makarova.piter.ru

Задания для Microsoft Word

Задание 1

1. Установите следующие параметры страницы: ориентация Альбомная, верхнее поле 1 см, нижнее – 4 см, левое – 1см, правое – 10 см.
2. Создайте автозамену для фразы Microsoft Word .
3. Установите автоматическую расстановку переносов.
4. Наберите предложенный текст.

Microsoft Word

Microsoft Word на глазах завоёвывает всемирную популярность. Конечно, можно использовать Microsoft Word только для написания писем, предложений и заметок, но с помощью того же Microsoft Word вы сможете создать отчёт, брошюру, газету или даже Web-страницу, на которую легко поместить данные электронных таблиц, диаграммы из Excel, слайды из Power Point и адреса из Access.

Если вы не любите печатать, не любите писать или провалили экзамен по русскому языку ещё в школе, вы, наверняка, обрадуетесь возможности с помощью Microsoft Word превратить ваш ПК в своего личного секретаря.

С помощью таких средств Microsoft Word, как проверка орфографии, проверка грамматики и словарь синонимов (Тезаурус), вы сможете выразить свои непричёсанные мысли в виде убедительных слов и предложений, которые поймёт даже ваш преподаватель. Microsoft Word – это интересно! Изучайте Microsoft Word!

5. Исправьте все ошибки (орфографические, синтаксические, лексические).
6. Добавьте в словарь незнакомые слова (если такие будут).
7. Подберите синонимы для выделенных слов.
8. Перед вторым и третьим абзацами вставьте символ «☺».
9. Сделайте сноски – расшифровки для слов «ПК» и «Microsoft Word».
10. Замените выражение Microsoft Word на MS Word.

Задание 2

1. Используя вставку символов, создайте следующую запись:

☎ номер моего телефона
 📱 номер моего сотового
 ✉ пишите мне по адресу.....
 ☺ всем привет!!!

2. Создайте нумерованный список (*Формат* → *Список*) из пяти пунктов следующего вида:

- a)
- b)

3. Создайте маркированные списки из пяти пунктов следующего вида:

- У
- Λ

4. Создайте любой многоуровневый список.

Используя режим табуляции¹, создайте оглавление с заполнителем (.....) следующего вида:

1. Введение.....	стр. 2
2. Программное обеспечение	стр. 3
2.1. Системные программы	стр. 4
2.2. Прикладные программы	стр. 6
2.2.1. Пакет MS Office.....	стр. 7
2.2.1.1. Microsoft Word	стр. 10
2.2.1.2. Microsoft Excel	стр. 13
2.2.1.3. Microsoft Access	стр. 16
2.3. Языки программирования	стр. 20
3. Заключение	стр. 25

Включите режим отображения непечатных символов ¶. Покажите, какие непечатные символы в этом режиме используются для отображения клавиш *Пробел*, *Enter*, *Tab*.

¹ При необходимости воспользуйтесь справкой. Введите искомые слова «*Табуляция*», выберите раздел «Использование позиций табуляции», нажмите кнопку *Показать*. Изучите разделы: «Установка позиций табуляции», «Удаление или перемещение позиций табуляции», «Табуляция с заполнением»

Задание 3

1. Наберите предложенный текст.
2. Примените форматирование к отдельным символам и абзацам (*Формат → Шрифт* и *Формат → Абзац*).
 - 2.1. Заголовок:
 - 2.1.1. **П** (Arial, 48 пт (пунктов), полужирный, чёрный цвет, контур с тенью);
 - 2.1.2. **A** (Arial, 14 пт, курсив, полужирный);
 - 2.1.3. **P** (Courier New, 48 пт, полужирный);
 - 2.1.4. **O** (Arial, 24 пт, белый цвет, приподнятый);
 - 2.1.5. **L** (Times New Roman, 36 пт, полужирный);
 - 2.1.6. **И** (Arial, 36 пт, полужирный, верхний индекс, выделение серым цветом).
 - 2.2. Межсимвольный интервал заголовка – разреженный на 12 пт.
 - 2.3. Интервалы между абзацами – 6 пт.
 - 2.4. 1-й абзац – использована *Буквица*.
 - 2.5. 2-й абзац – межстрочный интервал двойной с отступом первой строки.
 - 2.6. 3 – 6-й абзацы – использована маркировка каждого абзаца символом Y, шрифт – Arial, 12 пт.
 - 2.7. 7-й абзац – выступ первой строки и граница для всего абзаца.
 - 2.8. 8-й абзац – выравнивание по правому краю.
3. Организуйте сноску-объяснение для слова «взломан».
4. Отдельно раскопируйте фразу «Не используйте пароли короче 6-8 символов», используя шрифты Arial, Times New Roman, Courier New, Comic Sans. Запомните их особенности.
5. Используя верхний или нижний индекс, наберите следующее:
 - 5.1. H_2O , H_2SO_4 , $8^2+3^{16}=\text{---}$
 - 5.2. Пиши ^{в вышестоящие инстанции} МЫ _{нижеподписавшиеся}
6. Вставьте перед текстом вверху справа дату и время. Произведите нумерацию страниц внизу по центру.
7. Добавьте на страницу рамку (*Формат → Границы и Заливка → Страница → Рисунок*).

П а р о л и

Парольная защита информации является наиболее распространённой. Вы с ней сталкиваетесь везде: входя в сеть, проверяя почту, заглядывая на закрытый сайт... Но есть у паролей и одно неприятное свойство: их . А человек, наоборот, любит забывать, причём, как правило, в самый неподходящий момент.

Поэтому пароли часто записывают на бумажку или используют один и тот же «любимый» пароль всюду, что в значительной степени повышает риск того, что он будет *«взломан»*. Осмелимся дать несколько советов, которые, возможно, смогут облегчить жизнь не только вам, но и системному администратору:

У Постарайтесь избегать таких «сложных» паролей, как ваше имя, имя вашей собаки, день рождения и т. п.

У Не стоит в качестве пароля использовать осмысленные слова – при подборе пароля по словарю такой пароль «ломается» очень быстро.

У Если уж вы записали пароль на бумажку (чего делать не рекомендует-ся), то хотя бы запишите его в обратном порядке.

У Не используйте пароли короче 6–8 символов.

Ну а если всё-таки вы собрались придумать пароль самостоятельно, то выберите какое-нибудь запоминающееся предложение и образуйте пароль из него, например «Круглосуточное бесплатное кафе для студентов» → и пароль «Крубекадляст»

УДАЧИ ВАМ! Mr. New Холмс ☺

Задание 4

1. Наберите предложенные тексты.
2. Добавьте границы к абзацам.
3. Отформатируйте абзацы, используя *Форматную кисть*.

