

Министерство образования Республики Беларусь

Учреждение образования
«Полоцкий государственный университет»

А. В. Спиридонов

ИНФОРМАТИКА

Учебно-методический комплекс
для студентов специальностей 1-70 04 02
«Теплогазоснабжение, вентиляция и охрана воздушного бассейна»,
1-70 04 03 «Водоснабжение, водоотведение и охрана водных ресурсов»

В двух частях

Часть 1

Новополоцк
ПГУ
2013

УДК 004(075.8)
ББК 32.97я73
С72

Рекомендован к изданию методической комиссией
инженерно-технологического факультета
в качестве учебно-методического комплекса (протокол № 7 от 12.04.2012)

РЕЦЕНЗЕНТЫ:

канд. техн. наук, начальник отд. подготовки кадров
з-да «Полимир» ОАО «Нафтан» г. Новополоцка А. В. ДУБРОВСКИЙ;
канд. техн. наук, доц. каф. химической технологии топлива
и углеродных материалов УО «ПГУ» А. А. ЕРМАК

Спиридонов, А. В.

С72 Информатика : учеб.-метод. комплекс для студентов специальностей
1-70 04 02 «Теплогазоснабжение, вентиляция и охрана воздушного бассейна»
и 1-70 04 03 «Водоснабжение, водоотведение и охрана водных ресурсов».
В 2-х ч. Ч. 1 / А. В. Спиридонов. – Новополоцк : ПГУ, 2013. – 280 с.
ISBN 978-985-531-424-1.

Приведены тематический план дисциплины, объем в часах лекционных и лабораторных занятий. Изложены темы изучаемого курса. Представлены вопросы к экзамену, рекомендации по организации контроля изучения дисциплины, список литературы.

Предназначен для студентов специальностей 1-70 04 02 «Теплогазоснабжение, вентиляция и охрана воздушного бассейна» и 1-70 04 03 «Водоснабжение, водоотведение и охрана водных ресурсов». Может быть полезен преподавателям и студентам вузов инженерно-технологических специальностей.

УДК 004(075.8)
ББК 32.97я73

ISBN 978-985-531-424-1 (ч. 1)
ISBN 978-985-531-423-4

© Спиридонов А. В., 2013
© УО «ПГУ», 2013

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1. ВВЕДЕНИЕ В ИНФОРМАТИКУ И ВЫЧИСЛИТЕЛЬНУЮ ТЕХНИКУ	7
1.1. Определение и основные понятия информатики. Предмет и основная задача информатики	7
1.2. Виды и свойства информации	8
1.3. Восприятие, сбор, передача, обработка и накопление информации	9
1.4. Классификация ЭВМ	11
1.5. Представление информации в ЭВМ	13
1.6. Устройство персонального компьютера (базовая конфигурация)	15
1.7. Носители информации	18
1.8. Периферийные устройства персонального компьютера	22
1.9. Программное обеспечение средств вычислительной техники	24
1.10. Назначение, классификация и основные функции операционных систем	26
1.11. Обмен данными в Windows	29
1.12. Программы-архиваторы	30
2. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ	34
2.1. Понятие и основные свойства алгоритма	34
2.2. Введение в программирование на языке Паскаль (общая структура программ, основные элементы языка)	37
2.3. Программирование линейных алгоритмов	41
2.4. Программирование разветвляющихся алгоритмов	42
2.5. Программирование алгоритмов множественного выбора	43
2.6. Программирование циклических алгоритмов	44
2.7. Структурированные типы данных. Множества	46
2.8. Структурированные типы данных – массивы. Одномерные массивы	48
2.9. Подпрограммы, пользовательские процедуры и функции	55
2.10. Работа с двумерными массивами	58
2.11. Файловый тип данных	59
2.12. Стандартный модуль Graph	61
3. КОМПЬЮТЕРНЫЕ СЕТИ	64
3.1. Назначение компьютерных сетей	64
3.2. Локальные и глобальные сети. Основные понятия	64
3.3. Краткая история развития Internet	66
3.4. Службы Internet	69
3.5. Поиск информации в Internet	75
3.6. Компьютерные вирусы и антивирусные программы	81
4. ТЕКСТОВЫЙ ПРОЦЕССОР MICROSOFT WORD	87
4.1. Технология подготовки текстового документа	87
4.2. Таблицы	100
4.3. Использование графических объектов в документах Microsoft Word	102
4.4. Печать документа	111

5. ТАБЛИЧНЫЙ ПРОЦЕССОР MICROSOFT EXCEL	113
5.1. Ввод и редактирование данных	114
5.2. Работа с книгами Microsoft Excel	119
5.3. Форматирование данных	120
5.4. Организация вычислений	123
5.5. Диаграммы	128
5.6. Управление данными	134
6. МЕТОДЫ И АЛГОРИТМЫ ЧИСЛЕННОГО РЕШЕНИЯ УРАВНЕНИЙ	140
6.1. Основы теории погрешностей	140
6.2. Приближенное решение нелинейных уравнений	140
6.3. Методы решения систем линейных уравнений	145
6.4. Приближенное вычисление определенных интегралов	148
6.5. Приближенное решение дифференциальных уравнений	150
6.6. Интерполирование функций	151
6.7. Регрессия	157
7. ВВЕДЕНИЕ В ТЕХНОЛОГИЮ БАЗ ДАННЫХ	162
7.1. Структура простейшей базы данных	162
7.2. Свойства полей базы данных (на примере СУБД MS ACCESS)	163
7.3. Типы данных	165
7.4. Объекты базы данных	165
7.5. Проектирование баз данных. Режимы работы с базами данных	168
7.6. Взаимодействие заказчика базы данных с разработчиком	168
7.7. Работа с системой управления базой данных Microsoft Access	172
8. МАТЕМАТИЧЕСКИЙ ПАКЕТ MSCAD	190
8.1. Детали интерфейса	190
8.2. Редактирование с применением клавиатуры	193
8.3. Алфавит входного языка системы Mathcad	195
8.4. Простые вычисления арифметических выражений и их редактирование	197
8.5. Работа с графикой	200
8.6. Работа с векторами и матрицами	204
8.7. Решение уравнений	206
8.8. Выполнение символьных вычислений	208
8.9. Нахождение экстремумов функций	209
8.10. Аппроксимация функций	211
8.11. Вычисление определенных интегралов	225
8.12. Решение дифференциальных уравнений	226
8.13. Решение уравнений в частных производных	236
9. ОСНОВЫ РАБОТЫ В AUTOCAD	243
9.1. Окно AutoCAD 2008	243
ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ	276
ЛИТЕРАТУРА	279

ВВЕДЕНИЕ

Целью изучения дисциплины «Информатика» является подготовка студентов к самостоятельной работе на рабочих местах, оснащенных современной компьютерной техникой и соответствующими программными продуктами.

Задачи дисциплины – формирование начальных навыков работы на персональном компьютере и овладение основными пакетами прикладных программ.

В рамках академических и профессиональных компетенций в результате изучения дисциплины «Информатика» студент должен:

знать:

- технические и программные средства компьютера;
- основы алгоритмизации инженерных задач;
- программирование на алгоритмическом языке;
- технологии применения стандартных программ для компьютерного моделирования технических задач;

уметь:

- ставить прикладные задачи, строить их математические модели, разрабатывать алгоритмы решения;
- реализовать построенный алгоритм в виде собственной программы на алгоритмическом языке или с использованием стандартных программ (MathCAD и Excel);
- использовать разработанные программные комплексы в профессиональной деятельности (при выполнении курсовых работ по учебным дисциплинам и дипломной работы по специальности).

Изучение дисциплины «Информатика» рассчитано на 290 учебных часов, из них 188 – аудиторных, примерное распределение которых по видам занятий включает 52 лекционных и 136 практических часов. Примерный тематический план дисциплины представлен в таблице.

Таблица

Наименование раздела, темы	Всего аудиторных часов	Лекции (часы)	Практические (часы)
	188	52	136
1 семестр	52	18	34
Тема 1. Введение в информатику и вычислительную технику. Техническое и программное обеспечение средств вычислительной техники. Операционная система Windows	2	2	0

Окончание табл.

Тема 2. Основы алгоритмизации и программирования	24	6	18
Тема 3. MICROSOFT OFFICE. Текстовый процессор MICROSOFT WORD	14	4	10
Тема 4. MICROSOFT POWER POINT, средства подготовки презентаций	6	2	4
Тема 5. Компьютерные сети	6	4	2
2 семестр	50	16	34
Тема 6. Табличный процессор MICROSOFT EXCEL	20	8	14
Тема 7. Методы и алгоритмы численного решения уравнений	18	4	14
Тема 8. Система управления базами данных MICROSOFT ACCESS	10	4	6
3 семестр	52	18	34
Тема 9. Математический пакет MCAD	34	12	22
Тема 10. Компьютерная графика. Графические редакторы: ACAD, Компас, Raduga и др.	18	6	12
4 семестр	34	0	34
Тема 11. Выполнение проектировочных работ в графических редакторах: ACAD, Компас и др.	34	0	34

Формы контроля знаний – опрос или результат выполнения практического задания, выполнение контрольных работ по изученной теме, сдача зачетов и экзамена, выполнение курсовой работы.

1. ВВЕДЕНИЕ В ИНФОРМАТИКУ И ВЫЧИСЛИТЕЛЬНУЮ ТЕХНИКУ

1.1. Определение и основные понятия информатики. Предмет и основная задача информатики

Информатика – это техническая наука, систематизирующая приемы создания, хранения, воспроизведения, обработки и передачи информации средствами вычислительной техники (ВТ), а также принципы функционирования этих средств и методы управления ими.

Слово информатика происходит от объединения терминов *Information* (информация) и *Automatique* (автоматика), что выражает ее суть как науки об автоматической обработке информации.

Предмет информатики составляют следующие понятия:

- аппаратное обеспечение и программное обеспечение средств ВТ;
- средства взаимодействия аппаратного и программного обеспечения;
- средства взаимодействия человека с аппаратными и программными средствами.

Методы и средства взаимодействия человека с аппаратными и программными средствами называют пользовательским *интерфейсом*. *Основной задачей* информатики является систематизация приемов и методов работы с аппаратными и программными средствами ВТ. К основным понятиям информатики относятся следующие понятия:

1. *Информация* – это отражение реального мира с помощью знаков и сигналов. В узком смысле под информацией понимают те явления, которые человек получает из окружающего мира. Понятие «информация» тесно связано с понятием «информационные системы».

2. *Информационные системы* выполняют технологические функции по сбору, накоплению, хранению и обработке информации.

3. *Информационные технологии* – целенаправленный процесс преобразования информации, использующий совокупность средств и методов сбора, обработки, хранения и передачи информации.

4. *Информационные ресурсы* – информация, используемая на производстве, в технике, управлении обществом, специально организованная и обработанная на ЭВМ.

5. *Инфосфера* – совокупное информационное пространство.

6. *Информатизация общества* – повсеместное внедрение комплекса мер, направленных на обеспечение полного и своевременного использования достоверной информации, и зависит от степени освоения и развития новых информационных технологий.

1.2. Виды и свойства информации

Все многообразие окружающей нас информации можно сгруппировать по различным признакам. По признаку «область возникновения» информация делится на:

– *элементарную* – отражает процессы и явления неодушевленной природы;

– *биологическую* – отражает процессы растительного и животного мира;

– *социальную* – отражает процессы человеческого общества.

По способу передачи и восприятия различают информацию:

– *визуальную* – передается видимыми образами и символами;

– *аудиальную* – передается звуками;

– *тактильную* – передается ощущениями;

– *органолептическую* – передается запахами и вкусом;

– *машинную* – выдаваемую и воспринимаемую средствами ВТ.

Информацию, создаваемую и используемую человеком, по общественному назначению делят на виды (рис. 1.1).



Рис. 1.1. Виды информации

В информатике рассматривают две формы представления информации:

– *аналоговая* (непрерывная) – температура тела; мелодия, извлекаемая на скрипке, когда смычок не отрывается от струн и не останавливается; движение автомобиля;

– *дискретная* (прерывистая) – времена года, точка и тире в азбуке Морзе.

Информация обладает рядом свойств:

– *адекватность* – т.е. степень соответствия информации, полученной потребителем, тому, что автор вложил в ее содержание;

– *достоверность* – соответствие информации объективной реальности (как текущей, так и прошедшей) окружающего мира;

– *полнота* – т.е. достаточность информации для принятия решения. С понятием полноты информации сталкиваются все, кому приходится выполнять служебные задания. Если исходные данные неполны, принять верное решение непросто;

– *избыточность* – это качество позволяет человеку меньше напрягать свое внимание и меньше утомляться;

– *объективность и субъективность* – понятие объективности информации является относительным. Так, например, принято считать, что в результате наблюдения фотоснимка объекта образуется более объективная информация, чем в результате наблюдения рисунка того же объекта, выполненного человеком;

– *доступность* – это мера возможности получить ту или иную информацию;

– *актуальность* – это степень соответствия информации текущему моменту времени.

1.3. Восприятие, сбор, передача, обработка и накопление информации

Восприятие информации – процесс преобразования сведений, поступающих в техническую систему или живой организм из внешнего мира, в форму, пригодную для дальнейшего использования. Благодаря восприятию информации обеспечивается связь системы с внешней средой (в качестве которой могут выступать человек, наблюдаемый объект, явление или процесс и т.д.). Восприятие информации необходимо для любой информационной системы.

Сбор информации – это процесс получения информации из внешнего мира и приведение ее к стандарту для данной информационной системы. Обмен информацией между воспринимающей ее системой и окружающей средой осуществляется посредством сигналов. *Сигнал* можно определить как средство перенесения информации в пространстве и времени. В качестве носителя сигнала могут выступать звук, свет, электрический ток, магнитное поле и т.п. Сбор информации, как правило, сопровождается ее регистрацией, т.е. фиксацией информации на материальном носителе (документе или машинном носителе).

Передача информации осуществляется различными способами: с помощью курьера, пересылка по почте, доставка транспортными средствами, дистанционная передача по каналам связи. Дистанционная передача по каналам связи сокращает время передачи данных. Для ее осуществления необходимы специальные технические средства. Некоторые технические средства сбора и регистрации, собирая автоматически информацию с датчиков, установленных на рабочих местах, передают ее в ЭВМ. Поступление информации по каналам связи в центр обработки в основном осуществляется двумя способами: на машинном носителе и непосредственно в ЭВМ при помощи специальных программных и аппаратных средств.

В современных развитых информационных системах *машинная обработка информации* предполагает последовательно-параллельное во времени решение вычислительных задач. Это возможно при наличии определенной организации вычислительного процесса. Вычислительная задача по мере необходимости обращается с запросами в вычислительную систему. Организация процесса предполагает определение последовательности решения задач и реализацию вычислений. Последовательность решения задается, исходя из их информационной взаимосвязи, когда результаты решения одной задачи используются, как исходные данные для решения другой.

Технология электронной обработки информации – человеко-машинный процесс исполнения взаимосвязанных операций, протекающих в установленной последовательности с целью преобразования исходной (первичной) информации в результатную. Операция представляет собой комплекс совершаемых технологических действий, в результате которых информация преобразуется. Технологические операции разнообразны по сложности, назначению, технике реализации, выполняются на различном оборудовании многими исполнителями.

Хранение и накопление информации вызвано многократным ее использованием, применением постоянной информации, необходимостью комплектации первичных данных до их обработки.

В современном понимании *компьютер* – это универсальное электронное устройство, предназначенное для автоматизации создания, хранения, обработки, транспортировки и воспроизведения данных.

Совокупность устройств, предназначенных для автоматической или автоматизированной обработки данных, называют *вычислительной техникой*. Конкретный набор взаимодействующих между собой устройств и программ, предназначенный для обслуживания одного рабочего участка, называют *вычислительной системой*. Центральным устройством большинства вычислительных систем является *компьютер*.

1.4. Классификация ЭВМ

1.4.1. Классификация по назначению

Большие ЭВМ. Это самые мощные компьютеры. Их применяют для обслуживания очень крупных организаций и целых отраслей народного хозяйства. За рубежом компьютеры этого класса называют *мэйнфреймами (mainframe)*. В России за ними закрепился термин *большие ЭВМ*. Штат обслуживания большой ЭВМ составляет до многих десятков человек. На базе таких суперкомпьютеров создают *вычислительные центры*, включающие в себя несколько отделов или групп.

Мини-ЭВМ. От больших ЭВМ компьютеры этой группы отличаются уменьшенными размерами и, соответственно, меньшей производительностью и стоимостью. Такие компьютеры используются крупными предприятиями, научными учреждениями, банками и некоторыми высшими учебными заведениями, сочетающими учебную деятельность с научной. Для организации работы с мини-ЭВМ тоже требуется специальный вычислительный центр, хотя и не такой многочисленный, как для больших ЭВМ.

Микро-ЭВМ. Компьютеры данного класса доступны многим предприятиям. Организации, использующие микро-ЭВМ, обычно не создают вычислительные центры. Для обслуживания такого компьютера им достаточно небольшой вычислительной лаборатории в составе нескольких человек. Несмотря на относительно невысокую производительность по сравнению с большими ЭВМ, микро-ЭВМ находят применение и в крупных вычислительных центрах. Там им поручают вспомогательные операции, для которых нет смысла использовать дорогие суперкомпьютеры.

Персональные компьютеры (ПК). Начиная с 1999 г. в области ПК начал действовать международный сертификационный стандарт – *спецификация PC99*. Он регламентирует принципы классификации ПК и оговаривает минимальные и рекомендуемые требования к каждой из категорий. Новый стандарт установил следующие категории персональных компьютеров:

- Consumer PC (массовый ПК);
- Office PC (деловой ПК);
- Mobile PC (портативный ПК);
- Workstation PC (рабочая станция);
- Entertainment PC (развлекательный ПК).

Согласно спецификации PC99 большинство ПК, присутствующих в настоящее время на рынке, попадают в категорию *массовых*. Для *деловых*

ПК минимизированы требования к средствам воспроизведения графики, а к средствам работы со звуковыми данными вообще не предъявляются. Для *портативных ПК* обязательным является наличие средств для создания соединений удаленного доступа, т.е. средств компьютерной связи. В категории *рабочих станций* повышены требования к устройствам хранения данных, а в категории *развлекательных ПК* – к средствам воспроизведения графики и звука.

1.4.2. Классификация по уровню специализации

По уровню специализации компьютеры делятся на *универсальные* и *специализированные*. На базе универсальных компьютеров можно собирать вычислительные системы произвольного состава (состав компьютерной системы называется *конфигурацией*). Так, например, один и тот же ПК можно использовать для работы с текстами, музыкой, графикой, фото- и видеоматериалами.

Специализированные компьютеры предназначены для решения конкретного круга задач. К таким компьютерам относятся, например, бортовые компьютеры автомобилей, судов, самолетов, космических аппаратов.

1.4.3. Классификация по типоразмерам

По типоразмерам ПК можно классифицировать следующим образом: *настольные (desktop)*, *портативные (notebook)*, *карманные (palmtop)*.

Настольные модели распространены наиболее широко. Они являются принадлежностью рабочего места. Эти модели отличаются простотой изменения конфигурации за счет несложного подключения дополнительных внешних устройств или установки дополнительных внутренних компонентов. Достаточные размеры корпуса в настольном исполнении позволяют выполнять большинство подобных работ без привлечения специалистов, а это позволяет настраивать компьютерную систему оптимально для решения именно тех задач, для которых она была приобретена.

Портативные модели зачастую используют бизнесмены, коммерсанты, руководители предприятий и организаций, проводящие много времени в командировках и разъездах. С портативным компьютером можно работать при отсутствии рабочего места.

Карманные модели выполняют функции «интеллектуальных записных книжек». Они позволяют хранить оперативные данные и получать к ним быстрый доступ.

1.4.4. Классификация по совместимости

Аппаратная совместимость. В области ПК сегодня наиболее широко распространены две аппаратные платформы: *IBM PC* и *Apple McIntosh*. Кроме них существуют и другие платформы, распространенность которых ограничивается отдельными регионами или отдельными отраслями. Принадлежность компьютеров к одной аппаратной платформе повышает совместимость между ними, а принадлежность к разным платформам – понижает.

Кроме аппаратной совместимости существуют и другие виды совместимости: *совместимость на уровне операционной системы (ОС), программная совместимость, совместимость на уровне данных.*

1.5. Представление информации в ЭВМ

1.5.1. Кодирование информации

Кодирование данных двоичным кодом. Своя система кодирования существует и в ВТ – она называется *двоичным кодированием* и основана на представлении данных последовательностью всего двух знаков: 0 и 1. Эти знаки называются *двоичными цифрами*, по-английски – *binary digit*, или, сокращенно, *bit* (*бит*).

Кодирование целых и действительных чисел. Для кодирования целых чисел от 0 до 255 достаточно иметь 8 разрядов двоичного кода (8 бит). Комбинация из 8 бит называется *байтом*.

$$0000\ 0000 = 0$$

$$0000\ 0001 = 1$$

.....

$$1111\ 1110 = 254$$

$$1111\ 1111 = 255$$

Шестнадцать бит позволяют закодировать целые числа от 0 до 65535, и 24 бита – уже более 16,5 миллионов разных значений. Для действительных чисел используют 80-разрядное кодирование.

1.5.2. Системы счисления

Система счисления – это совокупность цифровых знаков и правил их записи, применяемая для однозначной записи чисел. Все системы счисления подразделяются на *позиционные* и *непозиционные*.

Непозиционной называется такая система счисления, в которой значение цифры не зависит от ее положения в ряду цифр, изображающих число.

Примером является римская система счисления. Цифры в римской системе обозначаются различными знаками: 1 – I; 3 – III; 5 – V; 10 – X; 50 – L; 100 – C; 500 – D; 1000 – M. Так, XC – 90; CX – 110; MCMLXXXVIII – 1988. Выполнять арифметические действия в непозиционных системах неудобно. Поэтому в настоящее время эти системы не используются для расчетов.

Позиционной называется такая система счисления, в которой значение цифры зависит от ее положения в ряду цифр, изображающих число, т.е. *веса*. В десятичной системе счисления вес каждой последующей цифры в 10 раз больше веса предыдущей. Например, цифра 2 в числе 1235 имеет значение 200, т.к. она расположена в третьей справа позиции числа.

В ЭВМ применяют позиционную систему счисления с недесятичным основанием: двоичную, восьмеричную, шестнадцатеричную и др. В таблице 1.1 показано соответствие записи чисел в десятичной системе, двоичной, восьмеричной и шестнадцатеричной .

Таблица 1.1

Соответствие записи чисел в различных системах счисления

X(10)	X(2)	X(8)	X(16)
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Пример 1. Число 321_{10} запишем в двоичной системе счисления. Для этого необходимо разложить число в виде суммы по степеням 2:

$$321_{10} = 1 \cdot 2^8 + 1 \cdot 2^6 + 1 \cdot 2^0$$

Пример 2. Число 10100101_2 перевести в десятичную систему счисления:

$$10100101_2 = 1 \cdot 2^0 + 1 \cdot 2^2 + 1 \cdot 2^5 + 1 \cdot 2^7 = 165_{10}.$$

1.5.3. Единицы измерения информации

Наименьшей единицей измерения информации является *байт*. Поскольку одним байтом, как правило, кодируется один символ текстовой информации. Более крупная единица измерения – *килобайт* (кбайт). 1 кбайт = 2^{10} байт = 1024 байт. В килобайтах измеряют сравнительно небольшие объемы данных.

Более крупные единицы:

1 Мбайт (Мегабайт) = 1024 Кбайт = 2^{20} байт;

1 Гбайт (Гигабайт) = 1024 Мбайт = 2^{30} байт;

1 Тбайт (Терабайт) = 1024 Гбайт = 2^{40} байт.

1.6. Устройство персонального компьютера (базовая конфигурация)

Персональный компьютер – универсальная техническая система. Его *конфигурацию* (состав оборудования) можно гибко изменять по мере необходимости. Существует также понятие *базовой конфигурации*. В таком комплекте компьютер обычно поставляется. Понятие базовой конфигурации может меняться. В настоящее время в базовой конфигурации рассматривают четыре устройства: системный блок, монитор, клавиатура, манипулятор «мышь».

1.6.1. Системный блок

Системный блок – это самая главная часть компьютера, внутри которой установлены наиболее важные компоненты. Устройства, находящиеся внутри системного блока, называют *внутренними*, а устройства, подключаемые к нему снаружи, – *внешними*. Внешние дополнительные устройства, предназначенные для ввода, вывода и длительного хранения данных, называют *периферийными*. По внешнему виду системные блоки различаются формой корпуса. Корпуса ПК выпускают в горизонтальном (*desktop*) и вертикальном (*tower*) исполнении.

В системном блоке размещаются следующие элементы:

- электронные схемы, управляющие работой компьютера (микропроцессор, оперативная память, контроллеры устройств и т.д.);
- дисководы (приводы) для компакт-дисков (CD-ROM);
- жесткий магнитный диск (*винчестер*);
- блок питания, преобразующий переменное сетевое напряжение в ряд постоянных, необходимых для работы электронных схем компьютера;
- прочие устройства, которые могут быть установлены в системном блоке.

Своеобразной базой ПК является *материнская плата*, на основе которой можно получить множество вариантов ЭВМ, наилучшим образом приспособленных для того или иного рода работ. На материнской плате располагаются различные устройства и микросхемы.

Микропроцессор – основная микросхема, выполняющая большинство математических и логических операций. В компьютерах IBM PC и совместимых чаще всего используются микропроцессоры фирм Intel и AMD. Важнейшим показателем микропроцессора является его *тактовая частота*, т.е. количество элементарных операций в секунду. Чем выше тактовая частота, тем выше производительность процессора.

Оперативная память или оперативное запоминающее устройство (ОЗУ) – набор микросхем, предназначенных для временного хранения данных, когда компьютер включен. Из ОЗУ процессор берет программы и исходные данные для обработки, в нее же записывает результат. При выключении компьютера ОЗУ очищается, и записанная в него информация уничтожается.

Шины – наборы проводников, по которым происходит обмен сигналами между внутренними устройствами компьютера.

Микросхема ПЗУ (постоянное запоминающее устройство) – микросхема, предназначенная для длительного хранения данных, в том числе и когда компьютер выключен. Комплект программ, находящихся в ПЗУ, образует базовую систему ввода-вывода (*BIOS – Basic Input-Output System*). Основное назначение программ этого пакета состоит в том, чтобы проверить состав и работоспособность компьютерной системы и обеспечить взаимодействие с клавиатурой, монитором, жестким диском и дисководом гибких дисков. Программы, входящие в BIOS, позволяют наблюдать на экране диагностические сообщения, сопровождающие запуск компьютера, а также вмешиваться в ход запуска с помощью клавиатуры.

Звуковая карта (плата). Как правило, звуковые карты обеспечивают возможность как цифро-аналогового, так и обратного преобразования звуковых колебаний, а также позволяют создавать некоторые звуковые эффекты.

В разъемы системной шины материнской платы встраиваются контроллеры различных устройств (монитора, DVD-ROM и т.д.). Разъемы для внешних устройств выводятся на заднюю панель системного блока.

На лицевой панели обычно выведены отсеки для приводов внешних запоминающих устройств, кнопки пуска и перезагрузки, светодиодные индикаторы питания и жесткого диска, а также (не всегда) цифровой индикатор тактовой частоты установленного процессора.

1.6.2. Мониторы

Монитор – это устройство вывода графической и текстовой информации, основанное на преобразовании электрического видеосигнала в видимое изображение. Монитор подключается к компьютеру через особую плату, находящуюся внутри компьютера и называемую *видеокартой*. В настоящее время существует два вида мониторов: с электронно-лучевой трубкой (ЭЛТ-мониторы) и жидкокристаллические (LCD-мониторы). Приведем основные характеристики мониторов.

Размер экрана – расстояние между противоположными углами экрана по диагонали. Обычно определяется в дюймах (14, 15, 17, 19, 20, 21 и т.д.). Для сложных чертежей и графического дизайна используются в основном мониторы с размером экрана более 17 дюймов.

Шаг сетки – минимальный размер одной точки экрана (пиксель – pixel). Обычно он составляет 0,25 мм. Чем мельче зерно, тем лучше качество изображения.

Разрешение – количество точек на экране, которые могут уместиться при образовании изображения (640×480, 800×600, 1024×768).

Частота кадровой развертки – частота управляющих сигналов, указывающих на необходимость перехода к новому кадру. Она важна не только для компьютера, но и для пользователя. При малой частоте смены кадров пользователь видит своеобразное мерцание экрана, что негативно влияет на зрение. Только при частоте кадровой развертки, равной или превышающей 75 – 80 Гц, это мерцание пропадает.

1.6.3. Клавиатура

Клавиатура – это клавишное устройство управления ПК. Служит для ввода данных, а также команд управления. Комбинация монитора и клавиатуры обеспечивает простейший интерфейс пользователя. Стандартная клавиатура имеет более 100 клавиш, функционально распределенных по следующим группам:

- *алфавитно-цифровые*. Для ввода знаковой информации и команд;
- *функциональные (F1-F12)*. Их назначение может быть различно в различных программах;
- *модификации кодов (Shift, Ctrl, Alt, Caps Lock)*. Изменяют стандартное назначение клавиш других групп;
- *клавиши управления курсором*. Управляют позицией ввода данных;
- *клавиши дополнительной панели*. Дублируют действия клавиш других групп.

1.6.4. Мышь

Мышь – устройство управления манипуляторного типа. Существуют мыши с двумя или тремя кнопками и, возможно, дополнительными органами управления. Перемещение мыши по плоской поверхности синхронизировано с перемещением указателя мыши на экране монитора.

Комбинация монитора и мыши обеспечивает графический интерфейс пользователя. Пользователь наблюдает на экране графические объекты и элементы управления. С помощью мыши он изменяет свойства объектов и приводит в действие элементы управления компьютерной системой, а с помощью монитора получает от нее отклик в наглядном виде.

В зависимости от принципа устройства, мыши делятся на *механические, оптомеханические* и *оптические*. В механических перемещение шарика внутри отслеживается механическими датчиками (колесиками), в оптомеханических также перемещается шарик, но его положение отслеживается оптическими датчиками, а в оптических – движущихся частей нет вообще.

1.7. Носители информации

1.7.1. Жесткий диск

Жесткий диск (винчестер, Hard Disc Drive, HD) – основное устройство для долговременного хранения больших объемов данных и программ. Жесткий диск представляет собой группу соосных дисков, имеющих магнитное покрытие и вращающихся с высокой скоростью. Таким образом, этот диск имеет не две поверхности, как должно быть у обычного плоского диска, а $2n$ поверхностей, где n – число отдельных дисков в группе. Над каждой поверхностью располагается головка, предназначенная для чтения-записи данных.

К основным параметрам жестких дисков относятся *емкость* и *производительность*. Емкость дисков зависит от технологии их изготовления. В настоящее время технологический уровень приближается к 80 Гбайт на пластину. Что касается производительности, то она характеризуется *скоростью внутренней передачи данных* и *средним временем доступа*. Сегодня все жесткие диски имеют очень высокий показатель скорости внутренней передачи данных. Параметр среднего времени доступа определяет интервал времени, необходимый для поиска нужных данных, и зависит от скорости вращения диска. Например, для дисков, вращающихся с частотой 7200 об/мин – 7 – 8 мкс, изделия более высокого уровня обеспечивают среднее время доступа к данным 5 – 6 мкс.

1.7.2. Гибкие магнитные диски

Гибкие магнитные диски (дискеты, флоппи-диски) используются, как правило, для оперативного переноса небольших объемов данных. Дискеты бывают 5- и 3-дюймовые. 5-дюймовые дискеты в последнее время практически не используются. 3-дюймовые дискеты имеют емкость 1,44 Мбайт.

1.7.3. Оптический привод

Оптический привод – устройство, имеющее механическую составляющую, управляемую электронной схемой, и предназначенное для считывания и (в некоторых моделях) записи информации с оптических носителей информации в виде пластикового диска с отверстием в центре (компакт-диск, DVD и т.д.); процесс считывания/записи информации с диска осуществляется при помощи лазера. Существуют следующие типы приводов: CD-ROM, CD-RW, DVD-ROM, DVD-RW, DVD-RW DL, HD DVD-ROM, HD DVD/DVD RW, BD-ROM, BD-RE, GD-ROM.

- CD-ROM – самый простой вид CD-привода, предназначенный только для чтения CD-дисков.
- CD-RW – такой же, как и предыдущий, но только способен записывать на CD-R/RW-диски.
- DVD-ROM – предназначение его состоит только в чтении CD/DVD-дисков.
- DVD/CD-RW – тот же DVD-ROM, но способный записывать на CD-RW-диски.
- DVD-RW – привод, способный не только читать CD(RW) И DVD (RW)-диски, но и записывать на них.
- DVD-RW DL – в отличие от предыдущего типа DVD-RW, способен также записывать на двухслойные оптические DVD-носители, отличающиеся от обычных большей емкостью.
- BD-RE-привод, способный читать/записывать на диски формата Blu-Ray.

Это усовершенствованная технология оптических носителей, в основе которой лежит использование лазера с длиной волны 405 нм (синий спектр излучения). Уменьшение длины волны лазера позволило сузить ширину дорожки в два раза по сравнению с DVD-диском и увеличить плотность записи данных. Уменьшение толщины защитного слоя в 6 раз повысило надежность операций чтения/записи на нескольких записываемых слоях. Диски Blue-Ray предназначены большей частью для записи

цифрового видео высокого разрешения. Например, на односторонний однослойный диск записывают до 2 часов видео в формате HDTV (телевидения высокой четкости) при скорости видеопотока до 54 Мбит/с.

- HD DVD-ROM-привод, читающий диски формата HD DVD.

HD DVD – это новое поколение оптических дисков, которые предназначены в первую очередь для хранения фильмов высокого разрешения (HDTV). Новый формат носителей позволяет записывать в три раза больший объем данных, по сравнению с DVD. Однослойные HD DVD-диски имеют емкость 15 Гб, двухслойные – 30 Гб. Как правило, HD DVD-привод может читать все форматы DVD- и CD-дисков.

- HD DVD/DVD RW – в отличие от предыдущего способен записывать на диски таких форматов как DVD-R, DVD-R, DVD-RW, DVD-RW, CD-R, CD-RW.

Современные приводы достигли высоких скоростей считывания информации с лазерного компакт-диска благодаря внедрению технологии CAV (Constant Angular Velocity – постоянная угловая скорость). В этом режиме частота оборотов диска остается постоянной, соответственно на периферийных участках данные считываются с большей скоростью (4 – 7,8 Мбайт/с), чем на внутренних участках (2 – 3,5 Мбайт/с). Средняя скорость считывания при этом гораздо ближе к минимальным значениям, поскольку запись на диске начинается с внутренних областей.

1.7.4. USB-флеш-накопитель

USB-флеш-накопитель – запоминающее устройство, использующее в качестве носителя флеш-память и подключаемое к компьютеру или иному считывающему устройству USB.

USB-флешки обычно съемные и перезаписываемые. Размер – около 5 см, вес – меньше 60 г. Получили большую популярность в 2000-е годы из-за компактности, легкости перезаписывания файлов и большого объема памяти (от 32 МБ до 256 ГБ). Основное назначение USB-накопителей – хранение, перенос и обмен данными, резервное копирование, загрузка операционных систем и др. Обычно устройство имеет вытянутую форму и съемный колпачок, прикрывающий разъем; иногда прилагается шнур для ношения на шее. Современные флешки могут иметь самые разные размеры и способы защиты разъема, а также «нестандартный» внешний вид и различные дополнительные возможности.

Принцип действия. Принцип работы полупроводниковой технологии флеш-памяти основан на изменении и регистрации электрического заряда в изолированной области (кармане) полупроводниковой структуры. Изменение заряда (запись/стирание) производится приложением между затвором и истоком большого потенциала чтобы напряженность электрического поля в тонком диэлектрике между каналом транзистора и карманом оказалась достаточна для возникновения туннельного эффекта. Для усиления эффекта туннелирования электронов в карман при записи применяется небольшое ускорение электронов путем пропускания тока через канал полевого транзистора. Чтение выполняется полевым транзистором, для которого карман выполняет роль затвора. Потенциал плавающего затвора изменяет пороговые характеристики транзистора, что и регистрируется цепями чтения.

Преимущества:

- Малый вес, бесшумность работы и портативность.
- Универсальность – современные компьютеры, телевизоры и DVD-проигрыватели имеют USB-разъемы.
- Более устойчивы к механическим воздействиям (вибрации и ударам) по сравнению с жесткими дисками.
- Работоспособность в широком диапазоне температур.
- Низкое энергопотребление (в отличие от CD, DVD и жестких дисков).
- Не подвержены воздействию царапин и пыли, которые были проблемой для оптических носителей и дискет.

Недостатки:

- Ограниченное число циклов записи-стирания перед выходом из строя.
- Способны хранить данные полностью автономно до 5 лет. Наиболее перспективные образцы – до 10 лет.
- Скорость записи и чтения ограничены пропускной способностью USB.
- В отличие от компакт-дисков, имеют недостатки, свойственные любой электронике:
 - чувствительны к электростатическому разряду – обычное явление в быту, особенно зимой;
 - чувствительны к радиации.

Недостаток многих разъемов USB – частые подключения/отключения.

1.8. Периферийные устройства персонального компьютера

1.8.1. Принтеры

Принтеры – это устройства для вывода графической и текстовой информации на бумагу или прозрачный носитель.

Матричные принтеры. *Матричные принтеры* печатают специальными иглами, ударяющими по красящей ленте. В результате образуется комбинация точек, изображающая символ. Матричные принтеры – медленные и шумные устройства, обеспечивающие наихудшее качество печати, но цена отпечатанной ими страницы минимальна.

Струйные принтеры. Изображение в струйных принтерах формируется из пятен, образующихся при попадании капель специальных чернил на бумагу. Струйные принтеры довольно быстрые и бесшумные, обеспечивают неплохое качество печати на бумаге определенного сорта. Стоимость отпечатанной страницы у них выше, чем у матричных принтеров. Важнейшей особенностью струйной печати является возможность создания высококачественного цветного изображения.

Лазерные принтеры. Эта группа принтеров обладает наивысшим качеством печати, близким к типографскому. Лазерные принтеры переносят изображение на бумагу при помощи светочувствительного барабана, к освещенным лазером участкам которого прилипает тонер. Затем, в результате соприкосновения бумажного листа с барабаном, происходит перенос тонера на бумагу. Лазерные принтеры отличаются высокой скоростью печати, бесшумны.

1.8.2. Устройства ввода графических данных

Для ввода графической информации в компьютер используют *сканеры, дигитайзеры, цифровые фотокамеры*. Сканеры можно также использовать для ввода знаковой информации. В этом случае исходный материал вводится в графическом виде, после чего обрабатывается специальными программными средствами – *программами распознавания образов*.

Основные виды сканеров:

– *планшетные*. Предназначены для ввода графической информации с прозрачного и непрозрачного листового материала;

– *барабанные*. Предназначены для сканирования исходных изображений, имеющих высокое качество, но недостаточные линейные размеры (фотонегативов, слайдов и т.п.);

– *сканеры форм*. Предназначены для ввода данных со стандартных форм, заполненных механически или вручную. Необходимость в этом воз-

никает при проведении переписи населения, обработке результатов голосований и анализе анкетных данных;

– *штрих-сканеры*. Предназначены для ввода данных, закодированных в виде штрих-кода. Такие устройства имеют применение в розничной торговой сети.

Дигитайзеры. Эти устройства предназначены для ввода художественной графической информации. В основе принципа действия этих устройств лежит фиксация перемещения специального пера относительно планшета. Такие устройства удобны для художников и иллюстраторов, т.к. позволяют им создавать экранные изображения привычными приемами и инструментами (карандаш, перо, кисть).

Цифровые фотокамеры. Основным параметром цифровых фотоаппаратов является разрешающая способность. Наилучшие потребительские модели в настоящее время обеспечивают разрешение 1920×1600 точек и более. У профессиональных моделей эти параметры еще выше.

1.8.3. Веб-камера

Веб-камера – цифровая видео или фотокамера, способная в реальном времени фиксировать изображения, предназначенные для дальнейшей передачи по сети Интернет (в программах типа Skype, Instant Messenger или в любом другом видеоприложении). Веб-камеры, доставляющие изображения через Интернет, записывают изображения на веб-сервер либо по запросу, либо непрерывно, либо через регулярные промежутки времени. Это достигается путем подключения камеры к компьютеру или благодаря возможностям самой камеры. Некоторые современные модели обладают аппаратным и программным обеспечением, которое позволяет камере самостоятельно работать в качестве веб-сервера, FTP-сервера, FTP-клиента и (или) отсылать изображения электронной почтой.

Веб-камеры, предназначенные для видеоконференций, – это, как правило, простые модели камер, подключаемые к компьютеру, на котором запущена программа типа Instant Messenger. Модели камер, используемые в охранных целях, могут снабжаться дополнительными устройствами и функциями (детекторы движения, подключение внешних датчиков и т.п.)

Первая в истории веб-камера была запущена в 1991 году и показывала кофеварку в Троянской комнате Кембриджского университета. Сейчас она не работает, поскольку была отключена 22 августа 2001 года. Подобно многим сетевым технологиям, веб-камеры и видеочаты приобрели массовую

популярность. Необходимость в «живых» видеоизображениях породила вебкамеры, способные вещать через Интернет в формате видеопотока, не требующего от зрителя необходимости вручную обновлять изображение.

Веб-камера содержит объектив, оптический фильтр, матрицу, схему цифровой обработки изображения, схему компрессии изображения и опционально веб-сервер для подключения к Сети.

1.8.4. Модемы

Модемы – это устройства, предназначенные для обмена информацией между удаленными компьютерами по каналам связи (проводным, оптоволоконным, кабельным, радиочастотным линиям). Модем необходим для подключения к сети Internet, а также для обеспечения внутрикорпоративной связи, создания локальных сетей и т.д. По своему исполнению модемы бывают *внешними (External)*, т.е. подключаемыми к системному блоку снаружи, и *внутренними (Internal)*, т.е. встроенными в системный блок.

К основным потребительским параметрам модемов относится производительность (бит/с). От производительности зависит объем данных, передаваемых в единицу времени. Максимальная скорость передачи данных у модемов может быть от 2400 до 115200 бит/с.

1.9. Программное обеспечение средств вычислительной техники

Программы – это упорядоченные последовательности команд. Конечная цель любой компьютерной программы – управление аппаратными средствами. Программное и аппаратное обеспечение ПК работают в неразрывной связи и непрерывном взаимодействии. Состав программного обеспечения (ПО) ПК называют *программной конфигурацией*. Все ПО подразделяется на несколько уровней: базовый, системный, служебный, прикладной.

1.9.1. Базовый уровень

Программы этого уровня отвечают за взаимодействие с *базовыми аппаратными средствами*. Базовые программные средства, как правило, входят в состав базового оборудования и хранятся в специальных микросхемах, называемых *постоянными запоминающими устройствами (ПЗУ)*. Программы и данные записываются в эти микросхемы на этапе производства и не могут быть изменены в процессе эксплуатации.

1.9.2. Системный уровень

Системный уровень – переходный. Программы этого уровня обеспечивают взаимодействие прочих программ компьютерной системы с программами базового уровня и непосредственно с аппаратным обеспечением. К программам системного уровня относятся *драйверы устройств* (отвечают за взаимодействие других программ с устройствами компьютера) и программные средства для обеспечения пользовательского интерфейса.

Совокупность ПО системного уровня образует *ядро операционной системы компьютера*. То есть, наличие ядра ОС – неперемнное условие для возможности практической работы человека с вычислительной системой.

1.9.3. Служебный уровень

Назначение служебных программ (*утилит*) состоит в автоматизации работ по проверке, наладке и настройке компьютерной системы. Эти программы взаимодействуют как с программами базового уровня, так и с программами системного уровня. Во многих случаях они используются для расширения или улучшения функций системных программ.

1.9.4. Прикладной уровень. Классификация прикладных программных средств

Программное обеспечение прикладного уровня представляет собой комплекс прикладных программ, с помощью которых на данном рабочем месте выполняются конкретные.

Текстовые редакторы. Основная функция этого класса прикладных программ заключается во вводе и редактировании текстов. С этого класса прикладных программ обычно начинают знакомство с ПО и на нем отбатывают первичные навыки взаимодействия с компьютерной системой.

Текстовые процессоры. В отличие от текстовых редакторов позволяют не только вводить и редактировать тексты, но и *форматировать* их. Под форматированием понимают оформление документов путем применения нескольких шрифтовых наборов, использования методов выравнивания текста, встраивания в текстовый документ объектов иной природы (рисунков), а также управления взаимодействием графики и текста.

Графические редакторы. Программы этого класса предназначены для создания и/или обработки графических изображений. Различают *растровые редакторы* (когда графический объект представлен в виде комбинации точек), *векторные редакторы* (когда элементарным объектом изображения является линия) и программные средства для создания и обработки трехмерной графики (*3D-редакторы*).

Системы управления базами данных (СУБД). Базами данных (БД) называют огромные массивы данных, организованных в табличные структуры. Основные функции СУБД: создание пустой (незаполненной) структуры БД; предоставление средств ее заполнения, обеспечение возможности доступа к данным, а также предоставление средств поиска и фильтрации.

Электронные таблицы. Электронные таблицы предоставляют комплексные средства для хранения различных типов данных (акцент на числовые данные) и их обработки (акцент на преобразование данных, а не на хранение).

Браузеры. Эти программные средства предназначены для просмотра электронных документов, выполненных в формате *HTML* (документы этого формата используются в качестве Web-документов).

Бухгалтерские системы. Это специализированные программные средства, сочетающие в себе функции текстовых и табличных редакторов, электронных таблиц и систем управления данными. Предназначены для автоматизации подготовки первичных бухгалтерских документов предприятия и их учета, для ведения счетов плана бухгалтерского учета, а также для автоматической подготовки регулярных отчетов по итогам производственной, хозяйственной и финансовой деятельности в форме, принятой для предоставления в налоговые органы, внебюджетные фонды и органы статистического учета.

Финансовые аналитические системы. Программы этого класса используются в банковских и биржевых структурах. Они позволяют контролировать и прогнозировать ситуацию на финансовых, товарных и сырьевых рынках, производить анализ текущих событий, готовить сводки и отчеты.

1.10. Назначение, классификация и основные функции операционных систем

Операционная система – это комплекс программ, которые загружаются при включении компьютера. Операционная система производит диалог с пользователем, осуществляет управление компьютером, его ресурсами (процессоры, память, дисковые накопители, сетевые коммуникационные средства, принтеры и другие устройства), запускает другие (прикладные) программы на выполнение, кроме того ОС обеспечивает пользователю и прикладным программам удобный *интерфейс* (способ общения) с устройствами компьютера.

Классифицировать ОС для ПК можно следующим образом:

- однозадачные и многозадачные;
- однопользовательские и многопользовательские.

Однозадачные ОС позволяют запустить одну программу в основном режиме. *Многозадачные ОС* позволяют запустить одновременно несколько программ, которые будут работать параллельно, не мешая друг другу. Большинство современных графических ОС – многозадачные.

Однопользовательские ОС позволяют работать на компьютере только одному человеку. В многопользовательской системе работу можно организовать так, что каждый пользователь будет иметь доступ к информации общего доступа, введя пароль, к личной информации, доступной только ему. Отличием многопользовательских систем является наличие средств защиты информации пользователей от несанкционированного доступа.

1.10.1. Файловая система

Файловая система – это часть ОС, предназначенная для организации работы с хранящимися на диске данными и обеспечения совместного использования файлов несколькими пользователями и процессами. Файловая система ОС определяет структуру хранения файлов и папок на диске, правила задания имен файлов, допустимые атрибуты, права доступа и др.

Разные ОС могут применять разные файловые системы. Например, операционные системы MS-DOS и Windows 95 могут работать только с файловой системой *FAT 16*. Предельный размер диска для нее равен 2 Гбайт. Если физический диск имеет больший размер, его делят на несколько логических дисков.

Операционная система Windows 98 работает не только с файловой системой *FAT 16*, но и *FAT 32*. В системе *FAT 32* нет практических ограничений на размер жесткого диска. Системы *FAT 32* и *FAT 16* совместимы сверху вниз, т.е. файлы, записанные в системе *FAT 16*, читаются на компьютерах, работающих в системе *FAT 32*, но не наоборот.

Операционные системы Windows 2000 и выше могут работать в файловых системах *FAT 16*, *FAT 32* и *NTFS*. *NTFS* – специфическая файловая система. Она обеспечивает повышенную скорость работы, но несовместима с ОС Windows 9x.

1.10.2. Файловая структура

Файл – это область на диске или другом носителе информации, обладающая уникальным собственным именем. Собственное имя файла состоит из двух частей: *собственно имени* и *расширения* (или *типа*). В имени разрешается использовать пробелы и несколько точек. Расширением имени считаются все символы, идущие после последней точки. Использование расширения в имени файла обязательным не является.

Тип файла (расширение) определяет его предназначение и способ использования. Например, обращение к программному файлу запускает программу. Помимо готовых к выполнению программ в файлах могут храниться тексты программ, документы, и любые другие данные. Примеры распространенных типов файлов:

- *.exe* – исполнимые;
- *.txt* – текстовые (читаемые);
- *.doc* – документы Microsoft Word;
- *.hlp* – файлы помощи и др.

При выборе имени создаваемого файла необходимо учитывать следующие требования:

- имя файла в папке должно быть уникальным;
- имя файла может иметь длину до 255 символов – такое имя файла называется *длинным*;
- не допускается использование в именах следующих символов: \, /, *, ?, :, “, |, < и >;
- не разрешается использовать в качестве имен файлов имена, зарезервированные в ОС под имена устройств: *prn* – принтер; *con* – консоль (при вводе – клавиатура, при выводе – монитор); *nul* – пустое устройство, т.к. все операции, указанные для него игнорируются; *lpt1-lpt3* – устройства, подключаемые к параллельным портам компьютера; *com1-com3* – устройства, подключаемые к последовательным портам;
- при сравнении имен не различается написание прописными и строчными буквами, хотя в списках и полях имена файлов выводятся с учетом того, в каком регистре они были введены.

Путь доступа к файлу начинается с имени устройства и включает все имена папок, через которые он проходит. В качестве разделителя используется символ «\». Собственное имя файла вместе с путем доступа к нему считается *полным именем файла* (рис. 1.2).



Рис. 1.2. Пример записи полного имени файла

Кроме имени и расширения файла ОС хранит для каждого файла дату его создания (изменения) и несколько атрибутов файла. *Атрибуты* – это дополнительные параметры, определяющие свойства файлов:

- *только для чтения (Read only)*;
- *скрытый (Hidden)*;
- *системный (System)*;
- *архивный (Archive)*.

1.11. Обмен данными в Windows

Буфер обмена – это область оперативной памяти, резервируемая системой Windows для организации обмена данными между приложениями. В любой момент времени в ней можно хранить только один объект. При попытке поместить туда другой объект предыдущий объект перестает существовать. Поэтому буфер обмена не используют для длительного хранения чего-либо. Поместив объект в буфер, немедленно выполняют вставку из буфера в нужное место.

Принцип работы с буфером обмена:

- 1) открыть папку-источник. Выделить щелчком нужный объект;
- 2) скопировать или вырезать объект в буфер. В первом случае объект остается в папке-источнике. Во втором случае он удаляется из папки источника, но может некоторое время храниться в буфере;
- 3) открыть папку-приемник и поместить в нее объект из буфера обмена.

Три указанные операции (*копировать*, *вырезать* и *вставить*) можно выполнять разными способами:

- использовать команды меню *Правка*;
- пользоваться кнопками панели инструментов:  – *вырезать*,  – *копировать*,  – *вставить*;
- использовать комбинации клавиш клавиатуры: *Ctrl + C* – *копировать в буфер*; *Ctrl + X* – *вырезать в буфер*; *Ctrl + V* – *вставить из буфера*.

Эти приемы работают во всех приложениях Windows. Через буфер обмена можно переносить фрагменты текстов из одного документа в другой. Можно также переносить иллюстрации, звукозаписи, видеофрагменты, файлы, папки и вообще любые объекты.

1.12. Программы-архиваторы

Назначение *программ-архиваторов* – экономить место на диске за счет сжатия (упаковки) одного или нескольких исходных файлов в архивный файл. Программы-архиваторы используются для хранения в

упакованном виде больших объемов информации, которая понадобится только в будущем; переноса информации между компьютерами с помощью дискет или электронной почты; создания в сжатом виде резервных копий файлов. В результате работы программ-архиваторов создаются архивные файлы (архивы).

Типовые функции программ-архиваторов состоят:

- в помещении исходных файлов в архив;
- в извлечении файлов, удалении файлов из архива;
- в просмотре оглавления архива;
- в верификации (проверки) архива.

Среди современных программ-архиваторов выделяют: WinRAR (разработка Е. Рошаль), WinZip (фирма Niko Mak Computing) и др. Мы рассмотрим использование архиватора WinRAR, отличающегося большой степенью сжатия, работой с длинными именами файлов, удобным интерфейсом. Пользовательский интерфейс WinRAR содержит строку меню, панель инструментов и рабочую область, в которой показаны все файлы текущей папки (рис. 1.3).

Строка меню архиватора включает пункты *Файл*, *Команды*, *История*, *Избранное*, *Параметры* и *Справка*, содержащие сгруппированные по функциональному назначению команды архиватора.

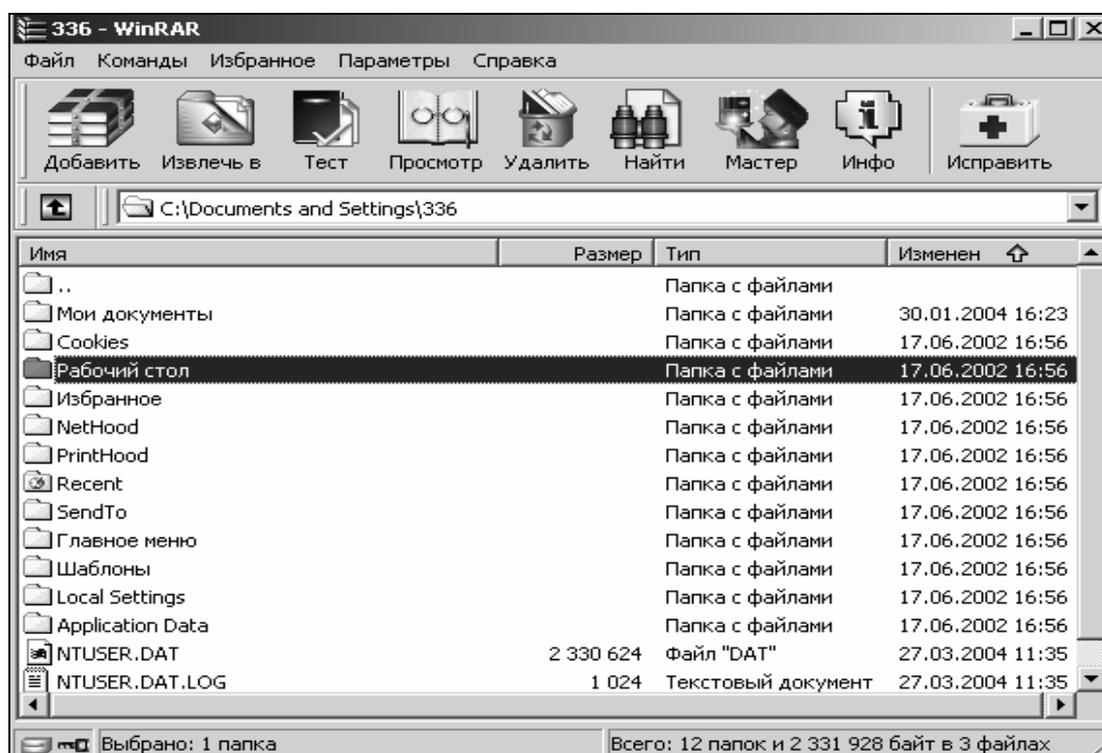


Рис. 1.3. Пользовательский интерфейс архиватора WinRAR

Команды меню *Файл* выполняют операции над файлами, содержащимися в архиве или помещаемыми в архив.