ВНИМАНИЕ, РУКОВОДИТЕЛИ!

**Хотите повысить трудоспособность Вашего коллектива?
Санаторий «Жемчужный» круглый год приглашает
на отдых и лечение**

СОЛНЦЕ...

Климатический, бальнеологический и грязевой санаторий «Жемчужный» расположен у основания Бердянской косы – целебного места Северного Приазовья.

ВОЗДУХ...

Умеренно-континентальный, приморско-степной климат курорта обладает лечебными свойствами. В воздухе Приазовья в 1,5 раза больше кислорода, чем, например, на побережье Чёрного моря.

ВОДА...

Минеральная вода наружного и внутреннего применения стимулирует функции различных органов и систем, оказывает противовоспалительное и болеутоляющее действие. Вода «Бердянская» стимулирует секреторный аппарат пищеварительного тракта, обладает противовоспалительным действием.

ЛЕЧЕНИЕ...

В санатории «Жемчужный» благодаря уникальным климатическим условиям, эффективному лечению поддаются заболевания нервной системы, органов опорно-двигательного аппарата и др.

НАШ АДРЕС:

Запорожская обл., г. Бердянск,
ул. Курортная, 79,
санаторий «Жемчужный».
Контактные телефоны:
(061-53) 4-19-68, 4-09-42

Добро
пожаловать





САДАМ ЦВЕСТИ



Чёрная смородина

Чёрная смородина является самой популярной культурой в садах Сибири. Она ценится садоводами Сибири за высокое содержание в ягодах витамина С, а также А, В, В₂, В₆ и т. д. Чёрную смородину принято считать технической культурой.

Из её ягод готовят высококачественные варенья, джемы, мармелады, соки, компоты, прохладительные напитки. Как лечебный продукт питания ягоды чёрной смородины используются в качестве профилактического средства против сердечно-сосудистых, инфекционных и лучевых заболеваний.

Основные сорта чёрной смородины Красноярского края: *Дружная, Зоя, Синяя, Бия, Память Шукшина.*

Крыжовник

Крыжовник нередко называют «северным виноградом». И, действительно, по содержанию в ягодах сахара некоторые сорта крыжовника не уступают винограду.

Крыжовник является культурой умеренного климата. Для него предпочтительны хорошо освещённые и защищённые от ветров участки в верхней или средней части склона.

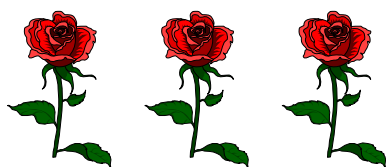
В итоге многолетней работы в Красноярской и Минусинской опытных станциях садоводства отобраны лучшие сорта для края: *Муромец, Челябинский Зелёный, Красный Крупный.*

Малина

Ягоды малины издавна являются лакомством у населения. Из них готовят соки, компоты, вина, настойки, используют для сушки, замораживания.

Ценится малина и как лекарственное растение. Отвар из сушеных ягод используют в народной медицине при лечении ревматизма, малокровия, кори, экзем.

В Красноярском крае районированы сорта: *Новость Кузьмина, Вислуха, Ласка, Полянка.*



САДАМ ЦВЕСТИ



ЧЁРНАЯ СМОРОДИНА

Чёрная смородина является самой популярной культурой в садах Сибири. Она ценится садоводами Сибири за высокое содержание в ягодах витамина С, а также А, В, В₂, В₆ и т.д. Чёрную смородину принято считать технической культурой. Из её ягод готовят высококачественные варенья, джемы, мармелады, соки, компоты, прохладительные напитки. Как лечебный продукт питания ягоды чёрной смородины используются в качестве профилактического средства против сердечно-сосудистых, инфекционных и лучевых заболеваний. Основные сорта чёрной смородины Красноярского края: *Дружная, Зоя, Синяя, Бия, Память Шукина*.

КРЫЖОВНИК

Крыжовник нередко называют «северным виноградом». И, действительно, по содержанию в ягодах сахара некоторые сорта крыжовника не уступают винограду. Крыжовник является культурой умеренного климата. Для него предпочтительны хорошо освещённые и защищённые от ветров участки в верхней или средней части склона. В итоге многолетней работы в Красноярской и Минусинской опытных станциях садоводства отобраны лучшие сорта для края: *Муромец, Челябинский Зелёный, Красный Крупный*.

МАЛИНА

Ягоды малины издавна являются лакомством у населения. Из них готовят соки, компоты, вина, настойки, используют для сушки, замораживания. Ценится малина и как лекарственное растение. Отвар из сушеных ягод используют в народной медицине при лечении ревматизма, малокровия, кори, экзем. В Красноярском крае районированы сорта: *Новость Кузьмина, Вислуха, Ласка, Полянка*.

Задание 5

1. Наберите текст.
2. Отформатируйте его по следующим параметрам:
 - Установите шрифт Bookman Old Style, 14 пт, курсив, синего цвета, межсимвольный интервал разреженный на 2 пт.
 - Абзацы – отступ справа и слева 0,5 см, отступ первой строки 1,5, выравнивание по ширине.
 - Для первого абзаца используйте *Буквицу* (*Формат* → *Буквица*).
3. Добавьте две *Сноски* (*Вставка* → *Ссылка* → *Сноска*)
 - Заповедник «Столбы» расположен между 55°38 – 55°58 северной широты и 92°20 и 93°20 восточной долготы.
 - Сборник «Енисей в стихах и легендах» (из материалов собирательницы сибирского фольклора М. В. Красножёновой). Красноярск, 1940.
4. Перед сносками вставьте символ ☞
5. Добавьте *Колонтитутлы* (*Вид* → *Колонтитутлы*):
 - верхние – Красноярские «Столбы»;
 - нижние – «Край причудливых скал».
6. В нижний правый угол вставьте номер страницы

Было это в глубокой древности. Жил в Сибири своенравный могучий царь Енисей. На гордо поднятой голове носил он прекрасную ледовую корону Саян.

Много дочерей имел Енисей, но самыми прекрасными были Базаиха и Лалетина.

Однажды приехал к царю со своей свитой богатырь, князь Такмак, сватать Лалетину. А Енисей хотел выдать Базаиху – старшей дочерью она была. Но наотрез отказался князь Такмак от такой невесты – слишком уж сварливой и капризной слыла она.

Рассердился тогда Енисей и, поднявшись во весь свой богатырский рост, сказал: «Коль так, – быть тебе, князь Такмак, и всем твоим богатырям каменными столбами. А своих дочерей я сделаю речками, и будете вы стоять подле них веки вечные».