Команда *Пароль* применяется при установке пароля на вновь создаваемый архивный файл. При выборе этой команды пользователю следует в появившемся окне набрать и подтвердить пароль. Впоследствии без знания этого пароля невозможно будет получить доступ к содержимому хранящихся в архиве файлов.

Следующая группа команд используется для выделения нескольких файлов. Так, команда *Выделить все* автоматически выделяет все файлы текущей папки. Команда *Выделить группу* активизирует маску ввода шаблона файлов, удовлетворяющих некоторому критерию. Аналогично команда *Снять выделение* вызывает маску шаблона для отмены выделения файлов.

Меню *Команды* содержит команды обработки содержимого архива.

Для создания архива или добавления данных в существующий архив используется команда *Добавить файлы в архив* или кнопка . При этом в архив помещаются предварительно выделенные файлы. При выборе этой команды на экране появляется диалоговое окно *Имя и параметры архива* (рис. 1.4).

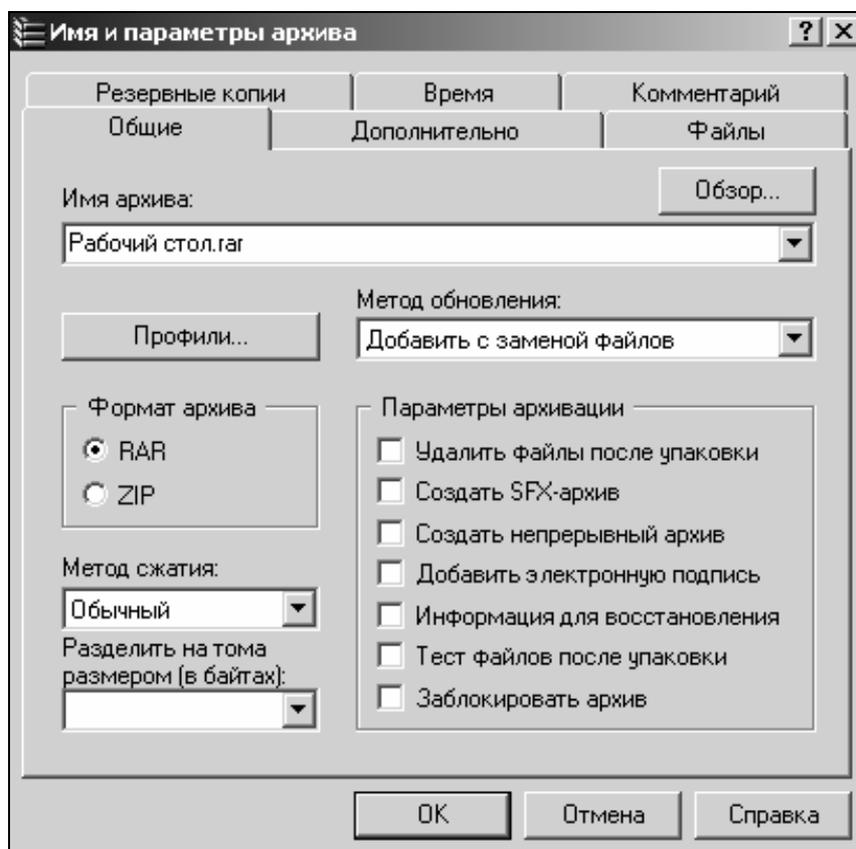


Рис. 1.4. Диалоговое окно *Имя и параметры архива*

В поле *Архив* этого окна указывается местонахождение помещаемых в архив файлов. По умолчанию архив создается в текущей папке. Для указания другой папки следует использовать кнопку *Обзор*.

Содержимое поля *Метод сжатия* определяет степень сжатия создаваемого архива. По умолчанию установлен вариант *Обычный*. Если установить наилучшую степень сжатия, то архивный файл будет занимать меньший объем, однако время его создания возрастет.

В поле *Размер словаря* задается длина фрагмента, в рамках которого алгоритм сжатия ищет повторения для кодирования и сжатия.

Параметр *Размер тома* применяется при создании многотомных архивов, его установка позволяет создавать архив в виде нескольких файлов, размер которых не превосходит заданного размера тома. Это особенно актуально при необходимости последующего переноса архива на дискетах, если архив превышает емкость имеющихся носителей, или при пересылке по электронной почте.

В поле со списком *Метод обновления* определяются варианты помещения файлов в архив. Устанавливаемый по умолчанию режим *Добавить с заменой файлов* определяет архивирование всех выделенных пользователем файлов. Указание режима *Обновить существующие файлы*, позволяет добавить в архив лишь те файлы, старые версии которых уже находятся в архиве. Режим *Добавить с обновлением файлов* помещает в архив файлы, копии которых в архиве отсутствуют.

Группа команд *Параметры архивации* позволяет выбрать алгоритм сжатия данных. По умолчанию программа настроена на базовый вариант. Однако, например, для мультимедийных данных целесообразно использовать вариант *Мультимедиа-сжатие*.

Когда настройка завершена, следует нажать кнопку *ОК*. Появившееся после этого окно иллюстрирует текущее состояние процедуры архивации (степень обработки очередного файла набора и общее состояние выполнения всей процедуры).

На этом процедура создания архива или добавления данных в существующий архив заканчивается.

Команда *Восстановить архив* используется при нарушении целостности архива, возникающем, например, в результате его длительного хранения.

Архиватор WinRAR позволяет удалять ненужные файлы, как это делается в программе *Проводник*. Для этого используется команда *Удалить файлы* нажатие клавиши *Delete* на клавиатуре или кнопки 

Остальные команды этого меню относятся только к файлам, содержащимся в архиве, и становятся доступными, если в рабочей области WinRAR выделен файл архива.

Команда *Извлечь файлы из архива* обеспечивает распаковку предварительно выделенных пользователем файлов из данного архива. Если необходимо разархивировать файл не в текущую папку, то следует воспользоваться командой *Извлечь в другую папку* и указать путь к этой папке-получателю. Тестирование отдельных файлов в архиве на предмет возможных повреждений их структуры производится с помощью команды *Протестировать файлы в архиве*. Эту команду следует применять для проверки целостности файлов при их длительном хранении, особенно на ненадежных магнитных носителях.

Архив можно снабдить комментарием, воспользовавшись командой *Добавить архивный комментарий* или соответствующей кнопкой  на панели инструментов. Выбор команды *Добавить информацию для восстановления* вызывает специальную процедуру, которая вносит в текущий архив дополнительные данные, повышающие его устойчивость к сбоям. Однако это приводит к некоторому увеличению объема архива.

Архиватор WinRAR позволяет создавать самораспаковывающиеся архивы, разворачивающиеся при запуске их на исполнение. Для этого требуется выполнить команду *Преобразовать архив в SFX*. Эта команда становится доступной, если в рабочей области WinRAR выделен файл архива.

Выбор команды *Информации об архиве* позволяет получить сведения об архиве, выделенном в данный момент в рабочей области (размер и количество файлов в архиве; коэффициент сжатия архива; наличие комментариев; наличие пароля; ОС, для которой этот архив создан).

Команды меню *История* обеспечивают доступ к последним обрабатываемым архивам, с которыми работал пользователь.

Группа команд меню *Параметры* предназначена для настройки основных параметров архиватора WinRAR и регистрации пользователей через Internet. С помощью этого меню пользователь имеет возможность определить интерфейс архиватора; задать значения по умолчанию основных параметров архиватора (метода сжатия, размера словаря); определить папку, в которую следует помещать файл архива и др.

Команды меню *Помощь* описывают возможности работы архиватора, поясняют технику работы с ним и содержат информацию о разработчике и процедуре приобретения архиватора.

2. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

Процесс решения задачи на ЭВМ состоит из восьми этапов:

- 1) постановка задачи;
- 2) математическое описание задачи;
- 3) алгоритмизация задачи;
- 4) программирование;
- 5) разработка тестовой задачи;
- 6) перенос программы на машинные носители;
- 7) отладка программы;
- 8) получение и анализ результатов.

Рассмотрим более подробно этап алгоритмизации задачи.

2.1. Понятие и основные свойства алгоритма

Алгоритм – это конечная последовательность однозначных предписаний, исполнение которых позволяет с помощью конечного числа шагов получить решение задачи, однозначно определяемое исходными данными.

Свойства алгоритма:

– *дискретность*. Это свойство состоит в том, что алгоритм должен представлять процесс решения задачи как последовательность простых шагов. Таким образом, преобразование исходных данных в результат осуществляется во времени дискретно;

– *определенность*. Каждая команда алгоритма должна быть четкой, однозначной;

– *результативность*. Алгоритм должен приводить к решению поставленной задачи за конечное число шагов;

– *массовость*. Алгоритм решения задачи разрабатывается не для одной конкретной задачи, а для целого класса однотипных задач, различающихся лишь исходными данными.

Алгоритм может быть предназначен для выполнения его человеком или автоматическим устройством, называемым *формальным исполнителем*.

Объекты, над которыми исполнитель может совершать действия, образуют так называемую *среду исполнителя*.

Существуют следующие способы записи алгоритма:

– *словесно-формульное описание* (на естественном языке с использованием математических формул). Данный способ записи алгоритма состоит из перечня действий (шагов), каждый из которых имеет порядковый номер.

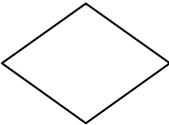
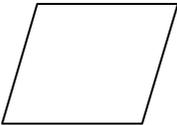
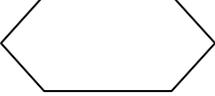
Словесное описание алгоритмов применяют при решении несложных задач, но оно малоприспособно для представления сложных алгоритмов из-за отсутствия наглядности;

– *графическое описание в виде блок-схемы*. Для обозначения шагов решения в виде схемы алгоритма используются специальные обозначения (символы). Перечень наиболее часто употребляемых символов приведен в таблице 2.1.

– *описание на каком-либо языке программирования* (программа).

Таблица 2.1

Символы для создания блок-схем алгоритмов

Наименование символа	Обозначение символа	Функция символа
Процесс		Выполнение операции или группы операций, в результате которых изменяется значение, форма представления или расположения данных
Решение		Выбор направления выполнения алгоритма или программы в зависимости от некоторых переменных условий
Ввод-вывод		Преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод)
Пуск-останов		Начало, конец, прерывание процесса обработки данных или выполнения программы
Предопределительный процесс		Использование ранее созданных или отдельно описанных алгоритмов и программ
Соединительный		Указание связи между прерванными линиями потока, связывающими символы
Модификация		Организация циклических вычислений
Комментарий		Связь между элементами схемы и пояснением

Программа – это набор машинных команд, который следует выполнить компьютеру для реализации того или иного алгоритма. *Программа* на языке программирования представляет собой совокупность операторов, записанных в соответствии с принятым синтаксисом.

Язык программирования – это набор символов и терминов, который в соответствии с правилами синтаксиса описывает алгоритм решения задачи.

Программирование (programming) – это процесс создания последовательности действий (операций), проводимый в целях достижения требуемого результата. Процесс программирования состоит из стадий: *формулирования, разработки программы*, включая *кодирование* и *тестирование*, и *создания новых версий*.

Существуют сотни реально используемых языков программирования, каждый для своей области применения. Уровень языка программирования определяется в зависимости от степени детализации алгоритма. Причем, чем меньше детализация, тем выше уровень языка. Различают три уровня языков программирования:

- *машинные*;
- *машинно-ориентированные* (ассемблеры);
- *машинно-независимые* (языки высокого уровня).

Машинные языки и машинно-ориентированные языки – это языки *низкого уровня*, которые требуют указания мелких деталей процесса обработки данных.

Языки высокого уровня во многом имитируют естественные языки, используют многие разговорные слова, общепринятые математические символы. Различают языки следующих групп: алгоритмические, предназначенные для однозначного описания алгоритмов (такие, как *Basic, Pascal, C*); логические, ориентированные не на разработку алгоритма решения задачи, а на систематическое и формализованное описание задачи (например, *Prolog, Lisp*); объектно-ориентированные, основанные на понятии объекта и действиях над ним (к примеру, *Object Pascal, C++, Java*). Языки программирования подробно изучаются при подготовке профессиональных программистов.

Средством разработки и развития ПО является *инструментальное программное обеспечение* (*software tools*). Инструментальное программное обеспечение, называемое также инструментарием, характеризуется набором программ по созданию и модернизации новых программ, среди них текстовый редактор, графический редактор, транслятор, загрузчик, а также средства отладки программ.

После написания программы, она подвергается *тестированию*. Цель тестирования состоит в определении ее работоспособности и степени го-

товности. После тестирования проводят анализ поведения программы. В связи и с тем, что процесс создания большой программы чрезвычайно трудоемок и ряд ошибок выявляются через некоторое время, доказательство правильности работы программы – сложный процесс.

2.2. Введение в программирование на языке Паскаль (общая структура программ, основные элементы языка)

Для изучения раздела «Основы алгоритмизации и программирование» целесообразно использовать приложение Pascal ABC, отличающееся наиболее простым и удобным интерфейсом, для начинающего пользователя. Паскаль позволяет в доступной форме изучить основы программирования, структурирования программ. Научившись программировать в Паскале, пользователь сможет разобраться при необходимости с более сложными профессиональными языками программирования Delphi, Visual Basic, в которых структуры программ аналогичны.

Общая структура Паскаль-программы:

Program {заголовок программы}
Uses {подключаемые библиотеки}
Label {объявление меток}
Const {объявление констант}
Type (объявление типов)
Var {объявление переменных}
Procedure (Function) {описание процедуры (функции)}

Раздел основного блока программы:

Begin{начало блока}
операторы
End.{конец блока}

Кроме того, в программе могут использоваться **комментарии** – любой текст, ограниченный (* *) или { }. Комментарий может быть помещен в любом месте программы.

В строке **Uses** производится подключение используемых в данной программе библиотек (стандартных модулей). Паскаль содержит ряд **модулей в своей библиотеке модулей**, в том числе **System, Dos и WinDos, Crt, Printer, Graph.**

В разделе описания меток **Label** содержатся сведения об используемых в программе **метках**. С помощью меток обеспечивается в программе безусловный переход к другим операторам. Переход по меткам выполняется оператором

Goto <имя метки>;

Метки отделяются от помечаемого оператора знаком двоеточие «:». Метки могут быть целочисленными (от 0 до 9999) или идентификаторами. Каждая описанная метка обязательно должна появиться в программе.

В разделе описания типов (**type**) программист может задавать свои типы, сформированные на основе определенных правил с использованием стандартных типов.

Type

Имя типа 1 = структура типа на основе базового типа 1;
... ;

В разделе **Var** описываются переменные, используемые в программе:

Var

список 1: имя типа 1;
список 2: имя типа 2;
... ;

В списке переменных их имена отделяются запятой.

В разделе текстов процедур и функций в соответствии с определенными правилами производится описание процедур и функций.

Основной блок программы состоит из ряда операторов и является выполняемой частью программы. Он начинается со слова **Begin** и заканчивается словом **End.**, после которого должна стоять **точка**.

Операторы языка **Паскаль** не привязаны к определенной позиции строки. В одной строке можно указывать несколько операторов. Исполняемые операторы отделяются друг от друга знаком «;». Допускается перенос с одной строки на другую частей операторов (но без разделения ключевых слов).

Алфавит языка Паскаль включает:

- латинские буквы;
- цифры;
- специальные символы (+ - * / = ^ < > () [] { } . , : ; ' # \$ и др.).

Кроме того, в символьных константах и комментариях могут использоваться и другие знаки (например, буквы русского алфавита).

При формировании и описании своих базовых элементов язык Паскаль использует зарезервированные слова, которые не могут быть переопределены пользователем:

and	end	not	shr
array	file	object	then
asm	for	of	to
abs	function	or	type
begin	goto	procedure	unit
case	if	program	until

const	in	record	uses
div	interrupt	repeat	var
do	label	string	with
downto	mod	set	while
else	nil	shl	xor

Простые типы данных

Два типа T1 и T2 являются эквивалентными или совместимы, если выполняется одно из двух условий:

- T1 и T2 представляют собой одно и то же имя типа;
- тип T2 описан с использованием типа T1 с помощью равенства или последовательности равенств. Например:

```

type
  T1 = integer;
  T2 = T1;

```

Каждому из перечисленных ранее типов соответствует свой набор операций. Так, для целочисленного и вещественного типов используются операции арифметических действий: сложение (+), вычитание (-), умножение (*), и деление (/), которые выполняются в соответствии с их математическим приоритетом. Приоритет действий в выражениях может быть изменен путем введения круглых скобок. Кроме этого, для целых значений применима операция целочисленного деления **div**, которая возвращает частное от деления (например, $7 \text{ div } 3 = 2$) и операция **mod**, которая возвращает остаток от деления двух целых чисел (например, $7 \text{ mod } 3 = 1$). Кроме того, Паскаль предусматривает возможность вычисления для каждого типа данных значений стандартных функций:

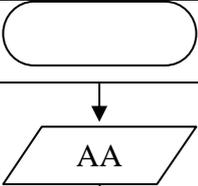
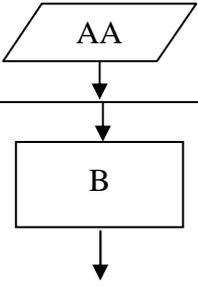
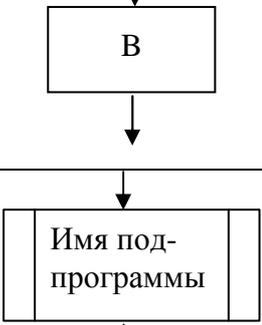
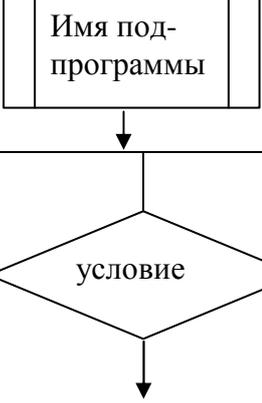
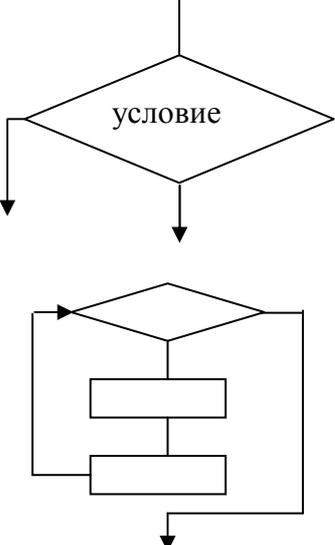
cos(x)	косинус числа;
sin(x)	синус числа;
abs(x)	абсолютное значения числа;
sqr(x)	квадрат числа;
sqrt(x)	квадратный корень числа;
arctan(x)	арктангенс числа;
exp(x)	экспонента числа;
ln(x)	натуральный логарифм числа;
round(x)	округление числа до целого;
trunc(x)	вычисление дробной части;
int(x)	вычисление целой части;
odd(x)	нечетность числа.

Примечание. Во всех тригонометрических функциях число x указывается в радианах.

Данные *логического типа* могут принимать два значения: True – «истина» и False – «ложь». Для логического типа выполняются операции сравнения (<, >, =, <=, >=, <>) и логические операции: **and** – логическое «и»; **or** – логическое «или»; **xor** – логическое исключающее «или»; **not** – логическое «не».

Символьный тип предназначен для хранения одного символа (буквы, знака, цифры или кода). Имя типа *Char*. Формат представления символического типа 1 байт. Диапазон порядковых номеров типа 0 .. 255. **CHR(x)** – эта функция преобразует выражение x типа *Byte* в символ и возвращает его своим значением, например, CHR(65) возвращает символ A.

Операторы языка и их аналогии в алгоритмических структурах

Графический символ	Назначение	Оператор языка
	Блок начала алгоритма. Блок конца алгоритма	BEGIN END.
	Ввод данных и вывод данных	READ(AA) WRITE(AA)
	Вычислительный процесс с сохранением результата в переменной B. Присвоение значений переменным	B:=AA+SQR(AA) G:=0; U:=0
	Предопределенный процесс: работа подпрограммы (функции или процедуры)	Вызов функции или вызов процедуры
	Блок принятия решения. Точка разветвления алгоритма может использоваться и в других случаях: а) при разработке циклического алгоритма с предусловием;	IF .. THEN...ELSE WHILE ... DO ...

Графический символ	Назначение	Оператор языка
	б) при разработке циклического алгоритма с послеусловием	REPEAT... . . UNTIL ...
	Блок модификации для циклических структур со счетчиком	FOR ..TO....DO

2.3. Программирование линейных алгоритмов

Алгоритм линейной структуры (следование) – это алгоритм, в котором все действия выполняются последовательно друг за другом. Такой порядок выполнения действий называется естественным.

В качестве примера составим программу для задачи, в которой требуется вычислить сумму двух заданных чисел, которые необходимо ввести с клавиатуры. Результат необходимо вывести на экран.

```

program sum;
var
a, b, s: real;
begin
write('Введите значения чисел a и b');
readln(a, b);
s := a + b;
writeln ('Сумма ='; s:7:2);
end.

```

Таким образом, на экран выведется **Сумма = 38.70**. Выведенное вещественное число имеет две цифры в дробной части и поле вывода в семь позиций экрана.

Пример 1. Пусть требуется преобразовать величину, выраженную в минутах, в соответствующее ей значение, выраженное в часах и минутах.

```

program Preobr_1;
var
min : integer; { интервал в минутах }
  h  : integer; { количество часов }
  m  : integer; { количество минут }
begin
write( 'Введите величину временного интервала в минутах : ');
readln(min);
h := min div 60; m := min mod 60;
writeln(min, ' мин. - это ', h, ' час.', m, ' мин. ');
readln;
end.

```

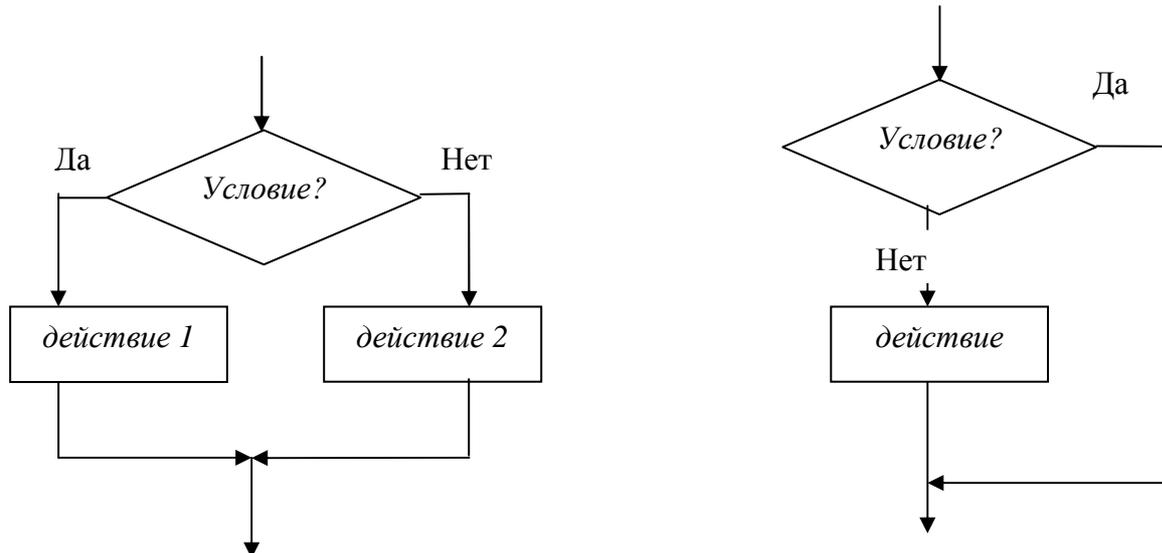
В приведенной программе для определения количества часов использовалась операция целочисленного деления `div`, а для определения минут – операция вычисления остатка от целочисленного деления `mod`.

2.4. Программирование разветвляющихся алгоритмов

На языке Паскаль для записи такого алгоритма используется оператор условных переходов сокращенной формы.

Оператор имеет вид:

if <условие> then <оператор 1> else <оператор 2>;



Простые условия записываются в виде равенств или неравенств. Сложные условия составляются из простых условий с помощью логических операций. Например, сложное условие может выглядеть так:

```

if (a>d) and (c>d) then min:=d else ...;
if (a>d) or (c>d) then min:=d else ...;
if (a>2) and (a<5) or (a>10) and (a<15) then s:=s+1 else s:=0;

```

Приведем пример программы с использованием оператора **if**. Пусть необходимо вывести на экран большее из двух данных чисел. Программа для решения этой задачи имеет вид:

```
program pr1;  
var  x,y:integer;  
begin  
writeln('Введите два числа');  
readln(x,y);  
if x>y then writeln (x)  
  else writeln(y);  
readln;  
end.
```

2.5. Программирование алгоритмов множественного выбора

Такие алгоритмы можно было бы программировать с помощью вложенных операторов условного перехода **if then else**, но в языке Паскаль есть более компактная конструкция – это оператор множественного выбора **case of**. Оператор **case** имеет вид:

```
case <порядковая переменная> of  
  <константа 1> : <оператор 1>;  
  ...  
  <константа n> : <оператор n>;  
else <оператор>  
end;
```

Пример 1. Приведем текст полной программы, переводящей школьные отметки в словесные оценки. Переменная – селектор – должна принимать значения целых чисел: 1, 2, 3, 4, 5.

```
PROGRAM Primer_3;  
  var BALL : Integer;  
BEGIN  
  Write ('Введите величину отметки: '); Read (BALL);  
  WriteLn;  
  Case BALL of { Перевод отметки в оценку }  
    1: WriteLn ('Очень плохо...');  
    2: WriteLn ('Плохо...');  
    3: WriteLn ('Удовлетворительно...');  
    4: WriteLn ('Хорошо!');  
    5: WriteLn ('Отлично!');  
  end;  
END.
```

2.6. Программирование циклических алгоритмов

2.6.1. Оператор for

Данный оператор применяют тогда, когда известно число повторений одного и того же действия. Начальное и конечное значения параметра цикла могут быть представлены константами, переменными или арифметическими выражениями. Оператор имеет две формы:

```
for <параметр> := a to b do <тело цикла>;  
for <параметр> := a downto b do <тело цикла>;
```

где **a** и **b** – величины целого типа, причем **a** – начальное значение, а **b** – конечное значение переменной-параметра.

Рассмотрим, как выполняется оператор

```
for <параметр> := a to b do <тело цикла>;
```

Сначала вычисляются значения выражений **a** и **b**. Если **a** окажется меньшим или равным **b**, то <параметр> последовательно принимает значения, равные **a**, **a+1**, ..., **b-1**, **b**, и для каждого из этих значений выполняется <тело цикла>. Если же **a** > **b**, то <тело цикла> не будет выполнено ни разу, и управление будет передано следующему оператору программы.

Оператор цикла с параметром

```
for <параметр> := a downto b do <тело цикла>;
```

выполняется аналогичным образом, но <параметр> принимает значения, равные **a**, **a-1**, ..., **b+1**, **b**, т.е. с отрицательным шагом (-1).

Если <тело цикла> состоит из нескольких операторов, то операторы тела цикла заключаются в операторные скобки **begin** – **end**. Например:

```
for x:=1 to 10 do  
begin  
    writeln(x);  
    for i:=10 to 100 do y:=y+5  
end;
```

2.6.2. Оператор while

Данный оператор используется для программирования циклических процессов, в которых число повторений неизвестно, но задается некоторое условие его окончания. Это оператор цикла с предусловием.

Оператор имеет вид

```
while <условие> do <тело цикла>;
```

Работа оператора начинается с проверки условия, записанного после слова **while**. Если это условие соблюдается, то выполняется <тело цикла>, а затем вновь проверяется условие и т.д. Как только на очередном шаге окажется, что условие не соблюдается, то выполнение <тела цикла> прекратится. Если <тело цикла> состоит из нескольких операторов, то они объединяются операторными скобками **begin ... end**. В <теле цикла> обязательно должен быть оператор, влияющий на соблюдение условия, в противном случае произойдет заикливание.