Сказал – и так свершилось. Но слишком высоко к солнцу поднял свою гордую голову царь. Растаял от солнца его ледяной шлем, и сам Енисей превратился в могучую реку.

Так о происхождении красноярских «Столбов» рассказывается в народной легенде, обработанной известным красноярским писателем Николаем Станиславовичем Устиновичем.

Задание 6

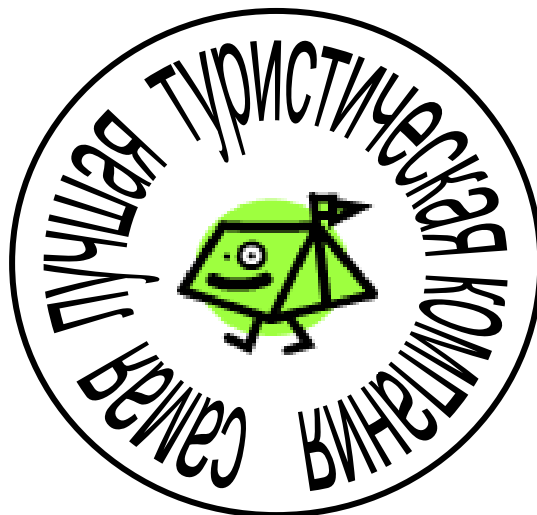
1. Используя вставку объектов WordArt и автофигур, создайте объявление.
2. Для фона используйте заливку.
3. Выделите все объекты рисунка и сгруппируйте их в один.



Задание 7

1. С помощью панели WordArt создайте эмблему с круговой надписью. Для этого создайте отдельно надписи «Торговая компания» и «ВОКРУГ СВЕТА». Используя кнопку *Форма WordArt* этой панели, для первой надписи выберите команду *Дуга вверх*, для второй надписи – *Дуга вниз*.

2. Самостоятельно попробуйте создать эмблему с надписью «Самая лучшая туристическая компания».



Задание 8

1. Создайте документ, используя нижеприведённый образец.
2. Добавьте необходимые поля форм.
3. При сохранении документа используйте команду *Сохранить как*, дайте имя файлу «Успеваемость», в поле *Тип файла* выберите *Шаблон документа*, в адресной строке укажите личную папку.
4. Закройте шаблон.
5. Создайте новый документ на основе вашего шаблона.
6. Заполните ведомость, проставив данные студента и оценки.

Текущие оценки на
студент _____ курс _____ группа _____


Предмет	Оценка
Английский язык	
История	
Экономическая теория	
Информатика	
Основы высшей математики	
История экономических учений	
Основы менеджмента	

Подпись методиста _____

М.п.

Задание 9

1. Создайте шаблон бланка-заказа пиццы по предложенному образцу.
2. Вставьте необходимые поля форм:
поле со списком сформируйте из 3–4 наименований;
поле DATE (текущая дата) для даты поступления заказа;
поле TIME (текущее время) для времени поступления заказа.
3. Обеспечьте защиту формы.
4. Сохраните файл под названием «Пиццерия ФИО» в своей папке.
5. Проверьте действие шаблона.

	Пиццерия НЯМ – НЯМ ул. Вавилова, дом 54, офис. 6 тел. 33-24-55
	ДОСТАВКУ ПИЦЦЫ НА ДОМ /поле TIME / Дата поступления заказа /Поле DATE/

Имя	/Текстовое поле/
Адрес	/Текстовое поле/
Телефон	/Текстовое поле/

	выбор	ассортимент
пицца	<input type="checkbox"/>	/Поле со списком/
напиток	<input type="checkbox"/>	/Поле со списком/
гамбургер	<input type="checkbox"/>	/Поле со списком/

<input type="checkbox"/>	предоплата	<input type="checkbox"/>	наличный расчёт
--------------------------	------------	--------------------------	-----------------

Задание 10

1. Используя меню *Таблица*, создайте таблицу и заполните её.

Страна	Столица	Достопримечательности	
		Архитектурные	Великие люди

Задание 11

1. Нарисуйте листок объявления в табличной форме с отрывными листочками для телефона следующего содержания:

Детский центр «Вундеркинд» приглашает на работу воспитателей.
Адрес: Красноярск, ул. Мира 6, офис 231.

☎ 22-11-33. Ждём Вас

2. Добавьте в объявление подходящую эмблему.

			Детский центр "ВУНДЕРКИНД"							
			приглашает на работу воспитателей							
			Адрес: Красноярск, ул. Мира, 6, офис 231. ☎ 22-11-33 ☺ Ждём Вас							
22-11-33	22-11-33	22-11-33	22-11-33	22-11-33	22-11-33	22-11-33	22-11-33	22-11-33	22-11-33	22-11-33

Задание 12

1. Вставьте на страницу текущую дату и время. Создайте верхние и нижние колонтитулы: Сибирский федеральный университет и Кафедра Бизнес-информатика (Arial; 10 пт; курсив).

2. Попробуйте создать вот такую таблицу (3 x 3):

Здесь мы использовали обычный шрифт Comic Sans 11 пт.	Шрифт Arial с выравниванием по центру, 12 пт ☺	А МОЖНО И ТАК ОФОР- МИТЬ
А здесь немного другой шрифт, Courier New, с выравниванием по левому краю	Тут попытались ввести много-много текста. Так, что читать его очень трудно, но всё-таки возможно (8 пт) И даже поместить два абзаца в одну ячейку.	А <u>здесь</u> <u>вовсе</u> безобразие !?
Вы читали когда-нибудь перпендикулярные тексты?	Если Вам уже плохо от этих таблиц, звоните 911	У Работы было много ☹ У Но я сделал это ☺!!! У Я многое могу! У Главное, не останавливаться.

Задание 13

1. Для создания таблицы сначала сделайте основу из 8 столбцов и 5 строк. А потом смело объединяйте, передвигайте, стирайте или дорисовывайте, выравнивайте по высоте, добавляйте границы и заливку. Выделив таблицу, скройте сетку (*Таблица* → *Скрыть сетку*). Объясните, в чём разница этого режима отображения таблицы с режимом *Отобразить сетку*.

Вот такая получилась таблица							
Один		Два	Три		Четыре		
1	2	3	4	5	6	7	8
					УРА!!		

Задания для Microsoft Excel**Задание 1**

В ячейку A1 запишите число 5267,468.

Раскопируйте его в диапазоне A2:B6.