2.6.3. Оператор repeat

Для программной реализации циклических процессов с неизвестным числом повторений существует еще один оператор – оператор цикла с послеусловием, он имеет вид

```
repeat  
<тело цикла>  
until <условие>;
```

Данный оператор аналогичен предыдущему оператору цикла, но отличается от него тем, что проверка условия производится после очередного выполнения <тела цикла>. Это обеспечивает его выполнение хотя бы один раз. Сначала выполняется последовательность операторов, входящих в <тело цикла>, после чего проверяется выполнение условия, записанного за служебным словом **until**. Если условие не соблюдается, цикл завершается. В противном случае <тело цикла> повторяется еще раз, после чего снова проверяется соблюдение условия. Естественно <тело цикла> должно содержать оператор, влияющий на условие окончания (продолжения), иначе цикл будет бесконечным.

В качестве примера рассмотрим задачу планирования закупки товаров в магазине на определенную сумму, не превышающую заданную величину. Решение этой задачи можно записать в виде следующей программы:

```
program pг2;  
var c, k, p, s; integer;  
begin  
writeln('Предельная сумма -'); readln(p);  
s:=0;  
repeat  
writeln ('Введите цену товара и его количество');  
readln(c, k);  
s:=s+c*k; writeln ('Стоимость покупки равна', s);  
until s>p;  
writeln('Общая стоимость покупки превысила предельную сумму');  
end.
```

2.7. Структурированные типы данных. Множества

Множество – совокупность однотипных элементов базового типа соответствующая понятию множества в математике с набором свойственных операций.

Массив – совокупность однородных элементов базового типа, обращение к которым выполняется с помощью индексов ординального типа.

Запись – совокупность неоднородных элементов базовых типов, обращение к которым выполняется с помощью имен полей.

Файл – совокупность однородных элементов базового типа, доступ к которым осуществляется последовательно.

Работа с множествами

Переменные множества должны быть описаны следующим предложением:

```
var <имя>: set of <базовый тип>;
```

Например:

```
var s: set of 2..n;  
d: set of 'a'.. 'z';
```

Для множеств определены три операции: «+», «*», «-», которые обозначают соответственно объединение, пересечение и разность множеств. С их помощью можно строить различные выражения множественного типа.

Объединением двух множеств называется множество элементов, принадлежащих обоим множествам. **Пересечением двух множеств** называется множество тех элементов, которые принадлежат одновременно двум множествам. **Разностью двух множеств** называется множество, содержащее те элементы первого множества, которые не являются элементами второго.

В языке Паскаль определены следующие отношения для множеств:

- принадлежность элемента x множеству A ($x \text{ in } A$);
- равенство двух множеств A и B ($A = B$);
- включение множества A в множество B ($A < B$);
- нестрогое включение множества A в множество B ($A \leq B$);
- отношения $>$, \geq , $<$, $>$.

Сведем все операции над множествами в следующую таблицу 2.2.

В Паскале на базовый тип наложено ограничение

$$(\text{Ord}(\text{Min}) \geq 0) \text{ and } (\text{Ord}(\text{Max}) < 256),$$

где Min и Max – минимальное и максимальное значения типа.

a in M	Элемент a принадлежит множеству M
$M1 \leq M2$	M1 содержится в M2
$M1 \supseteq M2$	M1 включает M2
$M1 = M2$	M1 и M2 эквивалентны
$M1 \not\subset M2$	M1 и M2 неэквивалентны
$M1 + M2$	Объединение множеств M1 и M2
$M1 * M2$	Пересечение множеств M1 и M2
$M1 - M2$	Разность множеств M1 и M2 (множество элементов из M1, которых нет в M2)

Пример 1. Программа выводит элементы множества.

```

program WtSet;
const MIN = 1; MAX = 12;
var i: Integer;
      S: set of MIN..MAX;
begin
  S := [2, 5..8];
  for i := MIN to MAX do
    if i in S then Write(i:3);
end.

```

Строковый тип данных

Для обработки строковой информации в Паскаль введен строковый тип данных. Строкой в Паскале называется последовательность из определенного количества символов. Количество символов последовательности называется длиной строки. Синтаксис:

```

var s : string[n];
var s : string;

```

где n – максимально возможная длина строки (целое число символов в диапазоне 1..255).

Если этот параметр опущен, то по умолчанию он принимается равным 255. Строковые константы записываются как последовательности символов, ограниченные апострофами. Допускается формирование строк с использованием записи символов по десятичному коду (в виде комбинации # и кода символа) и управляющих символов (комбинации ^ и некоторых заглавных латинских букв).

Пример 2.

```

'текстовая строка'
#54#32#61
'abcde', ^A^M

```

Пустой символ обозначается двумя подряд стоящими апострофами. Если апостроф входит в строку как литера, то при записи он удваивается.

Переменные, описанные как строковые с разными максимальными длинами, можно присваивать друг другу, хотя при попытке присвоить короткой переменной длинную лишние символы будут отброшены.

Выражения типа `char` можно присваивать любым строковым переменным.

2.8. Структурированные типы данных – массивы.

Одномерные массивы

Массив представляет собой совокупность связанных данных, состоящую из фиксированного числа элементов одного типа, который называется *базовым*. Для определения массива достаточно указать его базовый тип, а также число элементов в массиве и метод их нумерации. К элементу одномерного массива обращаются через индексную переменную, которая записывается как идентификатор массива (имя) с одним индексом в квадратных скобках. Одномерные массивы характеризуются следующим:

- каждый элемент массива может быть явно обозначен и к нему имеется прямой доступ, т.к. индекс указывает на номер элемента в массиве;
- число элементов массива определяется при его описании и в дальнейшем не меняется.

Тип индекса может быть только порядковым (кроме **longint**). Чаще всего используется интервальный тип (диапазон).

Описание типа массива задается следующим образом:

type

имя типа = **array**[*список индексов*] **of** *тип элементов*;

Здесь *имя типа* – идентификатор; *список индексов* – список диапазонов значений индексов или их типов, разделенных запятыми; *тип элементов* – любой тип данных.

Определить переменную как массив можно и непосредственно при ее описании, без предварительного описания типа массива. Например:

```
var a, b, c : array[1..10] of integer;
```

Таким образом, при описании массива задается имя массива, его размер и тип данных.

Массив можно описать через типизированную константу с исходными значениями элементов.

```
Const a : array[1..5] of integer = (13, 44, 45, 64, 20);
```

Если массивы *a* и *b* описаны как:

```
var
  a = array[1..5] of integer;
  b = array[1..5] of integer;
```

то переменные *a* и *b* считаются разных типов.

Для обеспечения совместимости применяйте описание переменных через предварительное описание типа.

```
type
  m_type = array[1..5] of byte;
var mas1, mas2 : m_type;
```

Если типы массивов идентичны, то в программе один массив может быть присвоен другому, например можно записать так: *mas2:=mas1*. В этом случае значения всех переменных одного массива *mas1* будут присвоены соответствующим элементам второго массива *mas2*.

Вместе с тем над массивами не определены операции отношения. Сравнивать два массива можно только поэлементно.

Поскольку *min*, идущий за ключевым словом **of** в описании массива, – любой тип Паскаль, то он может быть и другим массивом. Например:

```
type
  mas = array[1..5] of array[1..10] of integer
```

Такую запись можно заменить более компактной:

```
type
  mas = array[1..5, 1..10] of integer;
```

Таким образом, возникает понятие многомерного массива. Глубина вложенности массивов произвольная, поэтому количество элементов в списке индексных типов (размерность массива) не ограничено, однако не может быть более 65520 байт.

Программы обработки массивов всегда носят циклический характер. Так, например, ввод и вывод данных в массивы осуществляется с использованием цикла, т.к. процесс ввода (вывода) приходится повторять столько раз, сколько элементов содержит массив. Рассмотрим пример программы с вводом и выводом одномерного массива.

```
program pr1;
const      n = 5;
type      mas = array[1..n] of integer;
var
  a : mas;
```

```

    i : byte;
begin
    writeln('введите элементы массива ');
    for i:=1 to n do readln(a[i]);
    writeln('вывод элементов массива: ');
    for i:=1 to n do write(a[i] : 5);
end.

```

Особенно удобен при работе с массивами оператор цикла с известным числом повторений **for**.

Для задания размеров массивов можно применять именованные константы. Ниже приведен пример применения именованной константы и оператора цикла с известным числом повторений для определения суммы элементов массива

```

program SumArr2;
const N = 5;
type TArr = array[1..N] of Real;
var A: TArr;
    i: Integer;
    S: Real;
begin
    Write('A[1..5]: ');
    for i := 1 to N do Read(A[i]);
    S := 0;
    for i := 1 to N do S := S + A[i];
    WriteLn('S = ', S);
end.

```

2.8.1. Поиск минимального (максимального) элемента массива

Алгоритм поиска: делается предположение, что первый элемент массива является минимальным (максимальным), затем остальные элементы массива сравниваются с этим элементом. Если обнаруживается, что проверяемый элемент меньше (больше) принятого за минимальный (максимальный), то этот элемент принимается за минимальный (максимальный) и продолжается проверка оставшихся элементов.

```

M := A[1];
for i := 2 to N do
    if A[i] < M then M := A[i];

```

2.8.2. Вычисление произведения элементов массива

Фрагмент программы достаточно прост:

```

P := 1; for i := 1 to N do P := P * A[i];

```

2.8.3. Выборка элементов массива

Составить программу подсчета количества положительных элементов массива, состоящего из 10 элементов.

Текст программы:

```
program mas3;
var k, i : integer;
a: array[1..10] of integer;
begin
  for i:=1 to 10 do
  begin
    writeln(' Введите элемент массива'); readln(a[i]);
  end;
  writeln(' Элементы исходного массива');
  for i:=1 to 10 do
    write (a[i], ' ');
  writeln;
  k:=0;
  for i:=1 to 10 do
    if a[i]>0 then k:=k+1;
  writeln(' Количество равно',k);
end.
```

Найти значение и номер наибольшего элемента в массиве из 10 целочисленных элементов.

Текст программы:

```
program mas4;
var n, m, i : integer;
a: array[1..10] of integer;
begin
  for i:=1 to 10 do
  begin
    writeln('Введите элемент массива'); readln(a[i]);
  end;
  writeln('Элементы исходного массива');
  for i:=1 to 10 do
    write(a[i], ' ');
  writeln;
  m:=a[1]; n:= 1;
  for i:=2 to 10 do
    if a[i]>m then
      begin      m:= a[i];      n:= i; end;
  writeln('Максимальный элемент равен', m, 'и имеет порядковый номер', n);
end.
```

2.8.4. Сортировка массива

Под сортировкой массива подразумевается процесс перестановки его элементов с целью упорядочивания их в соответствии с каким-либо критерием. Сортировка проводится на основании сравнения значений элементов массива друг с другом. Можно сравнивать значения целых, вещественных, символьных, строчных и производных от них типов.

Сортировка методом прямого выбора. Алгоритм сортировки массива по возрастанию методом прямого выбора:

1. Просматривая массив от первого элемента, найти минимальный и поместить его на место первого элемента, а первый – на место минимального.
2. Просматривая массив от второго элемента, найти минимальный и поместить его на место второго элемента, а второй – на место минимального.
3. И так далее до последнего элемента.

Для демонстрации рассмотрим программу сортировки массива целых чисел по возрастанию (для контроля выполнения алгоритма программа выводит массив после каждого обмена элементами).

```
program sort1;
  Const  size=5;
  Var
    a: array[1..size] of integer;
    i: integer; {номер элемента, от которого ведется по-
иск минимального элемента}
    min: integer; {номер минимального элемента в части
массива от i до верхней границы массива}
    j: integer; {номер элемента сравниваемого с минимальным}
    buf: integer; {буфер, используемый при обмене элемен-
тов массива}
    k: integer;
begin
  writeln('Сортировка массива');
  write('Введите через пробел', size:3,'целых в одной
строке и нажмите клавишу ввод');
  for k:=1 to size do read(a[k]);
  writeln('Сортировка');
  for i:=1 to size-1 do
    begin
      min:=i;
      for j:=i+1 to size do
        begin
```

```

    if a[j]<a[min] then min:=j;
end;
buf:=a[i]; a[i]:=a[min]; a[min]:=buf;
for k:=1 to size do write(a[k], ' ');
writeln;
end; end.

```

Сортировка методом прямого обмена. В основе алгоритма лежит обмен соседних элементов массива. Каждый элемент массива, начиная с первого, сравнивается со следующим, и если он больше следующего, то элементы меняются местами. Таким образом, элементы с меньшим значением подвигаются к началу массива (всплывают), а элементы с большим значением – к концу массива (тонут), поэтому этот метод иногда называют «пузырьковым». Этот процесс повторяется на единицу меньше раз, чем элементов в массиве. Ниже представлена программа сортировки массива целых чисел по возрастанию. Для демонстрации процесса сортировки программа выводит массив после каждого цикла обменов.

```

program sort2;
    const
        size=5;
    Var
        a: array[1..size] of integer;
        i: integer; {счетчик циклов}
        buf: integer;
        k: integer; {текущий индекс элемента массива}
begin
    writeln('Сортировка массива методом пузырька');
    write('Введите через пробел', size:3, 'целых в одной
строке и нажмите клавишу ввод');
    for k:=1 to size do read(a[k]);
    writeln('Сортировка');
    for i:=1 to size-1 do
        begin
            for k:=1 to size-1 do
                begin
                    if a[k]>a[k+1] then
                        begin
                            buf:=a[k];
                            a[k]:=a[k+1];
                            a[k+1]:=buf;
                        end;
                end;
            end;
        end;
end;

```

```

        for k:=1 to size do write(a[k], ' ');
        writeln;
    end;
end.

```

2.8.5. Поиск в массиве

Поиск осуществляется последовательным сравнением элементов массива с образцом до тех пор, пока не будет найден элемент, равный образцу, или не будут проверены все элементы. Алгоритм простого перебора применяется, если элементы массива не упорядочены.

Рассмотрим программу поиска в массиве целых чисел. Перебор элементов массива осуществляет инструкция REPEAT, в теле которой инструкция IF сравнивает текущий элемент массива с образцом и присваивает переменной `naiden` значение `true`, если текущий элемент равен образцу. Цикл завершается, если в массиве обнаружен элемент равный образцу, или если проверены все элементы массива. По завершении цикла по значению переменной `found` можно определить, успешен поиск или нет.

```

program poisk;
var
    massiv: array[1..10] of integer; {массив целых}
    obrazec: integer; {образец для поиска}
    naiden: boolean; {признак совпадения с образцом}
    i: integer;
begin
    writeln('Введите через пробел 10 целых элементов массива и нажмите клавишу ввода');
    for i:=1 to 10 do
        read(massiv[i]);
        write('Введите образец для поиска');
        readln(образec);
        naiden:=false;
        i:=1; {проверяем с первого элемента массива}
        repeat
            if massiv[i]=образec then naiden:=true else i:=i+1;
        until (i>10)or(naiden);
        if naiden then writeln('Совпадение с элементом',i:3,'поиск завершен')
        else writeln('Совпадений с образцом нет');
    end.

```

2.9. Подпрограммы, пользовательские процедуры и функции

Структура описания процедур и функций до некоторой степени похожа на структуру Паскаль-программы: у них также имеются заголовок, раздел описаний и исполняемая часть. Раздел описаний содержит те же подразделы, что и раздел описаний программы: описания констант, типов, меток, процедур, функций, переменных. Исполняемая часть содержит собственно операторы процедур.

Формат описания процедуры имеет вид:

```
procedure имя процедуры (формальные параметры) ;  
    раздел описаний процедуры  
begin  
    исполняемая часть процедуры  
end;
```

Формат описания функции:

```
function имя функции (формальные параметры): тип результата;  
    раздел описаний функции  
begin  
    исполняемая часть функции  
    имя функции: =результат;  
end;
```

Формальные параметры в заголовке процедур и функций записываются в виде:

```
var имя параметра: имя типа;
```

и отделяются друг от друга точкой с запятой. Ключевое слово **var** может отсутствовать для входных параметров, а для *выходных должен быть обязательно*. Если параметры однотипны, то их имена можно перечислять через запятую, указывая общее для них имя типа. При описании параметров можно использовать только стандартные имена типов либо имена типов, определенные с помощью команды **type**, вызывающей программы. Список формальных параметров может отсутствовать.

Вызов процедуры производится оператором, имеющим следующий формат:

```
имя процедуры (список фактических параметров);
```

Список фактических параметров – это их перечисление через запятую. При вызове фактические параметры как бы подставляются вместо

формальных, стоящих на тех же местах в заголовке. Таким образом, происходит передача входных параметров, затем выполняются операторы исполняемой части процедуры, после чего происходит возврат в вызывающий блок. Передача выходных параметров происходит непосредственно во время работы исполняемой части.

Вызов функции в Паскаль может производиться аналогичным способом, кроме того, имеется возможность осуществить вызов внутри какого-либо выражения. В частности, имя функции может стоять в правой части оператора присваивания, в разделе условий оператора **if** и т.д.

Для передачи в вызывающий блок выходного значения функции в исполняемой части функции перед возвратом в вызывающий блок необходимо поместить следующую команду:

имя *функции* := результат;

При вызове процедур и функций необходимо соблюдать следующие правила:

- количество фактических параметров должно совпадать с количеством формальных;
- соответствующие фактические и формальные параметры должны совпадать по порядку следования и по типу.

Процедуры и функции должны обладать определенной независимостью в смысле использования переменных, типов и констант. В связи с этим возникает необходимость разделять данные на *глобальные* и *локальные* по их доступности из других подпрограмм.

Глобальные – это те константы, типы и переменные, которые объявлены в программе вне описания процедуры и функции и до их описания. Они доступны вызывающей программе и вызываемой подпрограмме.

Локальные константы, типы и переменные существуют только внутри подпрограммы, в которой они описаны через разделы объявлений или через список формальных параметров. Они доступны только подпрограмме, в которой они описаны.

Рассмотрим использование процедуры на примере программы поиска максимума из двух целых чисел.

```
var   x,y,m,n : integer;  
procedure MaxNumber (a,b: integer; var max: integer);  
  {в заголовке процедуры a,b – входные параметры, max – выходной параметр};  
  begin {начало процедуры}  
    if a>b then max := a else max := b;
```

```

end; {конец процедуры}
begin {начало основной программы}
write ('Введите x,y ');
readln(x,y);
MaxNumber(x,y,m); {формальный параметр a получает
значение x, b - значение y при вызове процедуры, m - значение
результата из процедуры через параметр max};
MaxNumber(2,x+y,n);
writeln('m=',m,'n=',n);
end. {Конец основной программы}

```

Аналогично задачу, но уже с использованием функций, можно решить так:

```

var x,y,m,n : integer;
function MaxNumber(a,b: integer) : integer;
var max: integer;
begin {начало функции}
if a>b then max := a else max := b;
MaxNumber := max;
{имя функции получает значение результата целого типа};
end; {конец функции}
begin {начало основной программы}
write('Введите x,y ');
readln(x,y);
m := MaxNumber(x,y);
{в основной программе результат функции копируется в переменную m};
n := MaxNumber(2, x+y);
writeln('m=',m,'n=',n);
end.

```

Рекурсивные подпрограммы. Рекурсия – это способ организации вычислительного процесса, при котором подпрограмма обращается сама к себе, но с более «простыми» параметрами (измененными). Пример вычисления факториала числа ($N! = N(N-1)!$) с рекурсивным вызовом функции:

```

program recl;
function fact (n : word) : word;
begin if n = 0 then fact := 1 else fact := n * fact (n-1); end;
begin
write('Введите число '); readln(n); write(' Факториал числа ',n,' равен: ',fact(n))
end.

```

В практических задачах глубина рекурсивных вызовов должна быть конечна и невелика. Большая глубина рекурсивных вызовов приводит к переполнению стека. Кроме того, подобно операторам цикла, рекурсивные вызовы могут приводить к «зацикливанию». Для того чтобы избежать этого, в теле процедуры или функции должно быть некоторое управляющее условие с ветвью не содержащей рекурсивный вызов. Эта ветвь будет определять ограничение рекурсии (граничное условие).

Рекурсивные процедуры эффективны при работе со сложными динамическими структурами данных: списками, деревьями, графами.

2.10. Работа с двумерными массивами

2.10.1. Действия над элементами массива

При работе с двумерным массивом указываются два индекса, например $V[4,4]$. Индексированные элементы массива называются индексными переменными и могут быть использованы так же, как и простые переменные. Например, они могут находиться в выражениях в качестве операндов, использоваться в операторах `for`, `while`, `repeat`, входить в качестве параметров в операторы `read`, `readln`, `write`, `writeln`; им можно присваивать любые значения, соответствующие их типу.

Копированием массивов называется присвоение значений всех элементов одного массива всем соответствующим элементам другого массива. Копирование можно выполнить одним оператором присваивания, например $A:=B$, если массивы идентичны по структуре, или с помощью оператора `for` поэлементно:

```
For i:=1 to 5 do  
  For j:=1 to 7 do  
    A[i,j]:=B[i,j];
```

2.10.2. Действия над идентичными двумерными массивами

Выражение	Результат
$A=B$	True, если значение каждого элемента массива A равно соответствующему значению элемента массива B.
$A<>B$	True, если хотя бы одно значение элемента массива A не равно значению соответствующего элемента массива B.
$A:=B$	Все значения элементов B присваиваются соответствующим элементам массива A. Значения элементов массива B остаются неизменными.

2.10.3. Использование процедуры ввода и процедуры вывода элементов двумерного массива

```
program P_r_i_m_e_r_2;
  const M = 2; { Количество строк в массиве }
        N = 3; { Количество столбцов в массиве }
  type T = Array [1..M,1..N] of Integer;
  var A : T;
procedure Wwod_Mass (var S: T);
  { Процедура ввода с клавиатуры элементов массива }
  { процедура с параметрами }
  { S - параметр, вызываемый по адресу }
  var i,j: Integer;
begin
  WriteLn ('Введите массив: ');
  For i:=1 to M do
    For j:=1 to N do
      begin Write ('Введите S['',i','',',j, ']: ');
ReadLn (S[i,j]) end
    END;
procedure Wywod_Mass (S: T);
  { Процедура вывода на экран элементов массива }
  { процедура с параметрами }
  { S - параметр, вызываемый по значению }
  var i,j: Integer;
BEGIN
  WriteLn ('Вот Ваш массив: ');
  For i:=1 to M do
    begin For j:=1 to N do Write (S[i,j], ' ');
WriteLn end
    end;
BEGIN
  Wwod_Mass (A); {Вызов процедуры ввода элементов массива}
  Wywod_Mass(A) {Вызов процедуры вывода элементов массива}
  end.
```

2.11. Файловый тип данных

Описание типа **Var** <имя файловой переменной>: **file of** <Имя базового типа (типа элементов файла)>

Имя переменной файлового типа принято называть *логическим именем файла*. Именно это имя используется в программе при обращении к его элементам. Кроме того, у файла есть имя, под которым он фигурирует

в ОС, – *физическое имя*. Установление связи «логическое имя – физическое имя» выполняется специальной процедурой (Assign). Одно логическое имя можно связывать в разное время с разными физическими, а это означает, что программу можно написать так, что ее не надо будет переделывать при смене имени файла в ОС или при работе с другим файлом.

Операции с файлами реализуются обращением к стандартным процедурам:

Assign(F, FName)	Связывает логическое имя файла F с его физическим именем FName
Reset(F)	Открывает файл F для ввода
Rewrite(F)	Открывает файл F для вывода
Close(F)	Закрывает файл F
Read(F,R)	Чтение элемента файла F в переменную R
Write(F,R)	Запись элемента файла A из переменной R
Eof(F)	Возвращает булевское значение "Конец файла"
Eof(F)	Возвращает булевское значение "Конец файла" для файла F
Eoln(F)	Возвращает булевское значение "Конец строки" для файла F

Рассмотрим пример, иллюстрирующий работу с текстовым файлом. Пусть требуется оценить объем текста рукописи, записанного на диск в так называемых издательских листах, которые вмещают по 40 000 символов. В программе реализуем идею подсчета общего количества символов в тексте рукописи, пока не дойдем до конца файла, а затем при выводе ответа накопленное количество поделим на 40 000.

```

program Rukopis;
var FTxt: Text;
    s: String;
    sum: LongInt;

begin
  Write('Введите имя файла: '); Readln(s);
  Assign(FTxt, s); Reset(FTxt);
  Sum := 0;
  while not Eof(FTxt) do
    begin
      Readln(FTxt, s);
      Inc(sum, Length(s));
    end;
  Close(FTxt);
  Writeln('Объем = ', sum/40000:6:2, ' издат. листов');
end.

```

2.12. Стандартный модуль Graph

Модуль Graph предназначен для работы с монитором в графическом режиме. В этом режиме весь экран монитора представляется в виде прямоугольной матрицы графических элементов – *пикселей*.

В модуле Graph положение на экране отдельного пикселя задается его целочисленными координатами в системе координат (СК) экрана. Начало этой СК находится в левом верхнем углу экрана. Ось Ox проходит горизонтально от начала СК вправо, а ось Oy – вертикально от начала СК вниз.

Кроме положения на экране каждый пиксель характеризуется цветом. Цвет пикселя кодируется целым числом из диапазона 0..15 (например, 1 – голубой, 2 – зеленый, 4 – красный). Кроме того, существует понятие «цвет фона», который кодируется целым числом из диапазона 0..7.

Интерфейс модуля Graph достаточно велик и сложен. Для первого знакомства с графическим вводом-выводом нам будет достаточно небольшой его части.

unit Graph;

...

procedure arc(x, y, un, uk, r) – вычерчивает дугу окружности радиусом r с центром в точке с координатами (x, y) от начального угла un до конечного угла uk ;

procedure bar($x1, y1, x2, y2$) – вычерчивает столбец по указанным координатам точек;

procedure ClearDevice; {Заполняет экран цветом фона}

procedure CloseGraph; {Закрывает графический режим}

procedure fillellipse(x, y, xr, yr) – вычерчивает эллипс с центром в точке (x, y) и радиусами xr и yr ;

function GetMaxX:Integer; {Возвр.макс.значение координаты X}

function GetMaxY:Integer; {Возвр.макс.значение координаты Y}

procedure SetColor(Color:Word); {Устанавливает цвет линий и литер текста}

procedure SetBkColor(Color:Word); {Уст. цвет фона}

procedure Line($X1, Y1, X2, Y2$:Integer); {Проводит линию}

procedure Circle(X, Y :Integer; Radius:Word); {Рисует окружность}

procedure Rectangle($X1, Y1, X2, Y2$:Integer); {Рисует прямоугольник}

procedure PutPixel(X, Y :Integer; Color:Word); {Выводит пиксел указанного цвета}

procedure OutTextXY(X, Y :Integer; TextString:string); {Выводит текст с указанной позиции}

Рассмотрим взаимосвязь некоторой заданной СК с СК экрана. В заданной СК с помощью четырех параметров $xmin$, $xmax$, $ymin$ и $ymax$ определим прямоугольную область. Рассмотрим задачу отображения этой прямоугольной области на экран монитора. Исходя из предположения, что угловые точки отображаемой области должны отобразиться в угловые пиксели графической матрицы экрана, получим следующие соотношения:

$$mx = GetMaxX/(xmax - xmin), \quad my = GetMaxY/(ymax - ymin),$$

$$xs = Round((-xmin+x)*mx), \quad ys = Round((ymax-y)*my),$$

где x , y – координаты некоторой точки в отображаемой СК, xs , ys – координаты отображения точки в СК экрана, mx и my – масштабы преобразования по соответствующим осям.

Эти соотношения использованы в следующем примере программы с применением модуля Graph. Программа Curve.pas строит график функции $y = \sin(x)$ на отрезке $(-2*Pi, 2*Pi)$.

```

program Curve;
uses Graph;
const
    xmin = -6.5; xmax = 6.5; { Размеры заданной          }
    ymin = -1.5; ymax = 1.5; { области                }
    a = -2*Pi;   b = 2*Pi;   { Отрезок построения графика }
    n = 50;     { Кол. интервалов графика      }
    h = (b - a)/n; { Длина одного интервала    }
var
    mx,my: Real;           { Масштабы                }
    x,y:   Real;           { Текущая точка кривой в зад. СК }
    xs,ys: Integer;       { Текущая точка графика а СК экрана }
    xc,yc: Integer;       { Начало заданной СК в СК экрана   }
    i:     Integer;       { Номер точки графика            }

procedure GrInit; {Иницирует графическую систему }
var   GraphDriver, GraphMode, ErrorCode: integer;
begin
    GraphDriver:=Detect; {автоопределение типа адаптера}
    InitGraph(GraphDriver,GraphMode, '');
    ErrorCode:=GraphResult;{обработка ошибок инициализации}
if ErrorCode<>grOk then
    begin
        WriteLn('Ошибка графики: ',GraphErrorMsg(ErrorCode));
        WriteLn('Программа остановлена. ');
        ReadLn;
        Halt(1);
    end
end; { GrInit }

```

```

procedure ToScr(x,y:Real; var xs,ys:Integer);
    { Отображает точку из заданной СК в СК экрана }
begin
xs:=Round((-xmin+x)*mx); ys:=Round((ymax-y)*my);
end; { ToScr }

begin
GrInit; { Иницилируем графическую систему }
    { Определяем масштабы по осям 0X и 0Y}
mx := GetMaxX/(xmax - xmin); my := GetMaxY/(ymax - ymin);
    { Определяем координаты начала заданной СК в СК экрана }
ToScr(0,0,xs,ys);
SetColor(1); { Устанавливаем цвет линий }
SetBkColor(7); { Устанавливаем цвет фона }
Line(0,ys,GetMaxX,ys); { Проводим ось 0X }
Line(xs,0,xs,GetMaxY); { Проводим ось 0Y }
OutTextXY(20,20,'Y = Sin(X)'); { Выводим заголовок графика }
    { РИСУЕМ ГРАФИК }
SetColor(12);
x:=a; y:=Sin(x);
ToScr(x,y,xs,ys);
MoveTo(xs,ys);
for i:=1 to n do
    begin
        x:=x+h; y:=Sin(x);
        ToScr(x,y,xs,ys);
        LineTo(xs,ys);
    end;
ReadLn; { Для выхода из гр. режима надо нажать Enter }
CloseGraph; { Возвращаемся в алфавитно-цифровой режим }
end.

```

3. КОМПЬЮТЕРНЫЕ СЕТИ

3.1. Назначение компьютерных сетей

При физическом соединении двух или более компьютеров образуется компьютерная *сеть*. В общем случае для создания компьютерных сетей необходимо специальное аппаратное обеспечение (*сетевое оборудование*) и специальное ПО (*сетевые программные средства*). Простейшее соединение двух компьютеров для обмена данными называется *прямым соединением*.

Назначение всех компьютерных сетей состоит в обеспечении совместного доступа к общим *ресурсам*. В зависимости от назначения сети в термин «ресурс» можно вкладывать различный смысл. Ресурсы бывают трех типов: *аппаратные*, *программные* и *информационные*. Например, устройство печати (принтер), емкость жестких дисков – это аппаратный ресурс. Когда все участники небольшой компьютерной сети пользуются одним общим принтером, это значит, что они разделяют общий аппаратный ресурс. То же можно сказать и о сети, имеющей один компьютер с увеличенной емкостью жесткого диска (*файловый сервер*), на котором все участники сети хранят свои архивы и результаты работы.

Кроме аппаратных ресурсов компьютерные сети позволяют совместно использовать *программные ресурсы*. Так, например, для выполнения очень сложных и продолжительных расчетов можно подключиться к удаленной большой ЭВМ и отправить вычислительное задание на нее, а по окончании расчетов точно так же получить результат обратно.

Данные, хранящиеся на удаленных компьютерах, образуют *информационный ресурс*. Роль этого ресурса видна наиболее ярко на примере Internet, который воспринимается, прежде всего, как гигантская информационно-справочная система.

При работе в компьютерной сети любого типа одновременно происходит совместное использование всех типов ресурсов. Так, например, обращаясь в Internet за какой-либо информацией, мы используем аппаратные средства, на которых работают программы, обеспечивающие поставку запрошенных нами данных.

3.2. Локальные и глобальные сети. Основные понятия

Для обеспечения необходимой совместимости как по аппаратуре, так и по программам в компьютерных сетях действуют специальные стандарты, называемые *протоколами*. Они определяют характер аппаратного взаимодействия компонентов Сети (*аппаратные протоколы*) и характер

взаимодействия программ и данных (*программные протоколы*). Физически функции поддержки протоколов исполняют аппаратные устройства (*интерфейсы*) и программные средства (*программы поддержки протоколов*). Программы, выполняющие поддержку протоколов, часто также называют *протоколами*.

Так, например, если два компьютера соединены между собой прямым соединением, то на физическом уровне протокол их взаимодействия определяют конкретные устройства физического порта (параллельного или последовательного) и механические компоненты (разъемы, кабель и т.п.). На более высоком уровне взаимодействие между компьютерами определяют программные средства, управляющие передачей данных. На самом высоком уровне протокол взаимодействия обеспечивают приложения ОС.

В соответствии с используемыми протоколами компьютерные сети принято разделять на локальные (*LAN – Local Area Network*) и глобальные (*WAN – Wide Area Network*). Компьютеры *локальной сети* преимущественно используют единый комплект протоколов для всех участников. По территориальному признаку локальные сети отличаются компактностью. Они могут объединять компьютеры одного помещения, этажа, здания, группы компактно расположенных сооружений. *Глобальные сети* имеют, как правило, увеличенные географические размеры. Они могут объединять как отдельные компьютеры, так и отдельные локальные сети, в том числе и использующие различные протоколы.

Группы сотрудников, работающих над одним проектом в рамках локальной сети, называются *рабочими группами*. В рамках одной локальной сети могут работать несколько рабочих групп. У участников рабочих групп могут быть разные права для доступа к общим ресурсам Сети. Совокупность приемов разделения и ограничения прав участников компьютерной сети называется *политикой Сети*. Управление сетевыми политиками (их может быть несколько в одной сети) называется *администрированием сети*. Лицо, управляющее организацией работы участников локальной компьютерной сети, называется *системным администратором*.

Создание локальных сетей характерно для отдельных предприятий или отдельных подразделений предприятий. Если предприятие (или отрасль) занимает обширную территорию, то отдельные локальные сети могут объединяться в Глобальные сети. В этом случае локальные сети связывают между собой с помощью любых традиционных каналов связи (кабельных, спутниковых, радиорелейных и т.п.).

Для связи между собой нескольких локальных сетей, работающих по разным протоколам, служат специальные средства, называемые *шлюзами*.

Шлюзы могут быть как аппаратными, так и программными. Например, это может быть специальный компьютер (*шлюзовой сервер*) или компьютерная программа (*шлюзовое приложение*). В последнем случае компьютер может выполнять не только функцию шлюза, но и какие-то иные функции, типичные для рабочих станций.

При подключении локальной сети предприятия к Глобальной сети важную роль играет понятие *сетевой безопасности*. В частности, должен быть ограничен доступ в локальную сеть для посторонних лиц извне, а также ограничен выход за пределы локальной сети для сотрудников предприятия, не имеющих соответствующих прав. Для обеспечения сетевой безопасности между локальной и глобальной сетью устанавливают так называемые *брандмауэры*. Брандмауэром может быть специальный компьютер или компьютерная программа, препятствующая несанкционированному перемещению данных между сетями.

3.3. Краткая история развития Internet

Internet – это Глобальная компьютерная сеть, состоящая из множества соединенных друг с другом меньших по размеру сетей и покрывающая весь земной шар. Internet можно рассматривать в физическом смысле как миллионы компьютеров, связанных друг с другом всевозможными линиями связи, однако такой «физический» взгляд на Internet слишком узок. Лучше рассматривать Internet как некое информационное пространство.

Ранние эксперименты по передаче и приему информации с помощью компьютеров начались еще в 50-х годах и имели лабораторный характер. В США решение о создании первой Глобальной сети национального масштаба было принято в 1958 г.

Поводом для создания Глобальной компьютерной сети стала разработка Пентагоном глобальной системы раннего оповещения о пусках ракет (*NORAD – North American Aerospace Defence Command*). Станции системы *NORAD* протянулись через север Канады от Аляски до Гренландии, а подземный командный центр расположился вблизи города Колорадо-Спрингс в недрах горы Шайенн. Центр управления был введен в действие в 1964 г., и, собственно, с этого времени можно говорить о работе первой Глобальной компьютерной сети, хотя и ведомственной. С середины 60-х годов к ней стали подключаться авиационные, метеорологические и другие военные и гражданские службы.

Курированием работы Сети занималась специальная организация – Управление перспективных разработок министерства обороны США

(*DARPA – Defense Advanced Research Project Agency*). Основным недостатком централизованной Сети была недостаточная устойчивость, связанная с тем, что при выходе из строя какого-либо из узлов полностью выходил из строя и весь сектор, находившийся за ним, а при выходе из строя центра управления выходила из строя вся Сеть. Во времена ядерного противостояния сверхдержав этот недостаток был критичным.

Решение проблемы устойчивости и надежности Сети было поручено управлению *DARPA*. Основными направлениями исследований стали поиск новых протоколов обслуживания Сети и новых принципов сетевой архитектуры. Полигоном для испытаний новых принципов стали крупнейшие университетские и научные центры США, между которыми были проложены линии компьютерной связи. Со стороны министерства обороны работы курировались тем же управлением *DARPA*, и первая вневедомственная национальная компьютерная сеть получила название *ARPANET*. Ее внедрение состоялось в 1969 г.

В 70-е годы сеть *ARPANET* медленно развивалась. В основном развитие происходило за счет подключения региональных сетей, воссоздающих общую архитектуру *ARPANET* на более низком уровне (в региональном или локальном масштабе). Основной объявленной задачей *ARPANET* стала координация групп коллективов, работающих над едиными научно-техническими проектами, а основным назначением стал обмен электронной почтой и файлами с научной и проектно-конструкторской документацией. В то же время не прекращались работы над основной необъявленной задачей – разработкой новых сетевых протоколов, способных обеспечить живучесть Глобальной сети даже в ядерном конфликте.

Второй датой рождения Internet принято считать 1983 г. В этом году произошли революционные изменения в ПО компьютерной связи. Проблема устойчивости Глобальной сети была решена внедрением протокола *TCP/IP*, лежащего в основе Всемирной сети и по нынешний день. Решив, наконец, эту задачу, управление *DARPA* прекратило свое участие в проекте и передало управление Сетью Национальному научному фонду (*NSF*). Так в 1983 г. образовалась Глобальная сеть *NSFNET*. В середине 80-х годов к ней начали активно подключаться академические и научные сети других стран, например академическая сеть Великобритании *JANET* (*Joined Academic Network*).

Во второй половине 80-х годов произошло деление Всемирной сети на домены по принципу принадлежности. Домен *gov* финансировался на средства правительства, домен *sci* – на средства научных кругов, домен *edu* – на средства системы образования, а домен *com* (коммерческий) не

финансировался никем, т.е. его узлы должны были развиваться за счет собственных ресурсов. Национальные сети других государств стали рассматриваться как отдельные домены. Например, *uk* – домен Великобритании, *by* – домен Беларуси, *ru* – домен России.

Когда во второй половине 80-х годов сложилась и заработала система доменных имен (*DNS, Domain Name System*), Национальный научный фонд США утратил контроль над развитием Сети. Тогда и появилось понятие «Internet» как саморазвивающейся децентрализованной иерархической структуры.

Стек протоколов TCP/IP. В техническом понимании *TCP/IP* – это не один сетевой протокол, а два протокола, лежащих на разных уровнях (это так называемый *стек протоколов*). Протокол *TCP* – протокол *транспортного уровня*. Он управляет тем, как происходит передача данных. Протокол *IP* – адресный. Он принадлежит *сетевому уровню* и определяет, куда происходит передача.

Протокол TCP. Согласно протоколу *TCP (Transmission Control Protocol)* отправляемые данные «нарезаются» на небольшие пакеты, после чего каждый пакет маркируется таким образом, чтобы в нем были данные, необходимые для правильной сборки документа на компьютере получателя.

Протокол IP. Суть адресного протокола *IP (Internet Protocol)* состоит в том, что у каждого участника Всемирной сети должен быть свой уникальный адрес (*IP-адрес*). Без этого нельзя говорить о точной доставке *TCP*-пакетов на нужное рабочее место. Этот адрес выражается четырьмя байтами, например 195.38.46.11. Структура *IP*-адреса организована так, что каждый компьютер, через который проходит какой-либо *TCP*-пакет, может по этим четырем числам определить, кому из ближайших «соседей» надо переслать пакет, чтобы он оказался «ближе» к получателю. В результате конечного числа перебросок *TCP*-пакет достигает адресата.

Слово «ближе» не случайно взято в кавычки, в данном случае оценивается не географическая близость. В расчет принимаются условия связи и пропускная способность линии. Два компьютера, находящиеся на разных континентах, но связанные высокопроизводительной линией космической связи, считаются более «близкими» друг к другу, чем два компьютера из соседних поселков, связанные простым телефонным проводом. Решением вопросов, что считать «ближе», а что «дальше», занимаются специальные средства – *маршрутизаторы*. Роль маршрутизаторов в Сети обычно выполняют специализированные компьютеры, но это могут быть и специальные программы, работающие на узловых серверах Сети.

3.4. Службы Internet

Когда говорят о работе в Internet или об использовании Internet, то на самом деле речь идет не об Internet в целом, а только об одной или нескольких из его многочисленных служб. В зависимости от конкретных целей и задач клиенты Сети используют те службы, которые им необходимы.

В простейшем понимании *служба* – что пара программ, взаимодействующих между собой согласно определенным правилам, называемым *протоколами*. Одна из программ этой пары называется *сервером*, а вторая – *клиентом*. Соответственно, когда говорят о работе служб Internet, речь идет о взаимодействии серверного оборудования и ПО и клиентского оборудования и ПО.

Так, например, для передачи файлов в Internet используется специальный прикладной протокол *FTP (File Transfer Protocol)*. Соответственно, чтобы получить из Internet файл, необходимо:

- иметь на компьютере программу, являющуюся клиентом *FTP (FTP-клиент)*;
- установить связь с сервером, предоставляющим услуги *FTP (FTP-сервером)*.

3.4.1. Telnet

Исторически одной из ранних является служба удаленного управления компьютером *Telnet*. Подключившись к удаленному компьютеру по протоколу этой службы, можно управлять его работой. Такое управление еще называют *консольным* или *терминальным*. В прошлом эту службу широко использовали для проведения сложных расчетов на удаленных вычислительных центрах. Так, например, если для очень сложных вычислений на персональном компьютере требовались недели непрерывной работы, а на удаленной супер-ЭВМ всего несколько минут, то ПК применяли для удаленного ввода данных в ЭВМ и для приема полученных результатов.

В наши дни в связи с быстрым увеличением мощности ПК необходимость в подобной услуге сократилась, но тем не менее службы *Telnet* в Internet продолжают существовать. Часто протоколы *Telnet* применяют для дистанционного управления техническими объектами (например, телескопами, видеокамерами, промышленными роботами, автоматизированными складами и даже торговыми автоматами).

3.4.2. Электронная почта (E-Mail)

Эта служба также является одной из наиболее ранних. Ее обеспечением в Internet занимаются специальные почтовые серверы.

Почтовые серверы получают сообщения от клиентов и пересылают их по цепочке к почтовым серверам адресатов, где эти сообщения накапливаются. При установлении соединения между адресатом и его почтовым сервером происходит автоматическая передача поступивших сообщений на компьютер адресата.

Почтовая служба основана на двух прикладных протоколах: *SMTP* и *POP3*. По первому происходит отправка корреспонденции с компьютера на сервер, а по второму – прием поступивших сообщений. Существует большое разнообразие клиентских почтовых программ. К ним относится, например, программа *Microsoft Outlook Express*. Из специализированных почтовых программ хорошую популярность имеют программы *The Bat!*, *Eudora*, *Pegasus mail*.

3.4.3. Служба телеконференций (Usenet)

Служба телеконференций похожа на циркулярную рассылку электронной почты, в ходе которой одно сообщение отправляется не одному корреспонденту, а большой группе (такие группы называются *телеконференциями* или *группами новостей*).

Обычное сообщение электронной почты пересылается по узкой цепочке серверов от отправителя к получателю. При этом не предполагается его хранение на промежуточных серверах. Сообщения, направленные на сервер группы новостей, отправляются с него на все серверы, с которыми он связан, если на них данного сообщения еще нет. Далее процесс повторяется.

На каждом из серверов поступившее сообщение хранится ограниченное время (обычно неделю), и все желающие могут в течение этого времени с ним ознакомиться.

Вся система телеконференций разбита на тематические группы. Сегодня в мире насчитывают порядка 50 000 тематических групп новостей. Они охватывают большинство тем, интересующих массы.

Основной прием использования групп новостей состоит в том, чтобы задать вопрос, обращаясь ко всему миру, и получить ответ или совет от тех, кто с этим вопросом уже разобрался. При этом важно следить за тем, чтобы содержание вопроса соответствовало теме данной телеконференции. При отправке сообщений в телеконференции принято указывать свой адрес электронной почты для обратной связи.

Для работы со службой телеконференций существуют специальные клиентские программы. Так, например, приложение *Microsoft Outlook Express*, указанное выше как почтовый клиент, позволяет работать также и со службой телеконференций. Для начала работы надо настроить программу на взаимодействие с сервером групп новостей, оформить «подписку» на определенные группы и периодически, как и электронную почту, получать все сообщения, проходящие по теме этой группы. В данном случае слово «подписка» не предполагает со стороны клиента никаких обязательств или платежей – это просто указание серверу о том, что сообщения по указанным темам надо доставлять, а по прочим – нет. Отменить подписку или изменить ее состав можно в любой удобный момент.

3.4.4. Служба World Wide Web (WWW)

World Wide Web – это единое *информационное пространство*, состоящее из взаимосвязанных электронных документов, хранящихся на *Web-серверах*.

Web-страница. Отдельные документы, составляющие *пространство Web* называют *Web-страницами*. Обычно это комбинированный документ, который может содержать текст, графические иллюстрации, мультимедийные и другие объекты. Для создания Web-страниц используется язык *HTML (HyperText Markup Language – язык гипертекстовой разметки)*, который при помощи вставленных в документ *тегов* описывает логическую структуру документа, управляет форматированием текста и размещением различных объектов. От обычного текста теги отличаются тем, что заключены в угловые скобки. Большинство тегов используются парами: открывающий тег и закрывающий. Закрывающий тег начинается с символа «/»:

```
<CENTER> Этот текст должен выравниваться по центру экрана </CENTER>  
<P ALIGN = "LEFT"> Этот текст выравнивается по левой границе экрана </P>  
<P ALIGN = "RIGHT"> Этот текст выравнивается по правой границе экрана </P>
```

Сложные теги имеют кроме ключевого слова дополнительные *атрибуты* и *параметры*, детализирующие способ их применения. Таким образом, Web-документ представляет собой обычный текстовый документ, размеченный тегами *HTML*. Такие документы также называют *HTML-документами* или документами в формате *HTML*. При просмотре *HTML-документа* на экране с помощью *браузера* (см. ниже) теги не отображаются, и пользователь видит только текст, составляющий документ.

Web-узел. Группы тематически объединенных Web-страниц называют *Web-узлами*. Один физический Web-сервер может содержать достаточно много Web-узлов, каждому из которых, как правило, отводится отдельный каталог на жестком диске сервера.

Web-каналы. Обычный Web-узел выдает информацию (запрошенный документ) только в ответ на обращение клиента. Чтобы следить за обновлением опубликованных материалов, пользователь вынужден регулярно обращаться к данному узлу. Современная модель Web-узла позволяет автоматически в заданное время передать обновленную информацию на компьютер зарегистрированного клиента. Такие Web-узлы, способные самостоятельно инициировать поставку информации, называют *каналами*.

Гиперссылки. Отличительной особенностью среды WWW является наличие средств перехода от одного документа к другому, тематически с ним связанному, без явного указания адреса. Связь между документами осуществляется при помощи *гиперссылок*. *Гиперссылка* – это выделенный фрагмент документа (текст или иллюстрация), с которым ассоциирован адрес другого Web-документа. При использовании гиперссылки (обычно для этого требуется навести на нее указатель мыши и один раз щелкнуть) происходит переход по гиперссылке – открытие Web-страницы, на которую указывает ссылка. Перемещение между Web-документами называют *Web-навигацией*. Механизм гиперссылок позволяет организовать тематическое путешествие по WWW без использования адресов конкретных страниц.

Средства просмотра Web. Программы для просмотра Web-страниц называют *браузерами*. Браузер выполняет отображение документа на экране, руководствуясь тегами, которые автор документа внедрил в его текст. Основные функции браузеров следующие:

- установление связи с Web-сервером, на котором хранится документ, и загрузка всех компонентов комбинированного документа;
- интерпретация тегов языка *HTML*, форматирование и отображение Web-страницы в соответствии с возможностями компьютера, на котором браузер работает;
- предоставление средств для отображения мультимедийных и других объектов, входящих в состав Web-страниц, а также механизма расширения, позволяющего настраивать программу на работу с новыми типами объектов;

– обеспечение автоматизации поиска Web-страниц и упрощение доступа к Web-страницам, посещавшимся ранее;

– предоставление доступа к встроенным или автономным средствам для работы с другими службами Internet.

Адресация документов. Гипертекстовая связь между миллиардами документов, хранящихся на серверах Internet, является основой существования пространства WWW. Однако такая связь не могла бы существовать, если бы каждый документ в этом пространстве не обладал своим уникальным адресом. Адрес любого файла в Глобальной сети Internet определяется *унифицированным указателем ресурса – URL*.

Адрес *URL* состоит из трех частей, разделенных точками:

1) указание службы, которая осуществляет доступ к данному ресурсу (обычно обозначается именем протокола, соответствующего данной службе). Так, например, для службы WWW таковым является протокол *HTTP* (*HyperText Transfer Protocol – протокол передачи гипертекста*). После имени протокола ставится двоеточие (:) и два знака «/» (слеш):

http://...

2) указание *доменного имени* компьютера (сервера), на котором хранится данный ресурс:

http://www.abcde.com...

3) указание полного пути доступа к файлу на данном компьютере. В качестве разделителя используется символ «/»:

http://www.abcde.com/Files/New/abcdefg.zip

При щелчке на гиперссылке, содержащей *URL*, браузер посылает запрос для поиска и доставки ресурса, указанного в ссылке. Если по каким-то причинам он не найден, выдается сообщение о том, что ресурс недоступен (возможно, что сервер временно отключен или изменился адрес ресурса).

Преобразование адреса *URL* в цифровую форму *IP*-адреса производит служба имен доменов (*DNS*).

3.4.5. Служба имен доменов (DNS)

Человеку неудобно работать с числовым представлением *IP*-адреса, зато доменное имя запоминается легко, потому что, как правило, это имя содержательное.

Автоматическая работа серверов Сети организована с использованием числового адреса. Благодаря ему промежуточные серверы могут осуществлять передачу запросов и ответов в нужном направлении, не зная, где

именно находятся отправитель и получатель. Поэтому необходим перевод доменных имен в связанные с ними *IP*-адреса. Этим и занимаются серверы службы имен доменов *DNS (Domain Name System)*. Запрос на получение одной из страниц сервера *www.abcde.com* (условный адрес) сначала обрабатывается сервером *DNS*, и далее он направляется по *IP*-адресу, а не по доменному имени.

3.4.6. Служба передачи файлов (FTP)

Прием и передача файлов составляют значительный процент от прочих интернет-услуг. Необходимость в передаче файлов возникает, например, при приеме файлов программ, при пересылке крупных документов, а также при передаче архивных файлов, в которых запакованы большие объемы информации.

Служба *FTP* имеет свои серверы в Мировой сети, на которых хранятся архивы данных. Со стороны клиента для работы с серверами *FTP* может быть установлено специальное ПО, хотя в большинстве случаев браузеры *WWW* обладают встроенными возможностями, реализующими простейшие операции протокола *FTP*, например загрузку файлов с сервера.

Протокол *FTP* работает одновременно с двумя *TCP*-соединениями между сервером и клиентом. По одному соединению идет передача данных, а второе соединение используется как управляющее. Протокол *FTP* также предоставляет серверу средства для идентификации обратившегося клиента. Этим часто пользуются коммерческие серверы и серверы ограниченного доступа, поставляющие информацию только зарегистрированным клиентам, – они выдают запрос на ввод имени пользователя и связанного с ним пароля. Однако существуют и десятки тысяч *FTP*-серверов с анонимным доступом для всех желающих. В этом случае в качестве имени пользователя надо ввести слово *anonymous*, а в качестве пароля задать адрес электронной почты. В большинстве случаев программы-клиенты *FTP* делают это автоматически.

3.4.7. ICQ

Эта служба предназначена для поиска сетевого *IP*-адреса человека, подключенного в данный момент к Internet. Необходимость в подобной услуге связана с тем, что большинство пользователей не имеют постоянного *IP*-адреса. Название службы является акронимом выражения *I seek you – я тебя ищу*. Для пользования этой службой надо зарегистрироваться на ее

центральном сервере (<http://www.icq.com>) и получить персональный идентификационный номер *UIN* (*Universal Internet Number*). При каждом подключении к Internet программа *ICQ*, установленная на компьютере, определяет текущий *IP*-адрес и сообщает его центральной службе, которая, в свою очередь, оповещает партнеров по контактам. Далее партнеры (если они тоже являются клиентами данной службы) могут установить с нужным пользователем прямую связь. Программа предоставляет возможность выбора режима связи («готов к контакту»; «прошу не беспокоить, но готов принять срочное сообщение»; «закрыт для контакта» и т.п.). После установления контакта происходит прямое общение в режиме реального времени.

Кроме того, номер *UIN* можно сообщить партнерам по контактам, и тогда служба *ICQ* приобретает характер Internet-пейджера. Зная номер *UIN* партнера, но не зная его текущий *IP*-адрес, можно через центральный сервер службы отправить ему сообщение с предложением установить соединение.

3.5. Поиск информации в Internet

3.5.1. Браузер Microsoft Internet Explorer

Как уже отмечалось выше, приложение, посредством которого выполняется просмотр *WWW*, называется *Web-браузером*. Наиболее известные браузеры – Internet Explorer (компания Microsoft), Netscape Navigator (компания Netscape Communications) и Opera (компания Opera Software).

Помимо *WWW*, браузеры допускают обращение к другим службам Internet (телеконференции Usenet, файловым архивам *FTP* и др.)

Для изучения выберем браузер Internet Explorer из соображений его особой распространенности.

Для запуска браузера Internet Explorer можно использовать ярлык Internet Explorer на Рабочем столе или Главное меню (*Пуск*→*Программы*→*Internet Explorer*). Кроме того, программа запускается автоматически при попытке открыть документ Internet или локальный документ в формате *HTML*.

Если соединение с Internet отсутствует, то после запуска программы на экране появится диалоговое окно для управления установкой соединения. При невозможности установить соединение сохраняется возможность просмотра в *автономном режиме* ранее загруженных Web-документов.

При наличии соединения, после запуска программы, на экране появится так называемая *начальная* страница, выбранная при настройке программы.

Открытие и просмотр Web-страниц. Просматриваемая Web-страница отображается в рабочей области окна Internet Explorer. По умолчанию воспроизводится все ее содержимое, включая графические иллюстрации и встроенные мультимедийные объекты. Управление просмотром осуществляется при помощи строки меню, панелей инструментов, а также активных элементов, имеющих в открытом документе, например *гиперссылок*.

Если *URL*-адрес Web-страницы известен, его можно ввести в строке *Адрес* и нажать клавишу Enter. Страница с указанным адресом открывается вместо текущей. Наличие средства автозаполнения адресной строки упрощает повторный ввод адресов. Вводимый адрес автоматически сравнивается с адресами ранее просматривавшихся Web-страниц. Все подходящие адреса отображаются в раскрывающемся списке строки *Адрес*. Если нужный адрес есть в списке, его можно выбрать клавишами  и , после чего нажать клавишу Enter. При отсутствии нужного адреса ввод продолжают как обычно.