Используя формат ячеек, приведите числа к следующему виду:

5267,47	5,27E+03
02.06.14. 11:13	526746,80 %
5267,5	00000-5267
11:13:55 AM	-5267
52,67 1/2	005267

Задание 2

В ячейку A1 запишите текст СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ.

Скопируйте его в ячейки A2, B3, C4, D5.

Используя формат ячеек и изменение высоты строк и ширины столбцов, приведите текст к следующему виду:

	A	B	C	D
1				
2	<i>СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ</i>			
3		<i>СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ</i>		
4			<i>СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ</i>	
5				<i>СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ</i>

Задание 3

В диапазон ячеек *A1:A5* запишите следующие формулы:

$$=5*8+54/2-10$$

$$=5*(8+54)/2-10$$

$$=(5*8+54)/2-10$$

$$=5*(8+54/2)-10$$

$$=5*(8+54/2-10)$$

Обратите внимание на результат. Объясните, почему так получилось.

В ячейку B1 запишите число 50.

В ячейку B2 запишите число 10.

В ячейку B3 запишите формулу = B1+B2.

В ячейку C2 запишите следующее выражение:

$$\frac{5}{2 \cdot 3} + \frac{20 - 3^2 + 4^3}{3 \cdot 4}.$$

В ячейку D1 поставьте число 20, в D2 – 8, в D3 – 4.

В D4 запишите следующее выражение:

$$D1 + 5 + D2 \cdot 4 - \frac{10\%D1 + 8}{D3}.$$

Задание 4

Создайте таблицу по следующему образцу:



Марка авто	Объем двигателя	Кол-во лошадиных сил	Стоимость в час	Стоимость в сутки	Стоимость в неделю
Mazda	2,0	95			
Opel	3,5	300			
Toyota	2,0	100			
Mersedes	4,0	350			
Nissan	1,6	75			
Volvo	3,5	250			

Рассчитайте стоимость часа, суток и недели аренды авто, если известно следующее: стоимость часа вычисляется по формуле

$$\frac{\text{Объем двигателя} \times \text{количество лошадиных сил}}{5},$$

стоимость аренды за сутки снижается на 15 %, а за неделю на 25 %.
Внимание! Расчет стоимости недели производится от стоимости часа.

Задание 5

Было задумано положительное четное число, которое утроили, разделили пополам и еще раз утроили. В результате вычислений получилось 27.

С помощью построения формулы определите, какое число было задумано, создав следующую таблицу:

Введите результат вычисления	Было задумано число

Задание 6

Немецкий физик Г. Фаренгейт в 1724 году предложил температурную шкалу, названную его именем. Температура по шкале Фаренгейта связана с температурой по шкале Цельсия соотношением $t_C^{\circ} = 5/9(t_F^{\circ} - 32)$.

Составьте таблицу, переводящую температуру воздуха, измеренную по шкале Цельсия, в температуру по шкале Фаренгейта.

t_C°	t_F°
10	
23	
-35	
-14	
0	
55	
12	
7	

Задание 7

Составьте формулы для нахождения процентов от общего и внесите изменения в незаполненные ячейки таблицы.

Поверхность земного шара	Северное полушарие		Южное полушарие		Земля в целом	
	в млн кв. км	в %	в млн кв. км	в %	в млн кв. км	в %
Суша	100,41		48,43			
Вода	154,64		206,62			
Всего						

Задание 8

Рассчитайте курс доллара на второй, третьей и четвертой неделе, если известно, что на второй неделе он на 2 % больше, чем на первой; курс доллара на третьей неделе уменьшился на один рубль по сравнению со второй неделей, а на четвертой неделе стал в два раза меньше, чем на третьей неделе. В зависимости от рассчитанного курса доллара определите стоимость товаров в рублях на каждой неделе.

Название товара	Стоимость \$	1-я неделя	2-я неделя	3-я неделя	4-я неделя
Товар 1	\$586				
Товар 2	\$890				
Товар 3	\$460				
Товар 4	\$1 090				
Товар 5	\$640				
Товар 6	\$550				

	1-я неделя	2-я неделя	3-я неделя	4-я неделя
Курс \$	29,00 руб.			

Задание 9

Рассчитайте заработную плату для сотрудников, если известно, что:
начисление = ставка/количество рабочих дней × количество отработанных дней;

профсоюзный налог = начисление × 1 %;

пенсионный = начисление × 1 %;

подоходный налог = (начисление – минимальная заработная плата) × 12 %;

сумма к выдаче = разность между начислением и суммой налогов с авансом.

Внесите произвольную сумму аванса каждому работнику.

	Кол-во рабочих дней в месяце	21		Мини-мальная з/п	100 руб.				
№ п/п	ФИО	Кол-во отработанных дней	Ставка	Начисление	Налоги			Аванс	Сумма к выдаче
					профсоюзный	пенсионный	подоходный		
1	Иванов А.Ф.	21	2 500 руб.						
2	Иванова Е.П.	21	6 000 руб.						
3	Китова В.К.	16	1 500 руб.						
4	Котов И.П.	5	980 руб.						
5	Круглова А.Д.	15	5 250 руб.						
6	Леонов И.И.	21	1 500 руб.						
7	Петров М.В.	3	3 000 руб.						
8	Сидоров И.А.	21	1 500 руб.						
9	Симонов К.Е.	8	2 900 руб.						
10	Храмов А.К.	21	7 000 руб.						
11	Чудов Е.Н.	21	4 650 руб.						
12	Яблоков Е.Г.	19	3 900 руб.						

Задание 10

Создайте следующую таблицу:

Стоимость номеров в гостинице «Турист»

Количество дней	Номер			
	люкс	1-местный	2-местный	3-местный
5				
7				
10				
14				

Питание (1 раз)	\$10
Проживание в сутки	\$97

Присвойте имена ячейкам, содержащим информацию о стоимости питания и проживания.

Рассчитайте стоимость проживания одного человека в номерах гостиницы, если известно:

- стоимость каждого складывается из стоимости трехразового питания и проживания;
- проживание в номере люкс на 5 % дороже;
- проживание в одноместном номере на 2 % дороже;
- в двухместных и трехместных номерах проживание делится на количество человек в номере.

Задание 11

Создайте таблицу по образцу:



№ п/п	Наименование товара	Партия	Оптовая цена за 1 шт.	Цена в розницу за 1 шт.	Доход	Общий доход
1	Матрешка	350	150,00 руб.			
2	Олимпийский мишка	273	20,50 руб.			
3	Самовар	176	1070,00 руб.			
4	Короб из бересты	320	48,20 руб.			
5	Резная шкатулка	300	120,00 руб.			
6	Соломенная корзинка	570	37,80 руб.			
7	Шкатулка из мельхиора	94	350,00 руб.			
8	Колокольчик	157	57,00 руб.			
9	Ручка деревянная, резная	1 020	49,90 руб.			
10	Картина «Русская зима»	7	5 560,00 руб.			