Навигация в Internet чаще выполняется не путем ввода адреса *URL*, а посредством использования *гиперссылок*. При отображении Web-страницы на экране, гиперссылки выделяются цветом (обычно синим) и подчеркиванием. При наведении указателя мыши на гиперссылку он принимает форму кисти руки, а сама гиперссылка при соответствующей настройке браузера изменяет цвет. Адрес *URL*, на который указывает ссылка, отображается в строке состояния. При щелчке на гиперссылке соответствующая Web-страница загружается вместо текущей. Если гиперссылка указывает на произвольный файл, его загрузка происходит по протоколу *FTP*.

На Web-страницах могут также встречаться графические ссылки (гиперссылки, представленные рисунком) и изображения-карты, объединяющие несколько ссылок в рамках одного изображения.

Дополнительные возможности использования гиперссылок предоставляют их контекстные меню. Чтобы открыть новую страницу, не закрывая текущей, применяют команду *Открыть в новом окне*. В результате открывается новое окно браузера. Адрес *URL*, заданный ссылкой, можно поместить в буфер обмена при помощи команды *Копировать ярлык*. Его

можно вставить в поле строки *Адрес* или в любой другой документ для последующего использования.

Другие операции, относящиеся к текущей странице и ее элементам, также удобно осуществлять через контекстное меню. Например, рисунок, имеющийся на странице, можно:

- сохранить как файл (*Сохранить рисунок как*);
- использовать как фоновый рисунок (*Сделать рисунком рабочего стола*) или как активный элемент (*Сохранить как элемент рабочего стола*).

Если рисунок выполняет функции графической ссылки, к нему можно применять как команды, относящиеся к изображению, так и команды, относящиеся к ссылке.

Приемы управления браузером. Необходимость определенных действий в ходе просмотра документов WWW часто диктуется самим ходом работы. В таких случаях удобно использовать кнопки панели инструментов. Для того чтобы вернуться к странице, которая просматривалась некоторое время назад, используют кнопку *Назад* . Для того чтобы возвратиться на несколько страниц назад, можно использовать присоединенную к ней кнопку раскрывающегося списка. Отменить действия, выполненные при помощи кнопки *Назад*, позволяет кнопка *Вперед* .

Если процесс загрузки страницы затянулся или необходимость в ней отпала, используют кнопку *Остановить* . Заново загрузить Web-страницу, если ее загрузка была прервана или содержание документа изменилось, позволяет кнопка *Обновить* . Чтобы немедленно загрузить начальную страницу, с которой браузер обычно начинает работу, пользуются кнопкой *Домой* .

Создать новое окно, сохранить открытый документ на своем компьютере, распечатать его, включить или выключить режим автономной работы, а также завершить работу с программой позволяют команды меню *Файл*.

Копирование фрагментов документа в буфер обмена и поиск текста на Web-странице осуществляются при помощи команд меню *Правка*.

Включение и выключение отображения служебных элементов окна (панелей инструментов, дополнительных панелей, строки состояния), выбор шрифта и кодировки символов осуществляются через меню *Вид*.

Ведение списка регулярно посещаемых страниц и быстрый доступ к ним осуществляются через меню *Избранное*. Переход к использованию

программ для работы с другими службами Internet, а также настройка браузера осуществляются через меню *Сервис*.

Работа с несколькими окнами. Нередко возникает необходимость открыть новый Web-документ, не закрывая текущий, например, в тех случаях, когда текущий документ содержит список интересных ссылок. Чтобы открыть новое окно программы Internet Explorer, используют команду меню *Файл→Создать→Окно*. Каждое окно отображает свой Web-документ и может использоваться самостоятельно. В частности, списки кнопок *Назад* и *Вперед* обновляются в каждом окне индивидуально. Закрывать окна программы Internet Explorer можно в любом порядке.

Прием файлов из Internet. Гиперссылки, имеющиеся на Web-страницах, могут указывать на документы разных типов. Если браузер не способен отображать файлы определенного типа (например, исполняемые файлы с расширением *.exe*, архивы *.zip* и пр.), инициируется процесс загрузки данного файла на компьютер.

Программа Internet Explorer запускает мастер загрузки файла, на первом этапе работы которого требуется указать, следует ли открыть файл или сохранить его на диске. Открытие файла подразумевает загрузку его в каталог временных файлов и немедленный запуск (если это исполняемый файл) или открытие с помощью программы, которая предназначена для работы с файлами этого типа. Такой подход открывает путь на компьютер для небезопасной информации. Надежнее выбрать сохранение файла на диске. В этом случае требуется выбрать папку, в которой следует сохранить файл.

Ход загрузки файла отображается в специальном окне загрузки. Процесс загрузки файла не препятствует параллельному просмотру Web-страниц или другим операциям в Internet.

Загрузку файла можно прервать в любой момент при помощи кнопки *Отмена*.

3.5.2. Рекомендации по поиску информации

Web-пространство отличается гигантскими размерами. Найти в этом пространстве именно то, что нужно, представляется весьма сложной задачей. Помощь в поиске нужной информации оказывают разнообразные поисковые системы.

Поисковая система представляет собой специализированный Web-узел. Пользователь сообщает поисковой системе данные о содержании ис-

комой Web-страницы, а система выдает ему список гиперссылок на страницы, соответствующие запросу. Наибольшую популярность приобрели эти поисковые каталоги и поисковые указатели.

Поисковые каталоги. Поисковые каталоги устроены по тому же принципу, что и тематические каталоги крупных библиотек. На основной странице поискового каталога размещен сокращенный список крупных тематических категорий. Каждая запись в списке категорий – это гиперссылка. Щелчок на ней открывает следующую страницу поискового каталога, на котором данная тема представлена подробнее. Щелчок на названии темы открывает страницу со списком разделов. Продолжая погружение в тему, можно дойти до списка конкретных Web-страниц и выбрать себе тот ресурс, который лучше подходит для решения задачи.

Поисковые машины. В поисковых машинах происходит автоматический просмотр, отбор и сортировка новых Web-страниц без участия человека. Основной принцип работы поисковой машины заключается в поиске Web-ресурсов по *ключевым словам*. Пользователь описывает искомый ресурс с помощью ключевых слов, после чего дает задание на поиск. Поисковая система анализирует данные, хранящиеся в своей базе, и выдает список Web-страниц, соответствующих запросу. Вместе с гиперссылками выдаются краткие сведения о найденных ресурсах, на основании которых пользователь может выбрать нужные ему ресурсы.

Сегодня в мире существует огромное количество поисковых машин. Вершину списка занимают около двух десятков зарубежных систем: Alta Vista (www.atavista.com), Excite (www.excite.com), Fast Search (www.alltheweb.com), Go/Infoseek (www.go.com), Google (www.google.com), HotBot (hotbot.lycos.com), Lycos (www.lycos.com), Netscape Search (search.netscape.com), WebCrawler (www.webcrawler.com) и др. В России также имеется несколько поисковых указателей, из которых наиболее крупными и популярными являются следующие: Aport (www.afort.ru), Яндекс (www.yandex.ru) и Rambler (www.rambler.ru).

Рекомендации по приемам эффективного поиска. Для проведения реферативного поиска, рекомендуется пользоваться поисковыми каталогами, такими, как Yahoo! (www.yahoo.com) или «Атрус» (atrus.afort.ru). Это позволит быстро установить местоположение основных первоисточников. При ознакомлении с первоисточниками следует, прежде всего, уделять внимание понятийной базе. Знание основных понятий и терминов позво-

лит перейти к углубленному поиску в поисковых указателях с использованием ключевых слов, наиболее точно характеризующих тему.

При наличии первичных сведений по теме поиска, документы можно разыскивать с помощью поисковых машин. При этом следует различать приемы *простого, расширенного, контекстного и специального поиска*.

Под *простым поиском* понимается поиск Web-ресурсов по одному или нескольким ключевым словам. Недостаток простого поиска заключается в том, что обычно он выдает слишком много документов, среди которых трудно выбрать наиболее подходящие.

При использовании *расширенного поиска* ключевые слова связывают между собой операторами логических отношений. Расширенный поиск применяют в тех случаях, когда приемы простого поиска дают слишком много результатов. С помощью логических отношений поисковое задание формируют так, чтобы более точно детализировать задание и ограничить область отбора, например по дате публикации или по типу данных.

Контекстный поиск – это поиск по точной фразе. Он удобен для реферативного поиска информации, но доступен далеко не во всех поисковых системах.

Специальный поиск применяют при розыске Web-страниц, содержащих ссылки на заданные адреса *URL*, а также содержащих заданные данные в служебных полях, например в поле заголовка.

Рекомендации по использованию поисковых систем. Для проведения научных поисков рекомендуется пользоваться поисковой системой Northern Light (www.northernlight.com). Эта система имеет один из лучших коэффициентов охвата Web-пространства, и ее администрация прилагает специальные усилия для поддержания актуальности своих указателей. Кроме того, система удачно сочетает свойства поискового каталога и поисковой машины. По наиболее популярным темам в ней можно найти специальные разделы каталожного типа; они называются *Special Editions* и подготавливаются вручную. Дополнительно система предоставляет платные услуги по поставке актуальных научных документов. Они находятся в разделе *Special Collection*.

Самым большим поисковым указателем обладает поисковая система Fast Search (www.alltheweb.com). Всего за один год после запуска эта поисковая система вышла на первое место в мире по объему проиндексированного пространства.

Исторически одной из наиболее популярных считается поисковая система Alta Vista (www.altavista.com).

В России в настоящее время действуют три примерно одинаковых по мощности поисковых указателя: Aport (www.afort.ru), Rambler (www.rambler.ru), Яндекс (www.yandex.ru). Все они обладают примерно одинаковым «знанием» о ресурсах российского сектора WWW и работают достаточно быстро. Систему Апорт удобно использовать в операциях простого поиска. В этой системе приняты специальные меры по устранению дубликатов, удалению неактуальных ссылок, наглядному представлению результатов поиска. Система Rambler по своей сути является не только поисковой, но и выполняет функции удобного Web-портала. Систему Яндекс удобно использовать при формировании сложных поисковых заданий, поскольку она обладает наиболее гибким языком для расширенного поиска.

3.6. Компьютерные вирусы и антивирусные программы

Компьютерный вирус – специальная программа, способная самопроизвольно присоединяться к другим программам («заражать» их) и при запуске последних выполнять различные нежелательные действия: порчу файлов и папок, искажение результатов вычислений, засорение или стирание памяти, создание помех в работе компьютера. Наличие вирусов проявляется в следующих ситуациях:

- некоторые программы перестают работать или начинают работать некорректно;
- на экран выводятся посторонние сообщения, сигналы и другие эффекты;
- работа компьютера существенно замедляется;
- структура некоторых файлов оказывается испорченной и т.д.

3.6.1. Классификация компьютерных вирусов

Имеется несколько признаков классификации существующих компьютерных вирусов:

- по среде обитания;
- по области поражения;
- по особенностям алгоритма;
- по способу заражения;
- по деструктивным возможностям.

Рассмотрим приведенную классификацию более детально.

Классификация вирусов по среде обитания. Различают файловые, загрузочные, макро- и сетевые вирусы.

Файловые вирусы – наиболее распространенный тип вирусов. Эти вирусы внедряются в выполняемые файлы, создают файлы-спутники (companion-вирусы) или используют особенности организации файловой системы (link-вирусы).

Загрузочные вирусы записывают себя в загрузочный сектор диска (boot-сектор) или в сектор системного загрузчика жесткого диска (Master Boot Record). Начинают работу при загрузке компьютера и обычно становятся резидентными (постоянно хранящимися во время работы в оперативной памяти). Как правило, эти вирусы состоят из двух частей, поскольку загрузочная запись имеет небольшой размер и в ней трудно разместить целиком программу вируса.

Макровирусы заражают файлы широко используемых пакетов обработки данных. Эти вирусы представляют собой программы, написанные на встроенных в эти пакеты языках программирования. Наибольшее распространение получили макровирусы для приложений Microsoft Office. Для своего размножения такие вирусы используют возможности встроенного языка Visual Basic for Applications (VBA).

Сетевые вирусы используют для своего распространения протоколы или команды компьютерных сетей и электронной почты. Основным принципом работы сетевого вируса является возможность самостоятельно передать свой код на удаленный сервер или рабочую станцию. Полноценные сетевые вирусы при этом должны обладать возможностью запустить на удаленном компьютере свой код на выполнение.

На практике существуют разнообразные сочетания вирусов, например файлово-загрузочные вирусы, заражающие как файлы, так и загрузочные сектора дисков, или сетевые макровирусы, которые заражают редактируемые документы и рассылают свои копии по электронной почте.

Классификация вирусов по области поражения. Как правило, каждый вирус заражает файлы одной или нескольких ОС: MS-DOS, Windows, Win95/NT, OS/2, Unix. Макровирусы заражают файлы форматов MS Word, MS Excel и других приложений MS Office. Многие загрузочные вирусы также ориентированы на конкретные форматы расположения системных данных в загрузочных секторах дисков.

Классификация вирусов по особенностям алгоритма. Выделяют резидентные вирусы, стелс-вирусы (*stealth* – *англ.* невидимка), полиморфные и др.

Резидентные вирусы способны оставлять свои копии (или части) в оперативной памяти, перехватывать обработку событий (например, обращения к файлам или дискам) и вызывать при этом процедуры заражения объектов (файлов и секторов). Эти вирусы активны в памяти не только в момент работы зараженной программы, но и после. Резидентные копии таких вирусов жизнеспособны до перезагрузки ОС, даже если на диске уничтожены все зараженные файлы. От таких вирусов сложно избавиться простым восстановлением копий файлов с дистрибутивных или резервных дисков. Это объясняется тем, что резидентная копия вируса остается активной в оперативной памяти и заражает вновь создаваемые файлы. Если резидентный вирус является также загрузочным и активизируется при загрузке ОС, то даже форматирование диска при наличии в памяти этого вируса его не удаляет. Это объясняет то, что многие резидентные вирусы заражают диск повторно, после того как он отформатирован.

Нерезидентные вирусы, напротив, активны довольно непродолжительное время – только в момент запуска зараженной программы. Для своего распространения они выбирают на диске незараженные файлы и записываются в них. После окончания работы зараженной программы вирус становится неактивным вплоть до очередного запуска какой-либо зараженной программы. Зараженные нерезидентными вирусами файлы восстанавливаются значительно проще.

Стелс-алгоритмы позволяют вирусам полностью или частично скрыть свое присутствие. Наиболее распространенным *стелс-алгоритмом* является перехват запросов ОС на чтение/запись зараженных объектов. Стелс-вирусы при этом либо временно «лечат» эти объекты, либо подставляют вместо себя незараженные участки информации.

Полиморфность (самошифрование) используется для усложнения процедуры обнаружения вируса. *Полиморфные вирусы* – это трудно выявляемые вирусы, не имеющие постоянного участка кода. В общем случае два образца одного и того же вируса не имеют совпадений. Это достигается шифрованием основного тела вируса и модификациями программы-расшифровщика. Так, например, некоторые макровирусы при создании своих новых копий случайным образом меняют имена своих переменных, вставляют пустые строки или модифицируют свой код иным способом.

Классификация вирусов по способу заражения. Различают так называемые троянские программы, утилиты скрытого администрирования, Intended-вирусы и пр.

Троянские программы получили свое название по аналогии с троянским конем. Назначение этих программ – имитация каких-либо полезных программ, новых версий популярных утилит или дополнений к ним. Очень часто они рассылаются через электронные конференции. При их записи пользователем на свой компьютер троянские программы активизируются и выполняют нежелательные действия.

Разновидностью троянских программ являются *утилиты скрытого администрирования* (backdoor). По своей функциональности и интерфейсу они во многом напоминают системы администрирования компьютеров в Сети, разрабатываемые и распространяемые различными фирмами-производителями программных продуктов. При инсталляции эти утилиты самостоятельно устанавливают на компьютере систему скрытого удаленного управления. В результате возникает возможность скрытого управления этим компьютером. Реализуя заложенные алгоритмы, утилиты без ведома пользователя принимают, запускают или отсылают файлы, уничтожают информацию, перезагружают компьютер и пр. Возможно использование этих утилит для обнаружения и передачи паролей и иной конфиденциальной информации, запуска вирусов, уничтожения данных.

К *Intended-вирусам* относятся программы, которые не способны размножаться из-за существующих в них ошибок. Например, вирусы при заражении не помещают в начало файла команду передачи управления на код вируса или записывают в нее неверный адрес своего кода. К этому классу также можно отнести вирусы, которые размножаются только один раз. Заразив какой-либо файл, они теряют способность к дальнейшему размножению через него.

Классификация вирусов по деструктивным возможностям. Вирусы разделяют на:

- *неопасные*, влияние которых ограничивается уменьшением свободной памяти на диске, замедлением работы компьютера, графическими и звуковыми эффектами;
- *опасные*, которые потенциально могут привести к нарушениям в структуре файлов и сбоям в работе компьютера;
- *очень опасные*, в алгоритм работы которых специально заложены процедуры уничтожения данных и, согласно одной из неподтвержденных

гипотез, возможность обеспечивать быстрый износ движущихся частей механизмов путем ввода в резонанс и разрушения головок чтения/записи некоторых накопителей на жестких дисках.

3.6.2. Конструкторы вирусов

Конструктор вирусов – это утилита, предназначенная для изготовления новых компьютерных вирусов. Известны конструкторы вирусов для MS-DOS, Windows и макровирусов. Они позволяют генерировать исходные тексты вирусов, объектные модули и/или непосредственно зараженные файлы. Некоторые конструкторы снабжены стандартным оконным интерфейсом, где при помощи системного меню можно выбрать тип вируса, поражаемые объекты, наличие или отсутствие самошифровки, противодействие отладчику, внутренние текстовые строки, сопровождающие работу вируса, эффекты и др.

3.6.3. Методы борьбы с компьютерными вирусами

Для борьбы с вирусами существуют программы, которые можно классифицировать по основным группам: мониторы, детекторы, доктора, ревизоры и вакцины.

Программы-мониторы. Программы-мониторы (иначе называемые программы-фильтры) располагаются резидентно в оперативной памяти компьютера, перехватывают и сообщают пользователю об обращениях ОС, которые используются вирусами для размножения и нанесения ущерба. Пользователь имеет возможность разрешить или запретить выполнение этих обращений. К преимуществу таких программ относят возможность обнаружения неизвестных вирусов. Это актуально при наличии самомодифицирующихся вирусов. Использование программ-фильтров позволяет обнаруживать вирусы на ранней стадии заражения компьютера.

Недостатками программ являются невозможность отслеживания вирусов, обращающихся непосредственно к BIOS, а также загрузочных вирусов, активизирующихся до запуска антивируса при загрузке MS-DOS; частая выдача запросов на выполнение операции.

Программы-детекторы. Программы-детекторы проверяют, имеется ли в файлах и на дисках специфическая для данного вируса комбинация байтов. При ее обнаружении выводится соответствующее сообщение. Однако если программа не опознается детекторами как зараженная, то возможно в ней находится новый вирус или модифицированная версия старого, неизвестная программе-детектору.

Программы-доктора. Программы-доктора восстанавливают зараженные программы путем удаления из них тела вируса. Обычно эти программы рассчитаны на конкретные типы вирусов и основаны на сравнении последовательности кодов, содержащихся в теле вируса, с кодами проверяемых программ. Программы-доктора необходимо периодически обновлять с целью получения новых версий, обнаруживающих новые виды вирусов.

Программы-ревизоры. Программы-ревизоры анализируют изменения состояния файлов и системных областей диска. Проверяют состояния загрузочного сектора и таблицы FAT; длину, атрибуты и время создания файлов; контрольную сумму кодов. Пользователю сообщается о выявлении несоответствий.

Программы-вакцины. Программы-вакцины модифицируют программы и диски так, что это не отражается на работе программ, но вирус, от которого производится вакцинация, считает программы или диски уже зараженными.

Существующие антивирусные программы в основном относятся к классу гибридных программ (детекторы-доктора, доктора-ревизоры и др.).

При заражении или при подозрении на заражение компьютера вирусом необходимо:

- оценить ситуацию и не предпринимать действий, приводящих к потере информации. Если вы не обладаете достаточными знаниями и опытом, лучше обратиться к специалистам;

- перезагрузить ОС компьютера. При этом использовать специальную, заранее созданную и защищенную от записи системную дискету. В результате будет предотвращена активизация загрузочных и резидентных вирусов с жесткого диска компьютера;

- запустить имеющиеся антивирусные программы, пока не будут обнаружены и удалены все вирусы. В случае невозможности удалить вирус и при наличии в файле ценной информации произвести архивирование файла и подождать выхода новой версии антивируса. После окончания перезагрузить компьютер.

К антивирусным программам, получившим распространение в России, странах СНГ и за рубежом, относят программы фирм Symantec (Norton Antivirus), Network Associates (Doctor Solomon) и российских фирм – Лаборатория Касперского (AntiViral Toolkit Pro) и ДиалогНаука (ADinf, Dr.Web).

4. ТЕКСТОВЫЙ ПРОЦЕССОР MICROSOFT WORD

Текстовые процессоры (редакторы) – наиболее широко используемый вид офисных программ. В настоящее время наибольшее распространение у нас получил текстовый процессор *Microsoft Word*.

Microsoft Word – мощный текстовый процессор, предназначенный для выполнения всех процессов обработки текста: от набора и верстки, до проверки орфографии, вставки в текст графики, распечатки текста. Этот процессор поддерживает принцип соответствия экранного изображения печатному (принцип *WYSIWYG*, т.е. «*What You See Is What You Get*»). Это означает примерно следующее: «Что Вы видите, то и получите». Иными словами, на экране показывается вид реального документа, который будет выглядеть так же после распечатывания на бумаге.

Microsoft Word позволяет использовать различные шрифты, их начертания, автоматическую коррекцию текста по границам, автоматический перенос слов и проверку правописания, набирать текст из нескольких колонок, создавать списки различных типов, размещать в тексте таблицы, диаграммы, графические объекты, создавать оглавления и предметные указатели. Наличие шаблонов позволяет быстро создать деловое письмо, факс, автобиографию, расписание, календарь и др. Наличие закладки в тексте позволяет быстро перейти к отмеченному месту в тексте.

Microsoft Word обеспечивает сохранение текста через определенный устанавливаемый промежуток времени, поиск заданного слова или фрагмента текста, замену его на указанный фрагмент. Можно так же автоматически включать в текст дату, время создания, обратный адрес и имя написавшего текст. Для ограничения доступа к документу можно установить пароль, который Word будет запрашивать при загрузке файла для выполнения с ним каких-либо действий. Word позволяет открывать множество окон для параллельной работы с несколькими документами, а также разделять по горизонтали одно активное окно на две части для одновременного просмотра двух частей документа.

4.1. Технология подготовки текстового документа

4.1.1. Режимы просмотра документов

1) *Обычный режим* – режим, в котором отображается форматированный текст и упрощенная разметка страницы (т.е. не отображаются ко-

лонтитулы, сноски, границы страниц, фон, графические объекты и рисунки). Обычный режим предназначен для ввода, редактирования и форматирования текста.

2) *Web-документ* – удобен для создания web-страниц или документов, предназначенных для просмотра на экране. В этом режиме отображается фон, перенос текста выполняется по размерам окна, а рисунки занимают те же позиции, что и в окне web-обозревателя.

3) *Режим разметки* – экранное представление документа полностью соответствует печатному. Этот режим используется для редактирования колонтитулов, настройки полей страницы, а также для работы с колонками и графическими объектами.

4) *Режим структуры* – режим, в котором заголовки документа смещены вправо в зависимости от соответствующего им уровня в структуре документа. В режиме структуры можно свернуть документ, оставив основные заголовки, или развернуть его, отобразив все заголовки и основной текст, а также перемещать, копировать и реорганизовывать текст посредством перетаскивания заголовков.

Выбор одного из четырех вышеуказанных режимов просмотра документа выполняют с помощью команд меню *Вид* или командных кнопок, расположенных в левом нижнем углу окна приложения .

5) *Схема документа* – в этом режиме вдоль левого края окна документа появляется вертикальная область, в которой изображается структура заголовков документа. Схема документа позволяет быстро перемещаться по документу и отслеживать текущее местонахождение в нем. Для перехода в режим *Схема документа* используется команда меню *Вид*→*Схема документа* или кнопка на панели инструментов *Стандартная* .

6) *Предварительный просмотр* – режим отображения документа в том виде, в каком он будет напечатан. Для перехода в режим *Предварительный просмотр* используется команда меню *Файл*→*Предварительный просмотр*.

Масштабирование изображения документа позволяет изменить (увеличить или уменьшить) размер изображения текста на экране. Масштабирование осуществляется с помощью поля *Масштаб* на панели инструментов *Стандартная* или командой меню *Вид*→*Масштаб*.

4.1.2. Установка параметров страницы

Поля страницы. Поля страницы представляют собой пустое пространство возле краев страниц. Как правило, текст и графические элементы вставляются в область печати, ограниченную полями страницы. Однако некоторые элементы, например верхние и нижние колонтитулы и номера страниц, можно разместить на полях. Для изменения полей страницы можно использовать команду меню *Файл*→*Параметры страницы*, после чего откроется окно диалога *Параметры страницы* (рис. 4.1).

В этом окне на вкладке *Поля* задаются размеры верхнего, нижнего, правого и левого полей страницы. Кроме того, для задания размеров полей можно использовать горизонтальную и вертикальную линейки.

Ориентация страницы. Чтобы выбрать ориентацию страницы, следует выполнить команду меню *Файл*→*Параметры страницы* – откроется окно диалога *Параметры страницы* (см. рис. 4.1). В этом окне на вкладке *Поля* в группе *Ориентация* выбрать вариант *книжная* или *альбомная*.

Размер бумаги. Microsoft Word позволяет создавать документы на листах формата A5 (размер почтовой карточки), формата A4 (обычный печатный лист размером 210×297 мм), формата A3 (удвоенный A4) или установить собственный размер листа. Выбор размера страницы выполняется на вкладке *Размер бумаги* путем выбора нужного значения из списка *Размер бумаги* или с помощью счетчиков *Ширина* и *Высота* (рис. 4.2).

4.1.3. Ввод и редактирование текста

Позиция ввода или редактирования текста указывается на экране текстовым курсором. Необходимо отличать текстовый курсор, изображаемый в тексте мигающей вертикальной чертой «|», от указателя мыши, ко-

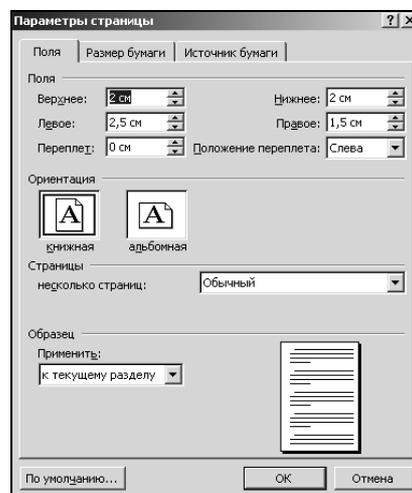


Рис. 4.1. Окно диалога *Параметры страницы* (вкладка *Поля*)

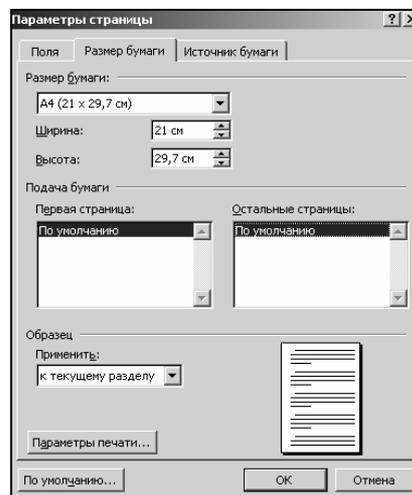


Рис. 4.2. Окно диалога *Параметры страницы* (вкладка *Размер бумаги*)

торый может менять свой вид в зависимости от того, в какой области окна он находится.

Ввод текста. При вводе текста необходимо соблюдать следующие правила:

– в пределах одного абзаца переход к новой строке осуществляется автоматически при достижении границы абзаца, поэтому Enter в конце каждой строки нажимать не надо;

– для принудительного перехода к началу новой строки в пределах одного абзаца нужно нажать Shift+Enter. Продолжение текста будет с новой строки, однако цельность абзаца при этом не будет нарушена;

– переход к вводу нового абзаца выполняется нажатием клавиши Enter;

– перед знаками . , ; : ? ! пробел не делается, а после них – обязательно. Тире с обеих сторон отделяется пробелами от остального текста.

Ввод специальных и произвольных символов. При вводе текста часто существует необходимость ввода специальных и произвольных символов, не имеющих соответствующей клавиши в раскладке клавиатуры. Основным средством для ввода специальных и произвольных символов, а также для закрепления их за избранными клавишами является диалоговое окно *Символ* (меню *Вставка*→*Символ*) (рис. 4.3).

В этом окне имеется две вкладки: *Символы* и *Специальные знаки*. На вкладке *Специальные знаки* присутствует список таких символов как «длинное» («полиграфическое») тире, «авторское право» ©, «торговая марка» ™, «параграф» § и др. Для вставки такого символа нужно выделить его в списке и щелкнуть на кнопке *Вставить*.

На вкладке *Символы* представлены элементы управления для ввода

произвольных символов из любых символьных наборов. Центральное положение в окне занимает таблица символов текущего набора. Выбор шрифта выполняют в раскрывающемся списке *Шрифт*. Если шрифт относится к категории универсальных шрифтов *UNICODE*, то для него имеется и возможность выбора символьного набора в соответствующем раскрывающемся списке *Набор*.

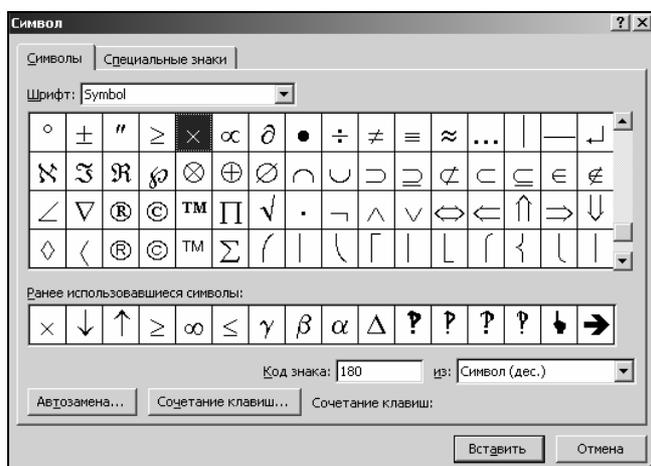


Рис. 4.3. Окно диалога *Символ*

Редактирование текста. Под редактированием текста понимается выполнение следующих операций: *добавление, удаление, перемещение и копирование* фрагмента текста. Перед выполнением операций удаления, перемещения и копирования соответствующий фрагмент текста предварительно должен быть выделен. Минимальный размер фрагмента – один символ, максимальный – весь текст документа.

Выделение фрагментов текста. Способы выделения фрагментов текста представлены в таблицах 4.1 и 4.2.

Таблица 4.1

Выделение фрагментов текста с помощью клавиатуры

Комбинация клавиш	Выделение
Shift + ← (→), Shift + ↑ (↓)	Одного символа, строки
Shift + Ctrl + ← (→), Shift + Ctrl + ↑ (↓)	Одного слова, абзаца
Shift + Home, Shift + End	До начала, конца строки
Shift + Ctrl + Home, Shift + Ctrl + End	До начала, конца документа
PgUp, PgDn	На одну экранную страницу вверх, вниз

Таблица 4.2

Выделение фрагментов текста с помощью мыши

Действие с мышью	Область щелчка	Выделение
Передвигать мышь при нажатой левой кнопке	Начальная точка выделения в тексте	От начальной точки выделения до желаемой посимвольно
Двойной щелчок левой кнопкой мыши	Внутри слова	Слова
Двойной щелчок левой кнопкой мыши	В левом поле	Абзаца
Тройной щелчок левой кнопкой мыши	Внутри абзаца	Абзаца
Тройной щелчок левой кнопкой мыши	В левом поле	Документа
Ctrl + однократный щелчок левой кнопкой мыши	Внутри предложения	Предложения
Alt + передвигать мышь при нажатой левой кнопке	Внутри абзаца	Вертикального блока

4.1.4. Форматирование текста

Различают два типа форматирования:

- символьное (настройка шрифта);
- форматирование абзаца.

Настройка шрифта. Настройку шрифта выполняют в диалоговом окне *Шрифт* (меню *Формат*→*Шрифт*) (рис. 4.4).

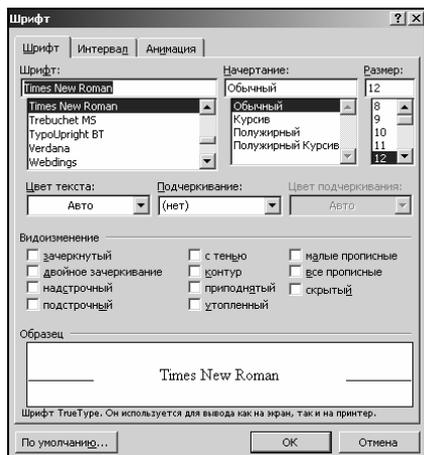


Рис. 4.4. Окно диалога *Шрифт*

Это окно содержит три вкладки: *Шрифт*, *Интервал* и *Анимация*.

На вкладке *Шрифт* выбирают:

- гарнитуру шрифта;
- его размер;
- вариант начертания;
- цвет символов;
- наличие подчеркивания;
- характер видоизменения (эффекты).

На вкладке *Интервал* задается интервал между символами текста путем выбора одного из трех значений (*Обычный*, *Разреженный*, *Уплотненный*) и указания нужной величины в поле *на*.

Вкладка *Анимация* позволяет использовать эффекты анимации при подготовке электронных документов (т.е. документов, распространяемых в формате текстового процессора). В печатных документах эти эффекты не воспроизводимы и, соответственно, их использование не имеет смысла.

Кроме диалогового окна *Шрифт* для настройки шрифта могут использоваться соответствующие кнопки панели инструментов *Форматирование* (рис. 4.5).

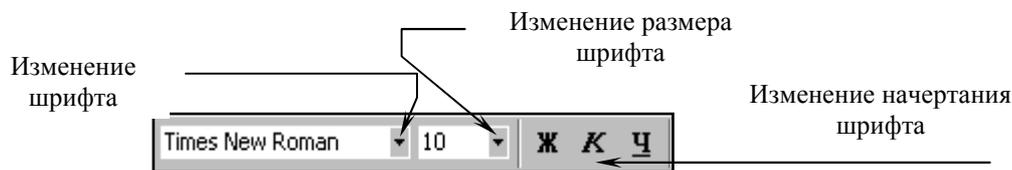


Рис. 4.5. Поля и кнопки для настройки шрифта

Настройка абзаца. Настраиваются следующие параметры абзаца:

- величина отступа слева (от левого поля);
- величина отступа справа (от правого поля);
- величина отступа первой строки абзаца («красная строка»);

- величина интервала (отбивки между абзацами) перед абзацем и после него;
- выравнивание текста в абзаце;
- междустрочный интервал.

Для печатных документов величину отступа основного текста, как правило, не задают (необходимое положение текста определяется шириной полей), но ее задают для дополнительных материалов и заголовков, если они не выравниваются по центру.

Роль отступа первой строки абзаца и интервалов между абзацами, состоит в том, чтобы визуально выделить абзацы. Совместное применение этих средств не рекомендуется.

Для текста в абзаце может использоваться выравнивание одного из следующих типов:

- по левому краю;
- по центру;
- по правому краю;
- по ширине.

Настройку абзаца выполняют либо на вкладке *Отступы и интервалы* диалогового окна *Абзац* (рис. 4.6), которое открывается командой меню *Формат*→*Абзац*, либо с помощью элементов горизонтальной линейки и соответствующих кнопок панели инструментов *Форматирование* (рис. 4.7).

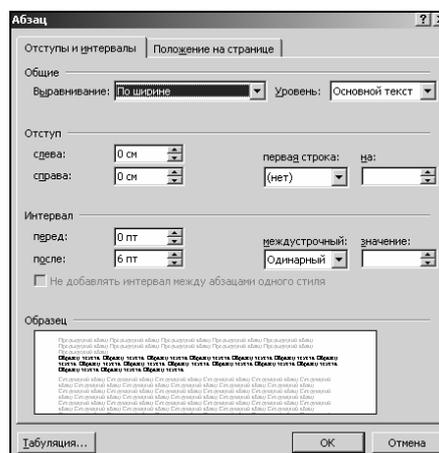


Рис. 4.6. Окно диалогого *Абзац*

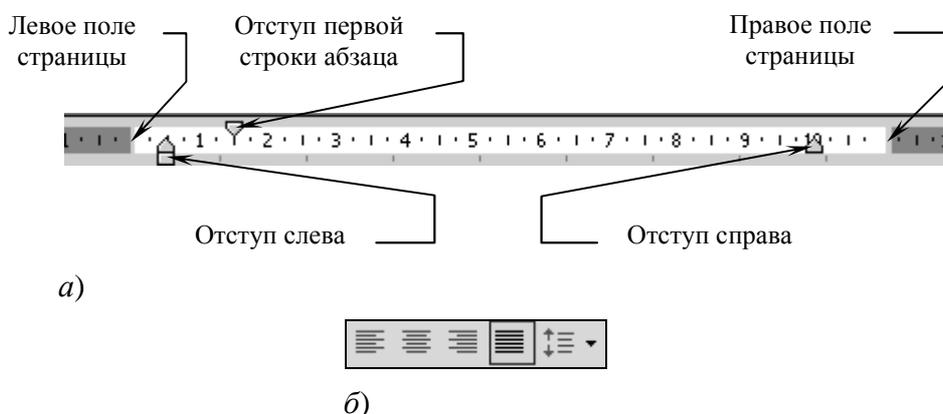


Рис. 4.7. Настройка параметров абзаца: *а* – с помощью линейки (отступы от левого и правого полей, абзацный отступ); *б* – с помощью кнопок панели инструментов «Форматирование» (выравнивание, междустрочный интервал)

Кроме того, абзацы можно снабдить линиями обрамления с любой из сторон и использовать заливку для затенения текста. Для этого необходимо войти в меню *Формат*→*Границы и заливка* – откроется окно диалога *Границы и заливка*→вкладки *Граница* и *Заливка* (рис. 4.8).

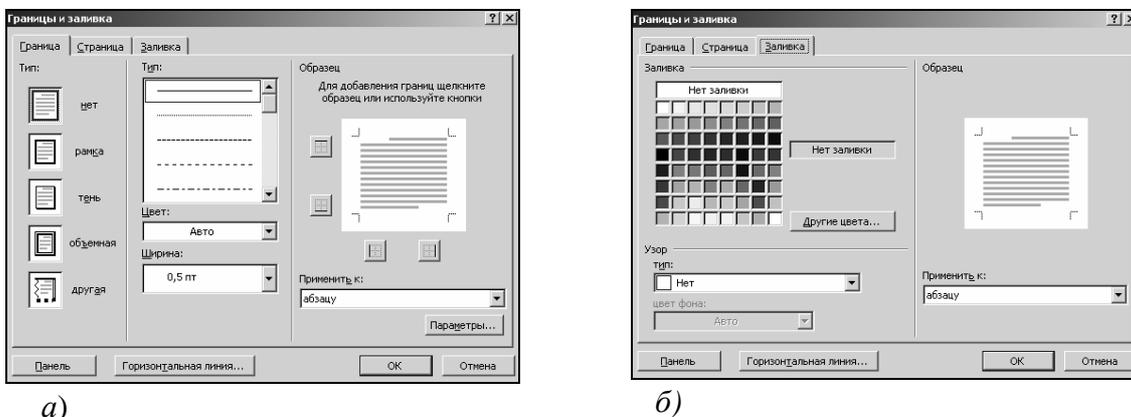


Рис. 4.8. Окно диалога «Границы и заливка»: а – вкладка *Граница*; б – вкладка *Заливка*

4.1.5. Использование стилей при создании документа

Стиль – это именованная совокупность параметров форматирования текста. Стили позволяют одним действием применить сразу всю группу атрибутов форматирования к какому-либо фрагменту текста.

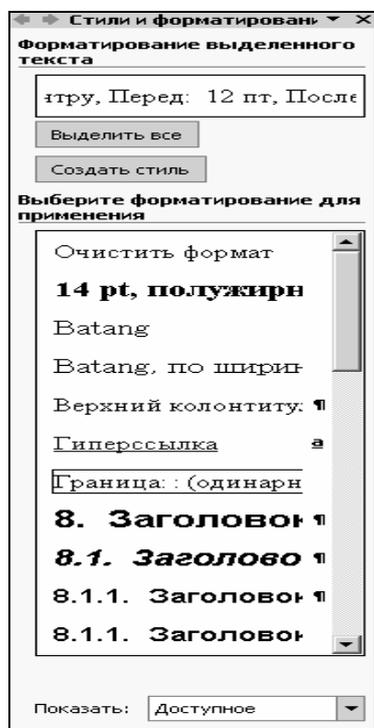


Рис. 4.9. Область задач *Стили и форматирование*

Благодаря этому обеспечивается простота форматирования абзацев и заголовков текста, а также единство их оформления в рамках всего документа.

Работа со стилями состоит в создании, настройке и использовании стилей. Некоторое количество стандартных стилей присутствует в настройке программы по умолчанию. Их используют путем выбора нужного стиля из раскрывающегося списка на панели инструментов *Форматирование*.

Настройка и создание стилей выполняются в области задач *Стили и форматирование* (рис. 4.9), которая открывается командой меню *Формат*→*Стили и форматирование*.

4.1.6. Верстка многостраничного документа

Если в Microsoft Word создается многостраничный документ, то в тексте могут появиться следующие структурные элементы: *закладки, сноски, перекрестные ссылки, колонтитулы*.

Под *закладкой* понимается определенное место в тексте, которому присвоено имя. Закладка позволяет быстро переходить к той части документа, где она расположена. Создание закладки, переход к ней и ее удаление выполняются в окне диалога *Закладка*, которое выводится командой меню *Вставка→Закладка*.

Иногда при создании документа в нем требуется размещать подстрочные примечания к основному тексту. Подстрочные примечания оформляются *сносками*. В состав сноски входят два неразрывно связанных элемента: *знак сноски* и *текст сноски*. Знак сноски (число, знак или сочетание знаков, указывающие на наличие в сноске дополнительных сведений) располагается в основном тексте документа и в начале самого примечания. В Microsoft Word различают *обычные* и *концевые сноски*. Обычные сноски могут располагаться у нижнего поля страницы или сразу под последней строкой текста на странице. Они используются для различных пояснений и комментариев по ходу текста. Концевые сноски располагаются в конце документа. Они могут использоваться для ссылок на литературные источники. Создание сносок выполняется в окне диалога *Сноски*, которое открывается командой меню *Вставка→Ссылка→Сноска*. Для удаления сноски нужно выделить знак сноски в основном тексте, а затем нажать клавишу Delete.

Перекрестная ссылка – это текст, позволяющий пользователю обратиться к элементу, который находится в другой части документа. Можно создать перекрестные ссылки на заголовки, сноски, закладки, названия и нумерованные абзацы. Создание перекрестных ссылок выполняется в окне диалога *Перекрестные ссылки*, которое открывается командой меню *Вставка→Ссылка→Перекрестная ссылка*. Для удаления перекрестной ссылки нужно выделить текст ссылки, а затем нажать клавишу Delete.

Колонтитулами называют области, расположенные в верхнем и/или нижнем поле каждой страницы документа. Соответственно различают *верхний* и *нижний колонтитулы*.

В колонтитулах обычно размещается текст и/или рисунок, который должен быть напечатан внизу или вверху каждой страницы документа (номер страницы (колонцифра), дата печати документа, эмблема организации,

название документа, имя файла, фамилия автора и т.п.). Для создания колонтитулов используется панель инструментов *Колонтитулы*, которая выводится командой меню *Вид*→*Колонтитулы*. Для удаления колонтитула нужно выделить содержимое колонтитула, а затем нажать клавишу Delete.

Стандартными параметрами оформления страниц документа являются также поля страниц, размер и ориентация печатного листа; расположение колонтитулов (меню *Файл*→*Параметры страницы*→вкладка *Источник бумаги*); количество колонок текста (меню *Формат*→*Колонки*).

4.1.7. Буфер обмена Microsoft Office

При создании документа путем использования фрагментов текста, взятых из разных первоисточников, удобно пользоваться *буфером обмена Microsoft Office*. Буфер обмена Microsoft Office позволяет накапливать текстовые и графические элементы из большого числа документов Microsoft Word и/или других приложений, а затем вставлять их в создаваемый документ. Буфер обмена Microsoft Office может вместить до 24 элементов. При копировании 25-го элемента первый элемент в буфере обмена Microsoft Office удаляется. Для работы с буфером обмена Microsoft Office используются стандартные команды *Вырезать*, *Копировать* и *Вставить*. Накопленные элементы остаются в буфере обмена Microsoft Office до закрытия всех приложений Microsoft Office.

Просмотреть содержимое буфера обмена Microsoft Office можно в области задач *Буфер обмена*, которая открывается командой меню *Правка*→*Буфер обмена Office*. Здесь в поле списка отображаются записи о скопированных в буфер элементах. Каждая запись включает в себя значок, указывающий на исходное приложение Microsoft Office, и часть скопированного текста или эскиз скопированного рисунка. Вставка в текущий документ какого-либо элемента из буфера выполняется щелчком мыши по соответствующей записи.

Буфер обмена Microsoft Office связан с системным буфером обмена:

- при копировании нескольких элементов в буфер обмена Microsoft Office последний из них всегда копируется в системный буфер обмена;
- при использовании команды *Вставить*, или кнопки *Вставить*, или сочетания клавиш *Ctrl + V* выполняется вставка содержимого системного буфера обмена, а не буфера обмена Microsoft Office.

4.1.8. Средства отмены и возврата действий

Все операции *ввода, редактирования и форматирования* текста запоминаются текстовым процессором, и поэтому необходимое количество последних действий можно отменить. Для этого используются следующие средства:

- комбинация клавиш `Ctrl + Z`;
- меню *Правка*→*Отменить действие*;
- кнопка *Отменить действие*  на панели инструментов *Стандартная*.

После отмены ряда действий существует возможность вернуться к состоянию, предшествовавшему отмене. Для этого служат:

- комбинация клавиш `Ctrl + Y`;
- меню *Правка*→*Вернуть действие*;
- кнопка *Вернуть действие*  на панели инструментов *Стандартная*.

4.1.9. Средства создания списков

Маркированные и нумерованные списки. Для создания маркированного или нумерованного списка можно воспользоваться одним из двух способов:

- установить параметры списка и набирать элементы списка;
- набрать элементы списка, проследить, чтобы каждый элемент списка находился в отдельном абзаце, выделить нужные абзацы и установить параметры списка.

Параметры списка можно установить в диалоговом окне *Список* (открывается командой меню *Формат*→*Список*) на одной из вкладок – *Маркированный*, *Нумерованный*, *Многоуровневый*. Кроме того, для нумерованных и маркированных списков можно использовать кнопки *Нумерация*  и *Маркеры*  панели инструментов *Форматирование*.

Создание нумерованного или маркированного списка при вводе:

- 1) ввести соответственно **1.** или *****;
- 2) нажать клавишу *Пробел* или `Tab`;
- 3) ввести элемент списка;
- 4) чтобы добавить следующий элемент списка, нажать клавишу `Enter`;

5) чтобы завершить список, дважды нажать клавишу Enter либо, чтобы удалить последний номер или маркер списка, нажать клавишу BackSpace.

Созданный нумерованный список может быть преобразован в маркированный и наоборот. Для этого нужно:

- 1) выделить список, который нужно преобразовать;
- 2) на панели инструментов *Форматирование* нажать кнопку *Маркеры* или *Нумерация*.

Изменение формата маркированного или нумерованного списка:

1) выделить список с номерами или маркерами, формат которых необходимо изменить;

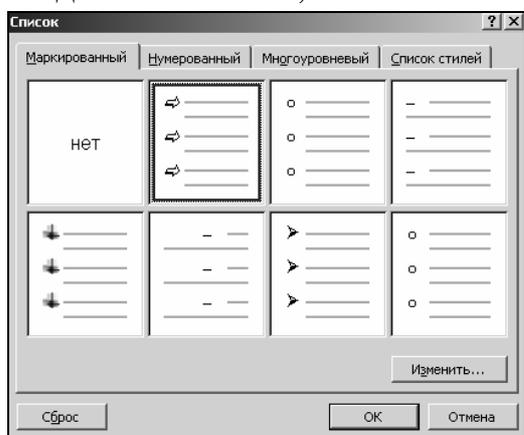


Рис. 4.10. Окно диалога *Список*

2) выполнить команду меню *Формат*→*Список* – откроется окно диалога *Список* (рис. 4.10), где нужно выбрать вкладку *Маркированный* или *Нумерованный*;

3) выбрать нужный формат из предлагаемых;

4) при необходимости для нумерованного списка изменить порядок нумерации (переключатель *начать заново*);

5) ОК.

Кроме того, с маркированными или нумерованными списками можно выполнить следующие действия:

- задать в качестве маркера уникальный символ или рисунок;
- изменить формат нумерованного списка;
- задать для маркеров или номеров списка формат, отличный от формата текста;
- изменить отступ между маркером или номером и текстом.

Алгоритм:

- 1) выделить список, который необходимо изменить;
- 2) выполнить команду меню *Формат*→*Список* – откроется окно диалога *Список* (см. рис. 4.10), где нужно выбрать вкладку *Маркированный* или *Нумерованный*;

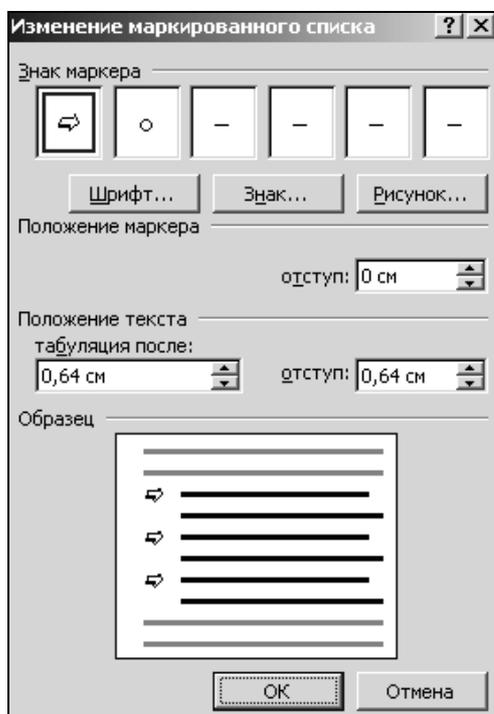
3) нажать кнопку *Изменить* – откроется окно диалога *Изменение маркированного списка* или *Изменение нумерованного списка* (рис. 4.11);

4) в этом окне задать необходимые параметры.

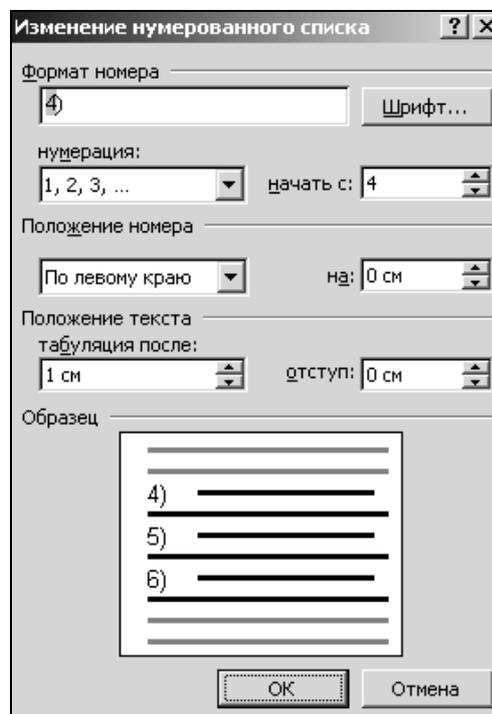
Многоуровневые списки. Многоуровневые списки используются для организации иерархической структуры нумерации. Такие списки могут содержать до девяти уровней иерархии и позволяют комбинировать маркированные и нумерованные абзацы. В многоуровневых списках номер или маркер каждого абзаца изменяется в соответствии с уровнем отступа от левого поля.

Алгоритм:

- 1) набрать элементы списка (проследить, чтобы каждый элемент списка находился в отдельном абзаце);
- 2) выделить список;
- 3) открыть вкладку *Многоуровневый* диалогового окна *Список*;
- 4) выбрать требуемый стиль многоуровневой нумерации из предлагаемых;
- 5) ОК;
- 6) выполнить одно из следующих действий:
 - чтобы переместить элемент на более низкий уровень иерархии, выделить нужный элемент и нажать кнопку *Увеличить отступ* ,
 - чтобы переместить элемент на более высокий уровень иерархии, выделить нужный элемент и нажать кнопку *Уменьшить отступ* .



а)



б)

Рис. 4.11. Окно диалога: а – *Изменение маркированного списка*; б – *Изменение нумерованного списка*

4.2. Таблицы

Таблица состоит из строк и столбцов, на пересечении которых образуются ячейки. Ячейки могут содержать текст (один или несколько абзацев), числа, формулы и рисунки. Использование формул позволяет создавать несложное подобие электронных таблиц.

4.2.1. Создание таблиц

Таблицы простой структуры создаются двумя способами.

1. При помощи кнопки *Добавить таблицу*  (внешний вид и назначение этой кнопки меняются в зависимости от ситуации). После щелчка по кнопке *Добавить таблицу* появляется прототип таблицы (4×5) для определения количества ячеек. Удерживая нажатой левую кнопку мыши, нужно выделить необходимое количество ячеек, затем отпустить кнопку мыши. Как только кнопка мыши будет отпущена, размер таблицы зафиксируется, и она будет помещена в документ. Изначально все ячейки новой таблицы пустые и имеют одинаковый размер.

2. При помощи команды меню *Таблица→Вставить→Таблица*. В появившемся диалоговом окне *Вставка таблицы* (рис. 4.12) в соответствующих полях нужно указать количество

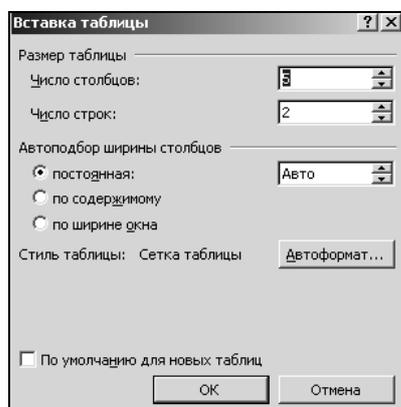


Рис. 4.12. Окно диалога *Вставка таблицы*

вующих полях нужно указать количество строк и столбцов будущей таблицы, в группе *Автоподбор ширины столбцов* выбрать параметры для настройки ширины столбцов таблицы. Если таблица сложная, т.е. содержит разное число столбцов в строках и/или разное число строк в столбцах, то для создания такой таблицы рекомендуется воспользоваться либо кнопкой *Таблицы и границы* , либо командой меню *Таблица→Нарисовать таблицу*.

В обоих случаях появится панель инструментов *Таблицы и границы*, а указатель мыши примет вид карандаша. Далее:

- 1) в поле *Тип линии*  выбрать линию нужного вида;
- 2) в поле *Толщина линии*  установить требуемую толщину линии;
- 3) выбрать цвет линии, используя кнопку *Цвет границы* ;
- 4) нарисовать таблицу:
 - удерживая левую кнопку мыши нажатой, протащить «карандаш» по диагонали – обозначится внешняя граница таблицы;

- поделить таблицу на строки и столбцы, проводя «карандашом» вертикальные и горизонтальные линии;

5) чтобы удалить одну или несколько линий, нажать кнопку *Ластик*  на панели инструментов *Таблицы и границы*, а затем щелкнуть линию, которую требуется удалить.