Используя массивы, рассчитайте розничную цену сувениров, если известно, что она выше оптовой на 17 %. Исходя из полученных данных, определите доход фирмы от 1-го изделия, а потом общий доход от партии сувениров.

ВНИМАНИЕ!!

Для того, чтобы рассчитать массив:

- выделите диапазон, в который должны быть помещены результаты вычисления;
- в первую ячейку запишите формулу с использованием диапазонов, необходимых при расчетах;
- нажмите комбинацию клавиш SHIFT+CTRL+ENTER.

Задание 12

Создайте таблицу по образцу:

Универмаг «Солнышко»				<i>Приход</i>		Расход		Оста-ток	
№ п/п	Отдел	Наимено-вание товара	Единицы измерения	Цена прихода, руб.	Кол-во прихода	Цена расхода	Кол-во расхода	Кол-во остатка	Сумма остатка
1	Кондитерский	Зефир в шоколаде	Упаковка	23,00	15		15		
2	Молочный	Молоко	Упаковка	6,90	32		28		
3	Мясной	Колбаса «Докторская»	1 кг	58,00	40		26		
4	Мясной	Сосиски	1 кг	36,00	12		10		
5	Кондитерский	Шоколад «Вдохновение»	1 шт.	12,00	50		35		
6	Вино-водочный	Пепси-кола	Бутылка 0,33л	6,70	35		13		
7	Кондитерский	Набор шоколадных конфет	Упаковка	45,00	20		14		
8	Кондитерский	Чупа-чупс	1 шт.	1,30	60		45		
9	Хлебобулочный	Хлеб турецкий	1 шт.	4,00	12		10		
10	Хлебобулочный	Хлеб ржаной	1 шт.	2,30	15		12		
11	Кондитерский	Халва	Упаковка	16,00	20		12		

Рассчитайте цену расхода, количество остатка и сумму остатка по следующим формулам:

- *цена расхода = цена прихода + 12 % накрутка;*
- *количество остатка = количество прихода – количество расхода;*
- *сумма остатка = цена прихода × количество остатка.*

Отсортируйте перечень по ключу «Отдел», затем по ключу «Наименование товара».

Измените внешний вид таблицы, оставив в ней графы «Отдел», «Наименование товара», «Единицы измерения», «Количество остатка», «Сумма остатка».

Над списком поставьте функцию текущей даты.

Задание 13

Создайте нижеприведенные таблицы по образцу.

Рассчитайте стоимость тура в рублях.

С помощью функции СУММЕСЛИ определите, сколько денег в \$ и рублях потратил каждый клиент на командировки.

Командировочные

Курс \$	30 руб.
---------	---------

Фамилия	Дата поездки	Страна	Оплата в \$	Оплата в руб.
Сидоров	Декабрь 2001 г.	Греция	850	
Петров	Январь 2002 г.	Франция	1 500	
Васечкин	Март 2002 г.	Египет	970	
Иванов	Апрель 2002 г.	Болгария	670	
Сидоров	Апрель 2002 г.	Россия	900	
Мышкин	Декабрь 2002 г.	Турция	780	
Сомов	Июль 2002 г.	Германия	1 500	
Васечкин	Июль 2002 г.	Болгария	700	
Сидоров	Сентябрь 2002 г.	Россия	980	
Петров	Декабрь 2002 г.	Египет	680	
Васечкин	Август 2002 г.	Турция	750	
Иванов	Декабрь 2002 г.	Греция	660	
Сомов	Декабрь 2002 г.	Болгария	1 200	
Мышкин	Февраль 2003 г.	Россия	400	

Фамилия	Денег потрачено	
	в \$	в рублях
Петров		
Сидоров		
Васечкин		
Иванов		
Мышкин		
Сомов		



Задание 14

Туристической фирме «Заплати и кати» поступил заказ на разработку проекта нового туристического маршрута, подготовить который нужно в определенный срок.

Вашей задачей является:

- оформить таблицу по образцу;
- вместо обозначения года XX поставить текущий год;
- с помощью функции ДНЕЙ360 определить количество дней отведенных на весь проект;
- определить количество дней отведенных на каждый этап проекта.

График проведения работ

Начало работ
22.02.XX



Окончание работ	Количество дней
06.07.XX	

	Этап 1	Этап 2	Этап 3	Этап 4
Дата начала	22.02.XX	03.03.XX	16.05.XX	11.06.XX
Кол-во дней				

Задание 15

Используя функцию ДЕНЬНЕД, определите, на какой день недели выпадают предложенные вам праздничные дни.

Вместо обозначения года XX поставьте следующий год.

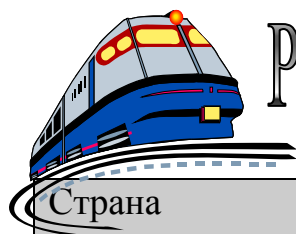
Праздничные дни



Название	Дата	№ дня недели
23 февраля 20XX г.		
8 марта 20XX г.		
1 мая 20XX г.		
9 мая 20XX г.		
1 января 20XX г.		
Мой день рождения в 20XX г.		

Задание 16

Создайте таблицу по образцу, вставляя вместо обозначения года XX текущий год:



Расписание поездок в 20XX году

Страна	Дата отправления	Дата возвращения	№ дня недели отправления	№ дня недели возвращения	Количество дней
Турция	22.03.XX	06.04.XX			
Греция	15.03.XX	22.03.XX			
США	10.09.XX	24.09.XX			
Франция	18.05.XX	19.06.XX			
ОАЭ	27.04.XX	14.05.XX			
Португалия	08.06.XX	20.06.XX			
Египет	30.08.XX	17.09.XX			
Германия	02.02.XX	16.02.XX			
Италия	11.11.XX	25.11.XX			

С помощью функций даты и времени определите номер дня недели отправления и возвращения, а также количество дней, отведенных на всю поездку.

На этом же листе постройте гистограмму, отображающую количество дней, отведенных на каждую поездку.

Задание 17

Оформите таблицу по следующему образцу:



Дни недели	Количество рейсов	1-й класс	2-й класс	Выручка за день
Понедельник	0			
Вторник	1			
Среда	3			
Четверг	1			
Пятница	0			
Суббота	5			
Воскресенье	2			
Планируемая выручка за неделю				
Максимальная выручка				
Минимальная выручка				
Среднедневная выручка				

Самолет авиакомпании «Аэрофлот» производит рейс Красноярск-Москва. Всего в самолете 1 220 посадочных мест. В 1-м классе 405 мест. Места во 2-м классе стоят 856 рублей, а в 1-м – в два раза дороже.

Рассчитайте планируемую выручку на каждый день недели и на всю неделю, а также максимальную, минимальную и среднедневную выручку.

Задание 18

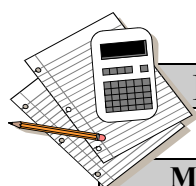
Турфирма «Заплати и кати» имеет филиалы в четырех городах России. В конце года происходит премирование филиалов по итогам продаж путевок за каждый месяц. Если сумма выручки превышает 100 000 рублей, то филиалу начисляется премия. Исходя из этого:

- при помощи функции СЧЕТЕСЛИ определите, сколько раз за весь год продажи превышали установленную сумму;
- зная количество продаж свыше определенной суммы, рассчитайте премию для каждого филиала.

Таблицу оформите по следующему образцу.

Турфирма «Заплати и кати»

Продажи филиалов за год



Премия за продажи свыше 100 000 руб.

15 000 р



Месяц	Филиал 1, руб.	Филиал 2, руб.	Филиал 3, руб.	Филиал 4, руб.
Январь	125 000.	150 000	80 000	140 000
Февраль	70 000	85 000	50 000	170 000
Март	125 500	83 000	97 000	100 000
Апрель	23 000	74 000	59 000	128 000
Май	42 000	150 000	62 000	56 500
Июнь	65 000	38 000	49 000	60 000
Июль	74 000	128 000	22 000	135 000
Август	280 000	222 000	190 000	147 000
Сентябрь	170 000	90 000	240 000	155 000
Октябрь	73 000	45 000	123 500	48 000
Ноябрь	67 000	123 000	57 000	98 000
Декабрь	155 000	321 000	180 000	189 500
Кол-во продаж				
Премия за год				

На этом же листе постройте любую диаграмму, показывающую суммы продаж по каждому месяцу.

Задание 19



Клиент	Стоимость услуг	Полученная сумма	Долг	Действия
Уилсон	\$300	\$290		
Баскервиль	\$625	\$536		
Сазерленд	\$265	\$265		
Тенер	\$422	\$421		
Крамм	\$514	\$500		

Частное детективное агентство «Холмс и Ватсон» предоставляет различные услуги населению.

Оформите таблицу по образцу.

Рассчитайте долг по каждому клиенту.

Если долг превышает \$2, то этому клиенту нужно выслать уведомление об оплате, в противном случае не высылать.

Задание 20

Филиалу фирмы «Бигус» была выделена недельная материальная помощь в размере 900 рублей. Ее нужно поделить между сотрудниками следующим образом:

- тем сотрудникам, у кого недельный заработок меньше 5 минимальных зарплат, делается доплата до 5 минимальных зарплат;
- остаток материальной помощи делится между всеми сотрудниками, пропорционально зарплате каждого.

Создайте таблицу по следующему образцу:

Распределение материальной помощи



Материальная помощь	900 руб.	Минимальная з/п		100 руб.
Ф.И.О.	Недельная зарплата	Помощь		Итого
		максимальная	фактическая	
Петров П. П.	400			
Сидорова О. О.	1 200			
Иванов И. И.	300			
Васечкин В. В.	100			
Всего				

Определите максимальную помощь и общую максимальную помощь по следующим условиям:

1. Если зарплата меньше 5 минимальных зарплат, то нужно найти разницу между 5 минимальными зарплатами и недельной зарплатой сотрудника.
2. Если зарплата больше 5 минимальных зарплат, то помощь равна 0.

Зная максимальную и общую максимальную помощь, рассчитайте фактическую помощь по следующим условиям:

1. Если общая максимальная помощь меньше выделенной материальной помощи филиалу, то помощь выдается максимальная.
2. Если общая максимальна помощь больше материальной помощи сотрудника, то помощь уменьшается и делится пропорционально планируемой максимальной помощи по такой формуле: $\text{помощь максимальная} \times \text{материальная помощь} / \text{общую максимальную помощь}$.

3. Если есть остаток, материальная помощь делится между всеми работниками пропорционально их зарплате.
4. Если общая фактическая помощь меньше выделенной материальной помощи, то остаток рассчитывается по формуле: *материальная помощь – общая фактическая помощь) / общая зарплата × недельная зарплата сотрудника + фактическая помощь*.
5. Если общая фактическая помощь больше материальной помощи, то остается фактическая помощь.

Задание 21

Создайте таблицу расчетов с клиентами за наем автомобилей (для столбцов *Взято с*, *Оплачено по* и *Дата возврата* назначьте пользовательский формат даты вида ДД.МММ. чч:мм):

Коэффициент возврата	0,5
Коэффициент доплаты	1,3

Аренда авто



Тип авто	Цена часа	Взято с	Оплачено по	Оплачено		Дата возврата	Разница		Доплата/ Возврат
				часов	сумма		часы	сумма	
Волга	110	02 нояб. 10:00	12 нояб. 00:00			18 нояб. 05:00			
БМВ	300	13 нояб. 01:30	21 нояб. 10:00			21 нояб. 10:00			
Рено	200	04 нояб. 09:06	14 нояб. 00:00			16 нояб. 08:00			
Ауди	250	01 нояб. 01:00	05 нояб. 10:00			07 нояб. 00:30			
ВАЗ	100	06 нояб. 06:30	19 нояб. 12:00			14 нояб. 00:00			

Зная цену часа, дату, время взятия авто и планируемую дату возврата, вычислите время проката в часах и его стоимость.

Зная фактическую дату возврата, найдите разницу в часах и деньгах.

Если клиент вернул машину раньше оплаченного времени, то ему возвращается некоторая сумма денег ($\text{сумма} \times \text{коэффициент возврата}$), а если позже, то с него берется недостающая сумма, увеличенная в несколько раз ($\text{сумма} \times \text{коэффициент доплаты}$) за задержку авто в прокате.

Задание 22

Создайте таблицу по образцу:

Стоимость 1 км полета		50 руб.
Питание пассажиров за каждые 1000 км		
До 1 000 км		5 руб.
До 3 000 км		10 %
Свыше 3 000 км		15 %

Стоимость авиаперевозок



Рейс	Расстояние до пунктов посадки			Длина маршрута	Стоимость питания	Стоимость полета	Доставка в аэропорт	Цена билета
	1 пункт	2 пункт	3 пункт					
СПб - Баку	1 000	900	1 179	3 079				
Москва - Казань	450	365		815				
Москва - Рим	1 500	1 636		3 136				
Москва - Цюрих	1 877	700		2 577				
Москва - Одесса	1 342			1 342				
Москва - Калининград	708	500		1 208				
Москва - Воронеж	489			489				
Уфа - Мурманск	1 100	1 000	1 319	3 419				

Средняя длина маршрута	
Всего рейсов	
Из них:	
с двумя посадками	
с одной посадкой	
беспосадочных	

Вычислите цену авиабилета в зависимости от полной протяженности маршрута до всех пунктов посадок
Цена билета состоит из трех слагаемых:

1. Стоимости питания. Пассажиров кормят каждые 1 000 км полета.
Внимание! Используйте формулу с функцией ОКРУГЛВНИЗ.
2. Стоимости перевозки пассажира. Определяется умножением длины маршрута на стоимость одного километра.

Если длина перелета менее 1 000 км, то она равна 5 руб. за каждый км.
Если длина перелета от 1 000 до 3 000 км – цена снижается на 10 %.
Если длина перелета свыше 3 000 км – цена снижается на 15 %.

3. Стоимости доставки в аэропорт. Выполняется только для тех пассажиров, которые следуют на расстояние не менее 3 000 км, в размере 100 рублей.
После определения цены авиабилетов, используя статистические функции найдите среднюю протяженность маршрутов; общее количество рейсов; количество рейсов с двумя посадками; количество рейсов с одной посадкой; количество беспосадочных рейсов.

Задание 23

С помощью функции *БС* вычислите значение суммы на расчетном счете по истечении 10 лет, если в банке было размещено 19 000 руб. под 12 % годовых при условии, что проценты начисляются каждый квартал.

С помощью функции *БС* определите будущую сумму вклада через 1,5 года, если в конце каждого месяца вкладчик вносит 980 руб. под 26 % годовых.

С помощью функции *БС* вычислите значение суммы на расчетном счете по истечении 9 лет, если в банке было размещено 30 000 руб. под 15 % годовых при условии, что проценты начисляются каждые полгода и в конце каждого периода вкладчику выплачивается 1000 руб.

С помощью функции *ПС* определите, какую сумму нужно вложить в банк под 23 % годовых сроком на 10 лет, чтобы накопить 50 000 руб.

С помощью функции *ПЛТ* определите ежемесячные и ежегодные выплаты по взятому вами кредиту в размере 100 000 руб., вносимые в течение 5 лет, при годовой ставке 22 %.

Задание 24

Оформите таблицы по следующему образцу:

Кредитная организация

«Возьми – верни сполна»

Размер ссуды	Годовой процент	Кол-во платежей	Ежемесячный платеж	Прибыль

Ваша организация оформила для турфирмы «Заплати и кати» кредит в размере 100 000 руб. под 47 % годовых сроком на 2 года.

Определите ежемесячный платеж и прибыль, полученную с этого кредита (% , начисленные за 2 года).

Определите, каким туром вы можете воспользоваться на полученную прибыль. В столбце «Результат» должно стоять «подходит» или «не подходит».

Прайс-лист «Заплати и кати»

Курс \$	
---------	--

Страна	Начало тура	Окончание тура	Кол-во дней	Стои- мость в \$	Стои- мость в руб.	Скидка	Резуль- тат
Греция	27.07.00	04.08.00		1 200			
Германия	20.05.00	03.06.00		1 500			
Болгария	01.06.00	15.06.00		900			
Чехослова- кия	08.08.00	18.08.00		400			
Италия	14.07.00	01.08.00		1 900			
США	18.05.00	10.06.00		2 800			

Определите количество дней, отведенных на каждый тур, и стоимость в рублях.

В столбце «Скидка» установите скидку 5 %, если длительность тура равна или превышает 14 дней, в противном случае скидки «нет».

Задания для Microsoft VBA

Задание 1

Создайте новую рабочую книгу и сохраните ее в своей папке под именем «Платежи».

Добавьте рабочие листы так, чтобы их было 9.

Первый лист назовите «Данные» и на этом листе создайте следующую таблицу:

<i>Страна</i>	<i>Город</i>	<i>Вид</i>	<i>Платеж</i>	<i>Дата</i>
РФ	Курск	Налично	50	5 января
Украина	Киев	Через банк	100	14 января
Беларусь	Минск	Налично	80	25 января
Беларусь	Брест	Через банк	24	9 февраля
РФ	Москва	Через банк	90	12 февраля
РФ	Курск	Налично	300	20 февраля
Украина	Киев	Через банк	120	8 марта
Украина	Киев	Через банк	150	18 марта
РФ	Москва	Налично	65	24 марта
РФ	Москва	Налично	70	26 марта
Украина	Львов	Налично	200	1 апреля
РФ	Курск	Через банк	120	6 апреля
Беларусь	Минск	Налично	60	8 апреля
Беларусь	Брест	Через банк	90	20 апреля
Беларусь	Минск	Через банк	50	21 апреля
РФ	Москва	Налично	100	25 апреля

Скопируйте эту таблицу на все листы.

Второй лист назовите *Сортировка* и проведите на нем сортировку данных по столбцу «Страна» (по возрастанию), затем по столбцу «Город» (по возрастанию) и в последнюю очередь по столбцу «Платеж» в порядке убывания.

Третий лист назовите *Автофилтр1* и проведите на нем отбор платежей, относящихся к Российской Федерации, сделанных в феврале месяце.

Четвертый лист назовите *Автофилтр2* и проведите на нем отбор платежей, превышающих 65 тыс. руб., относящихся к Москве и Минску.

Пятый лист назовите *Автофилтр3* и проведите на нем отбор наличных платежей, относящихся к апрелю и поступивших из РФ и Беларуси, превышающих 55 тыс. руб.

Шестой лист назовите *Итоги1* и подведите итоги по наличному/безналичному расчету, предварительно отсортировав данные по ключам «Страна» и «Вид платежа» по возрастанию.

Седьмой лист назовите *Итоги2* и подведите итоги по месяцам платежей, предварительно установив для столбца «Дата» пользовательский формат вида МММ.

Восьмой лист назовите *Сводная таблица1* и постройте на нем сводную таблицу, которая отображала бы суммы платежей по странам и городам. Покажите наличные/безналичные платежи, переместив кнопку *Вид* на ось страниц.

Девятый лист назовите *Сводная таблица2* и постройте на нем сводную таблицу, которая отображала бы суммы помесечных платежей по странам (предварительно в списке присвойте для столбца «Дата» пользовательский формат вида МММ).

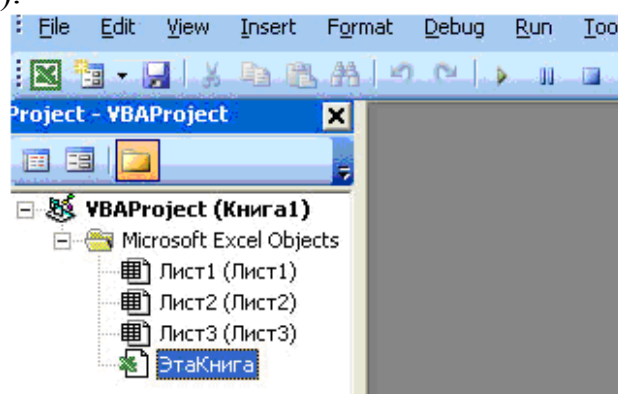
Создайте макросы перехода с листа «Данные» на все остальные листы и обратно.

Назначьте макросы на кнопки (панель инструментов «Формы» элемент «Кнопка») с соответствующими названиями.

Откройте созданные макросы в редакторе VBA и проанализируйте программный код процедуры.

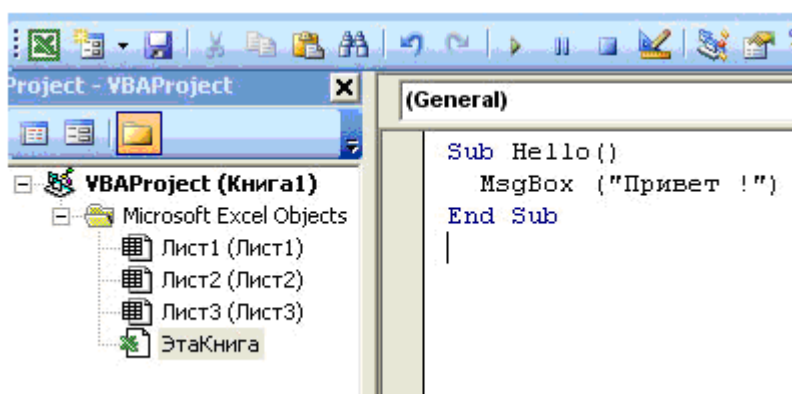
Задание 2

Запустите Excel, выберите в меню *Сервис->Макрос->Редактор Visual Basic* (Alt+F11).



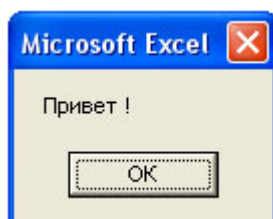
Двойным щелчком на объекте «ЭтаКнига» откройте окно редактирования макросов. Введите такой текст:

```
Sub Hello()  
MsgBox ("Привет !")  
End Sub
```



Первая строка этого макроса является заголовком процедуры, Hello – имя макроса, по которому его можно вызвать. Во второй строке происходит вызов функции вывода диалогового окна с текстом «Привет!». Третья строка – окончание процедуры.

Теперь запустите этот макрос на выполнение. Выберите команду меню *Run->Run Sub/UserForm* или нажмите F5. В результате на экране появится следующее окно:



Добавьте на Лист 1 кнопку с названием «Привет» и назначьте на нее созданный макрос.

Задание 3

Для объекта «Лист 1» запишите макрос, добавляющий в ячейку A1 число 2, в ячейку A2 число 3, а в ячейке A3 создающий формулу, суммирующую ячейки A1 и A2. Для этого в редакторе VBA дважды щелкните мышкой по объекту «Лист1» в окне «Project» и введите следующий текст:

```
Sub summA1andA2()  
' Суммирует в ячейке A3 числа, добавленные в ячейку A1 и A2  
Range("a1").Value = 2  
Range("a2").Value = 3  
Range("a3").Value = ("=a1+a2")  
End Sub
```

На листе 1 создайте кнопку с названием «Расчет» и присвойте ей созданный макрос.

На этом же листе создайте макрос, очищающий диапазон A1:A3. Для этого в редакторе VBA введите следующий текст:

```
Sub clear()  
' Удаляет значения из диапазона A1:A3  
Range("a1:a3").clear  
End Sub
```

Создайте кнопку с названием «Очистить» и присвойте ей созданный макрос.

Запишите макрос, выделяющий диапазон ячеек D5:E7 на листе 1. Для этого в редактор VBA напишите следующий текст:

```
Sub vibor()  
' Выделяет ячейки D5:E7  
Range("d5:e7").Select  
End Sub
```

Создайте кнопку с названием «Выбор» и присвойте ей созданный макрос.

Задание 4

Создайте макрос, заполняющий диапазон ячеек A1:C5 произвольным числом и подсчитывающий в ячейке C7 сумму всех ячеек. По окончании работы макроса должно появляться диалоговое окно с надписью «Расчет окончен».

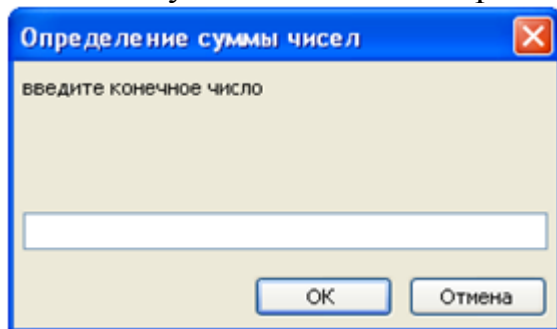
Создайте кнопку с названием «Мой макрос» и назначьте на нее созданный макрос.

Задание 5

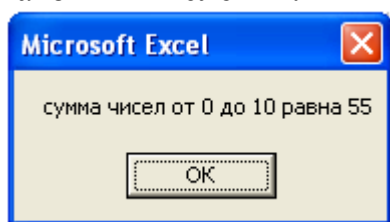
Напишите программу, суммирующую числа от 0 до указанного целого положительного числа.

Последовательность работы программы:

- Выводится диалоговое окно «InputBox» под названием «Определение суммы чисел» с предложением ввести конечное число;



- После ввода пользователем числа, с помощью цикла происходит суммирование от 0 до введенного числа;
- по окончании суммирования выводится окно «MsgBox» с результатом вычисления.



Для выполнения задания необходимо:

- объявить 3 переменные (1 для конечного числа, 2 для цикла, 3 для суммы);
- назначить переменной 1 объект «InputBox» следующим образом: `i = Application.InputBox(prompt:="введите конечное число", Title:="определение суммы чисел");`
- с помощью цикла провести суммирование чисел и присвоить результат переменной 3;

вывести окно «MsgBox» с результатом.