4.2.2. Форматирование таблиц

Microsoft Word позволяет выполнять с таблицами следующие операции *форматирования*:

- разбиение одной ячейки на несколько ячеек;
- объединение нескольких ячеек в одну ячейку;
- менять направление текста;
- менять выравнивание текста по горизонтали и вертикали;
- менять вид границ как отдельных ячеек, так и таблицы в целом и др.

Эти действия можно выполнять с помощью соответствующих кнопок, расположенных на панели инструментов *Таблицы и границы*, а также с помощью команд меню *Таблица*.

В существующих таблицах можно добавлять и удалять различные элементы. Так, для вставки строк нужно:

- 1) выделить строки, перед которыми или после которых будут вставлены новые (выделяется столько строк, сколько нужно вставить);
- 2) выполнить команду меню *Таблица*→*Вставить*→*Строки выше* (*Строки ниже*).

Для удаления строк нужно выделить удаляемые строки, затем выполнить команду меню *Таблица*→*Удалить*→*Строки*.

Для вставки столбцов нужно:

- 1) выделить столбцы, левее или правее от которых будут вставлены новые (выделяется столько столбцов, сколько нужно вставить);
- 2) выполнить команду меню *Таблица*→*Вставить*→*Столбцы слева* (*Столбцы справа*).

Для удаления столбцов нужно выделить столбцы, которые необходимо удалить, затем выполнить команду меню *Таблица*→*Удалить*→*Столбцы*.

К таблицам простой структуры в некоторых случаях целесообразно применять *автоформат*.

Алгоритм:

- 1) поместить текстовый курсор внутрь таблицы;
- 2) выполнить команду меню *Таблица*→*Автоформат таблицы*;
- 3) в диалоговом окне *Автоформат таблицы* выбрать стиль форматирования таблицы в списке *Стили таблиц*;
- 4) ОК.

4.3. Использование графических объектов в документах Microsoft Word

Текстовый процессор Microsoft Word обладает значительными возможностями по работе с объектами нетекстовой природы. Среди таких объектов могут быть стандартные объекты, созданные другими программами (рисунки, анимационные и звуковые клипы и многое другое), а также объекты, созданные средствами самого текстового процессора. В частности, программа позволяет создавать и встраивать геометрические фигуры, художественные заголовки, диаграммы, формульные выражения, заготовленные векторные иллюстрации, в том числе и анимированные, т.е. в ней имеются средства, отдаленно напоминающие средства специализированных графических редакторов.

4.3.1. Общие приемы работы с графическими объектами

Управление графическими объектами можно выполнять с помощью:

- 1) мыши;
- 2) панели инструментов, соответствующей типу объекта (она открывается автоматически, когда объект выделен);
- 3) диалогового окна *Формат объекта*, которое открывают из контекстного меню объекта.

С помощью мыши можно управлять только *размером, поворотом и положением* объекта на странице. Размер и поворот объекта мышью изменяются перемещением соответствующих маркеров объекта, а положение объекта на странице изменяется перетаскиванием самого объекта.

С помощью панели инструментов управляют индивидуальными свойствами объектов (у разных типов объектов они различны), а с помощью диалогового окна *Формат объекта* управляют наиболее общими свойствами объектов всех типов.

Окно диалога *Формат объекта*, вкладка *Размер*. На этой вкладке определяются размер и поворот объекта путем задания необходимых значений в полях *высота, ширина и поворот* (рис. 4.13).

Окно диалога *Формат объекта*, вкладка *Положение*. Позволяет управлять положением объекта на странице (рис. 4.14).

На вкладке *Положение* в группе *Обтекание* можно определить характер взаимодействия объекта с окружающим текстом, т.е. выбрать вариант обтекания:

- *В тексте* – объект вставляется в текстовую строку на правах графического символа;

- *Вокруг рамки* – текст располагается вокруг воображаемой прямоугольной рамки, охватывающей весь контур объекта;
- *По контуру* – текст плавно обтекает контур объекта;
- *Перед текстом* – текст и объект располагаются на разных слоях, причем объект лежит выше и закрывает текст (вставка без обтекания);
- *За текстом* – текст и объект располагаются на разных слоях, но объект лежит в нижнем слое и перекрывается текстом (вставка без обтекания).

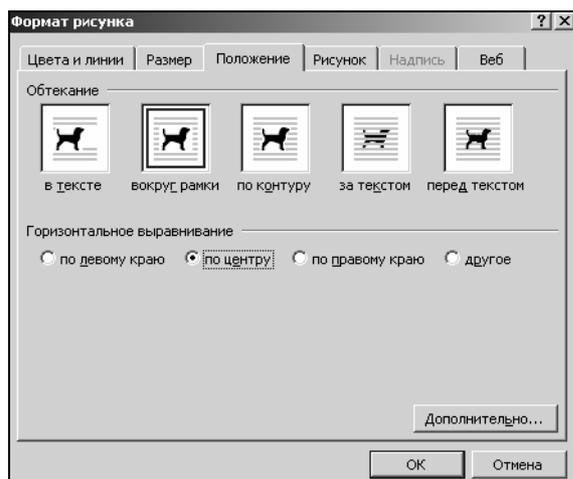


Рис. 4.13. Окно диалога *Формат объекта*, вкладка *Положение*

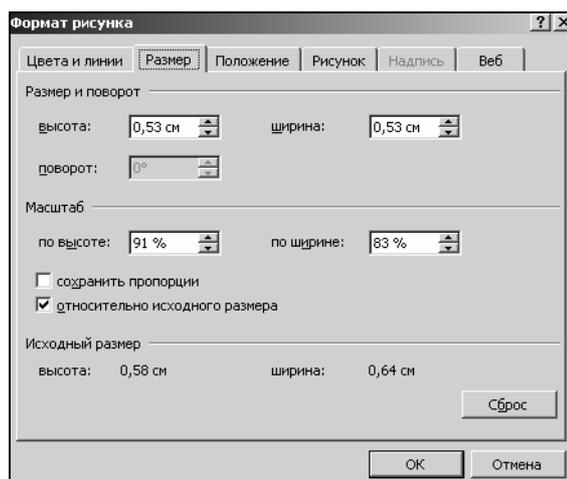


Рис. 4.14. Окно диалога *Формат объекта*, вкладка *Размер*

На этой же вкладке в группе *Горизонтальное выравнивание* можно задать параметры выравнивания объекта по горизонтали (*по левому краю, по центру, по правому краю, другое*). Варианты *По левому краю* и *По правому краю* обычно используют при обтекании *По контуру* или *Вокруг рамки*.

Вариант *По центру* часто сочетают с обтеканием *Сверху и снизу*, а вариант *другое* соответствует ручному размещению объекта перетаскиванием с помощью мыши. Для задания дополнительных параметров взаимодействия текста с графическим объектом следует перейти в окно диалога *Дополнительная разметка* (рис. 4.15), нажав кнопку *Дополнительно* на вкладке *Положение*.

Вкладка *Цвета и линии*. Большинство объектов, создаваемых средствами самой программы Microsoft Word, имеют *векторную природу*, т.е. в их основе лежат простейшие геометрические фигуры – *линии*. Эти линии, в свою очередь, имеют собственные свойства: *толщину, цвет и тип*. Управление этими свойствами выполняют с помощью элементов управления вкладки *Цвета и линии* диалогового окна *Формат объекта* (рис. 4.16).

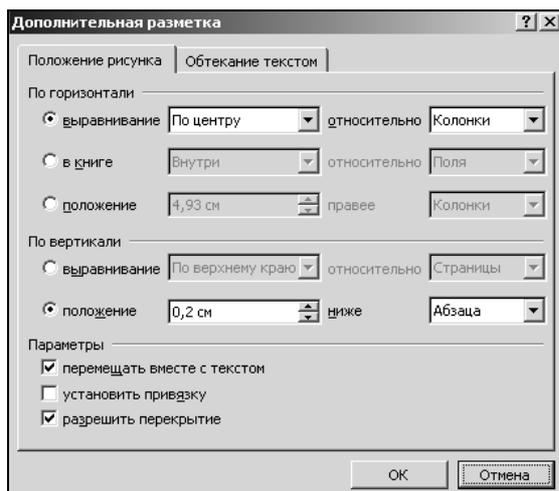


Рис. 4.15. Окно диалога *Дополнительная разметка*

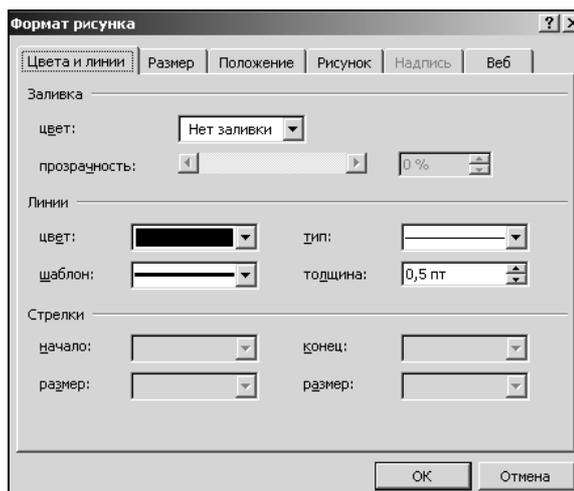


Рис. 4.16. Окно диалога *Формат объекта*, вкладка *Цвета и линии*

На этой вкладке для незамкнутых линий в группе *Линии* можно выбрать цвет линии из предлагаемой палитры, шаблон (сплошная или пунктирная), тип (одинарная, двойная, тройная) и точно задать толщину линии (в пт). Для замкнутых линий в группе *Заливка* можно выбрать цвет заливки из предлагаемой палитры и задать ее *Прозрачность* (т.е. через покрашенные контуры может просматриваться текст или объект нижележащего слоя).

Заливка может быть *простой* и *комбинированной*. Простая заливка – *одноцветная*. Цвет заливки может быть одним из стандартных, имеющих в палитре, или одним из дополнительных (выбирается в палитре с помощью кнопки *Другие цвета*).

Комбинированная заливка существует четырех видов:

- *градиентная заливка* – многоцветная заливка, при которой осуществляется плавный переход между заданными цветами;
- *текстурная заливка* – это заливка, воспроизводящая нерегулярную текстуру. Обычно используется для имитации поверхности материала;
- *заливка узором* – заливка изображением, имеющим регулярный характер;
- *заливка изображением-картой* – замкнутый контур заполняется специально подготовленным графическим изображением.

Для выбора метода комбинированной заливки используется кнопка *Способы заливки* в палитре цветов. Она открывает диалоговое окно *Способы заливки*, имеющее четыре вкладки: *Градиентная*, *Текстура*, *Узор* и *Рисунок*.

4.3.2. Взаимодействие графических объектов друг с другом

На одной странице может располагаться несколько графических объектов. Если для них определен одинаковый стиль обтекания текстом, то может быть разрешено перекрытие таких объектов друг другом. Для разрешения перекрытия следует выполнить команду контекстное меню объекта→*Формат объекта*→окно диалога *Формат объекта*→вкладка *Положение*→кнопка *Дополнительно*→окно диалога *Дополнительная разметка*→вкладка *Положение рисунка*→в группе *Параметры* установить флажок *Разрешить перекрытие*.

Также для управления взаимным расположением нескольких объектов могут использоваться следующие операции:

– *группирование* – для объединения нескольких расположенных на странице объектов в один комплексный объект, свойствами которого можно управлять также как свойствами простейших объектов. Для того чтобы сгруппировать объекты, их надо выделить мышью при нажатой клавише Shift и в контекстном меню группы выбрать команду *Группировка*→*Группировать*; чтобы разгруппировать объекты и получить доступ к каждому из них, надо выделить комплексный объект и выполнить команду контекстного меню *Группировка*→*Разгруппировать*;

– *задание порядка следования* – если на странице документа размещается несколько объектов, то предполагается, что у каждого объекта есть свой *слой*. По умолчанию порядок следования слоев связан с порядком создания объектов, т.е., те объекты, которые были созданы раньше, лежат на нижних слоях и могут перекрываться объектами, созданными позже. Для того чтобы изменить текущее положение объекта, следует в контекстном меню объекта выбрать команду *Порядок* – откроется дополнительное меню, средствами которого можно поднять объект на передний план, переместить на задний план, сместить на один слой вверх или вниз и задать положение объекта относительно текста;

– *выравнивание* – это средство используется, если объекты, составляющие композицию, не перекрывают друг друга. Для выравнивания нескольких объектов между собой их следует выделить при нажатой клавише Shift, вывести на экран панель инструментов *Рисование*. На этой панели нажать кнопку *Действия* – в открывшемся меню выбрать команду *Выровнять/распределить*. В появившемся дополнительном меню выбрать нужный вид горизонтального или вертикального выравнивания из предлагаемых.

Существует шесть методов выравнивания. Им соответствуют три команды горизонтального выравнивания (*По левому краю, По правому краю, По центру*) и три команды выравнивания вертикального (*По верхнему краю, По нижнему краю, По середине*). Следует обратить внимание на особенность действия команд выравнивания. Так, например, если два объекта выравниваются по нижнему полю, значит, они выравниваются по нижнему полю нижнего объекта. Выравнивание по правому полю – это выравнивание по правому полю самого правого объекта из числа выделенных и так далее. Если необходимо выполнить выравнивание относительно краев страницы, следует предварительно выполнить команду: кнопка *Действия* панели инструментов *Рисование*→*Выровнять/распределить*→установить флажок *Относительно страницы*;

– *распределение* – сущность этой операции состоит в том, что между объектами устанавливаются равные интервалы по горизонтали и/или вертикали. Распределение объектов выполняется командой: кнопка *Действия* панели инструментов *Рисование*→*Выровнять/распределить*→*Распределить по горизонтали* или *Распределить по вертикали*. Команда распределения выполняется не менее чем для трех объектов.

4.3.3. Создание художественных надписей

Для создания художественных графических надписей, например заголовков, в текстовом процессоре Microsoft Word есть специальное средство – *WordArt*. Доступ к нему осуществляется через специальную панель инструментов *WordArt* (меню *Вид*→*Панели инструментов*→*WordArt*). Вся работа производится с помощью кнопок этой панели (рис. 4.17).



Рис. 4.17. Панель инструментов *WordArt*

Ввод текста графического заголовка. Текст будущего заголовка можно ввести двумя способами:

- 1) выделить любой фрагмент текста в документе и использовать его в качестве заготовки;
- 2) ввести текст вручную.

И в том и в другом случае надо использовать кнопку *Добавить объект WordArt* . Если какой-то текст в документе был предварительно выделен, он будет передан программному средству *WordArt*, а если текст выделен не был, будет предоставлена возможность ввести его вручную.

После щелчка на кнопке *Добавить объект WordArt* открывается диалоговое окно *Коллекция WordArt* с палитрой стилей оформления (рис. 4.18).

На первом этапе можно принять любой стиль и создать базовый заголовок. Далее его можно изменить посредством редактирования. Выбранный стиль утверждается щелчком на кнопке ОК, после чего открывается диалоговое окно *Изменение текста WordArt* (рис. 4.19). В этом окне можно ввести текст заголовка, отредактировать его, задать гарнитуру, размер и начертание шрифта. Избранные параметры утверждаются щелчком на кнопке ОК, и объект вставляется в документ.

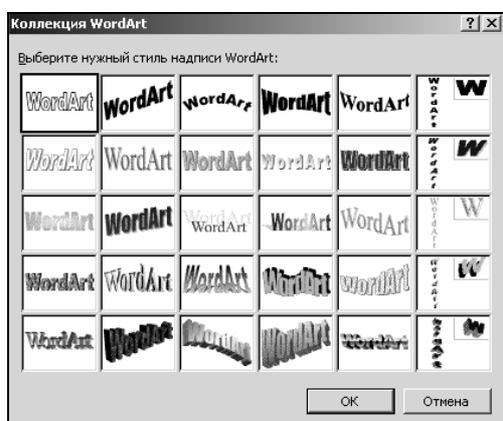


Рис. 4.18. Окно диалога
Коллекция WordArt

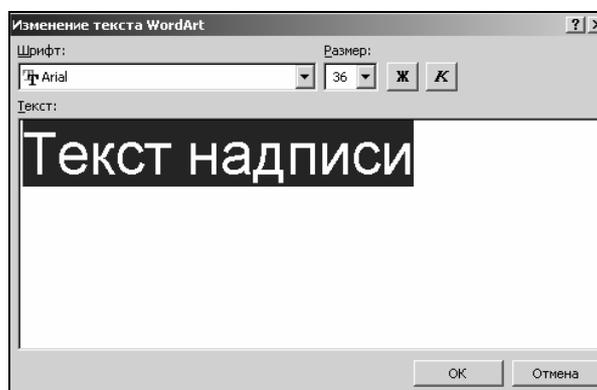


Рис. 4.19. Окно диалога
Изменение текста WordArt

Управление заголовком в составе документа. Средства управления находятся в диалоговом окне *Формат объекта WordArt*, которое открывают либо с помощью одноименной команды контекстного меню, либо с помощью одноименной кнопки на панели инструментов WordArt .

Редактирование объекта WordArt в составе документа. Когда установлено положение графического заголовка в документе, можно приступить к его визуальному редактированию путем настройки специальных свойств. Инструменты, необходимые для этого, представлены на панели инструментов *WordArt*. Они позволяют:

- изменить текст заголовка (кнопка *Изменить текст...*);
- изменить стиль оформления ;
- изменить форму заголовка ;
- повернуть изображение на произвольный угол (*Свободное вращение*);

- изменить параметры цвета, свойства линий и характер взаимодействия с окружающим текстом ;
- задать равную высоту всех символов в заголовке ;
- расположить текст по вертикали ;
- задать горизонтальное положение заголовка относительно страницы ;
- изменить расстояния между символами без изменения размеров объекта в целом .

4.3.4. Создание и редактирование графических объектов

Для работы с векторными графическими объектами служит панель инструментов *Рисование* (меню *Вид*→*Панели инструментов*→*Рисование*) (рис. 4.20).



Рис. 4.20. Панель инструментов *Рисование*

Основным средством этой панели, предназначенным для создания простейших объектов, является раскрывающийся список *Автофигуры*. В его категориях представлены заготовки для создания прямых и кривых линий, простейших геометрических фигур, фигурных стрелок и выносных линий, чертежных элементов для блок-схем и функциональных схем и т.д.

Добавление фигуры выполняется следующим образом:

- 1) на панели инструментов *Рисование* нажать кнопку *Автофигуры*;
- 2) в появившемся списке выбрать категорию, а затем – нужную фигуру;
- 3) щелчком мыши указать в документе место вставки фигуры.

Примечания:

1. Для того чтобы добавить круг или квадрат, нужно воспользоваться кнопкой *Овал*  или *Прямоугольник* .

2. Удобным средством, упрощающим создание прямолинейных геометрических фигур, является вспомогательная *координатная сетка*. Командой *Действия*→*Сетка* открывают диалоговое окно *Привязка к сетке*. В нем задают требуемые параметры сетки и только потом приступают к созданию или редактированию фигуры.

4.3.5. Создание надписей

Рисованные объекты могут сопровождаться текстовыми элементами (например, заголовками, буквенными или цифровыми обозначениями). Для создания текстовых элементов, присоединенных к рисованным объектам можно использовать элемент *Надпись*.

Алгоритм:

- 1) создать рисунок;
- 2) щелкнуть на кнопке *Надпись*  на панели инструментов *Рисование*;
- 3) щелкнуть левой кнопкой мыши в документе рядом с рисунком, который будет сопровождаться текстовой надписью – появится поле надписи;
- 4) в поле надписи ввести необходимый текст;
- 5) для завершения ввода текста щелкнуть мышью вне поля надписи.

На вкладке *Надпись* диалогового окна *Формат надписи* можно задать размеры внутренних полей между текстом и внешней рамкой поля надписи.

Для автофигур есть особое средство создания текстового оформления – текст может размещаться в поле автофигуры. Это выполняют командой *Добавить текст* из контекстного меню *Автофигуры*.

4.3.6. Работа с рисунками

Текстовый процессор Microsoft Word не имеет средств для создания рисунков, поэтому они вставляются как внешние объекты из файла, подготовленного другими средствами (графическим редактором, с помощью сканера, цифровой фотокамеры, графического планшета).

Существует два способа вставки рисунков:

- 1) из файла:
 - определить место в тексте, где требуется поместить рисунок;
 - меню *Вставка*→*Рисунок*→*Из файла* – откроется окно диалога *Добавление рисунка*;
 - в этом окне выбрать файл, содержащий нужный рисунок;
 - нажать кнопку *Вставить*;
- 2) из коллекции картинок:
 - определить место в тексте, где требуется разместить рисунок;
 - меню *Вставка*→*Рисунок*→*Картинки* – откроется область задач *Вставка картинки* (рис. 4.21);
 - в нижней части области задач *Вставка картинки* щелкнуть ссылку *Коллекция картинок* – откроется окно диалога *Коллекция картинок (Microsoft)* (рис. 4.22);

- в левой части окна выбрать нужную коллекцию, а в правой – нужный клип (файл, содержащий рисунок, звук, анимацию или фильм);
- выбранный клип скопировать в буфер обмена;
- вставить в документ клип из буфера обмена.

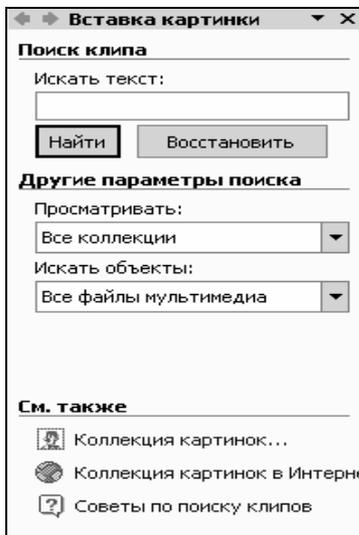


Рис. 4.21. Область задач *Вставка картинок*

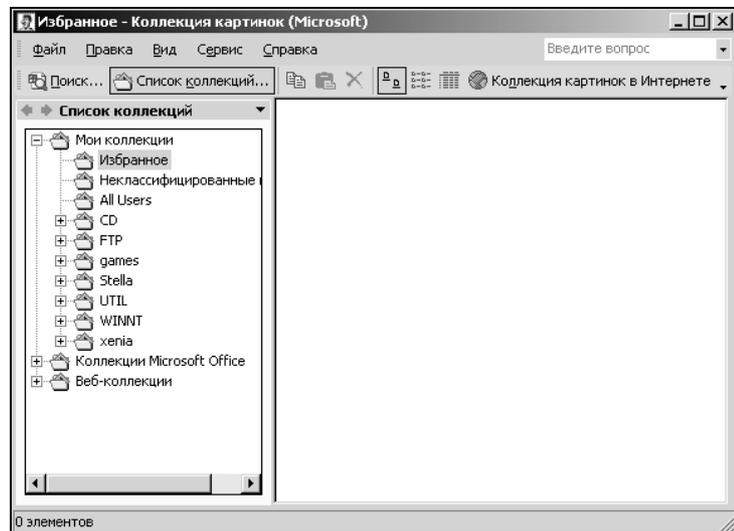


Рис. 4.22. Окно диалога *Коллекция картинок (Microsoft)*

Редактирование рисунков средствами текстового процессора невозможно, но возможно управление их положением и их визуализацией. Под *визуализацией* понимается метод отображения с учетом яркости и контрастности. Основная часть инструментов для настройки свойств изображений в текстовом документе находится на панели инструментов *Настройка изображения* (меню *Вид*→*Панели инструментов*→*Настройка изображения*). Панель инструментов *Настройка изображения* позволяет:

- увеличить или уменьшить контрастность;
- увеличить или уменьшить яркость;
- выполнить обрезку изображения;
- установить прозрачный цвет (можно назначить любой из цветов изображения (но только один) в качестве «прозрачного»; при размещении такого графического объекта поверх других объектов все объекты нижележащего слоя видны через те участки верхнего изображения, которые имеют цвет, назначенный прозрачным).

4.3.7. Специальные средства оформления

Эти средства оформления представлены кнопками на панели инструментов *Рисование*. Они позволяют:

- управлять цветом заливки, цветом контура и цветом текста;

- управлять толщиной сплошных линий и параметрами штриховых линий;
- преобразовывать линии в стрелки и управлять формой их начала и/или конца;
- создавать теневые и объемные эффекты.

Для каждой из указанных кнопок открывается палитра, позволяющая настроить результат действия эффекта. Если к объекту применен теневой или объемный эффект, то для редактирования этих эффектов используют следующие приемы:

- 1) в поле документа выделяют объект;
- 2) нажимают кнопку *Стиль тени*  или *Объем*  на панели инструментов *Рисование*;
- 3) в открывшейся палитре выбирают элемент управления *Настройка тени* или *Настройка объема* – открывается одноименная панель инструментов;
- 4) кнопками открывшейся панели выполняют изменение эффектов тени или объема.

4.4. Печать документа

Перед началом печати документа рекомендуется *проверить правильность расположения* на страницах рисунков, таблиц, заголовков и т.п. Такую проверку можно выполнить при одновременном отображении на экране нескольких страниц документа. Для этого следует использовать *режим предварительного просмотра*.

Печать документа выполняется по команде меню *Файл→Печать* – открывается диалоговое окно *Печать* (рис. 4.23).

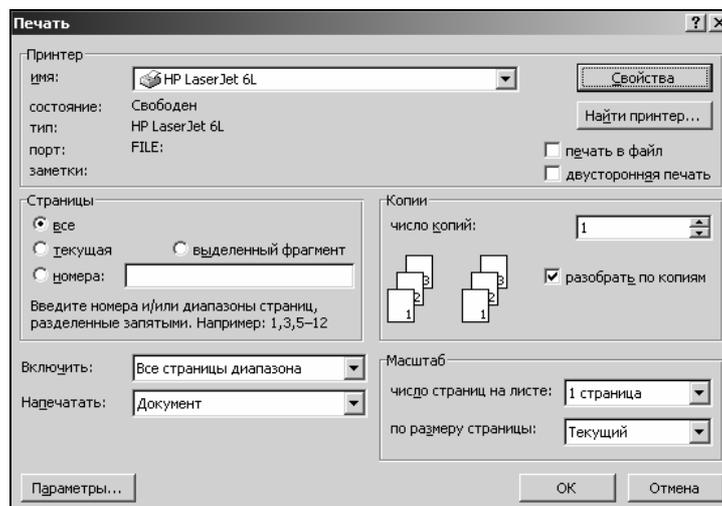


Рис. 4.23. Окно диалога *Печать*

К основным устанавливаемым параметрам печати относят:

– область *Принтер*. Содержит информацию о принтере, на котором будет печататься документ. Изменяя содержимое поля, пользователь имеет возможность выбрать для печати документа любой из принтеров сети;

– поле *Копии*. Указывает число бумажных копий документа;

– поле *Страницы*. Опция *все* задает печать всех страниц документа; опция *текущая* – печать страницы, на которой находится текстовый курсор; опция *выделенный фрагмент* позволяет распечатать заранее выделенный фрагмент; опция *номера* определяет номера страниц, которые требуется напечатать. Номера страниц разделяются запятыми; диапазон страниц указывается через дефис. Например, при печати страниц с номерами 2, 4, 5, 6 и 8 требуется набрать 2, 4-6, 8;

– поле *Включить*. В нем определяются страницы, которые будут распечатаны. При этом на печать можно вывести весь документ, только страницы с четными или с нечетными номерами.

5. ТАБЛИЧНЫЙ ПРОЦЕССОР MICROSOFT EXCEL

Microsoft Excel является мощным программным средством для работы с таблицами данных. С его помощью можно выполнять:

- автоматическую обработку зависящих друг от друга данных;
- автоматизацию итоговых вычислений и создание сводных таблиц;
- ведение простых БД и обработку записей;
- совместную работу с экономическими документами и подготовку табличных документов;
- построение диаграмм и графиков по имеющимся данным.

Основное рабочее пространство Microsoft Excel – это *рабочая книга*, представляющая собой набор *рабочих листов* (рис. 5.1).

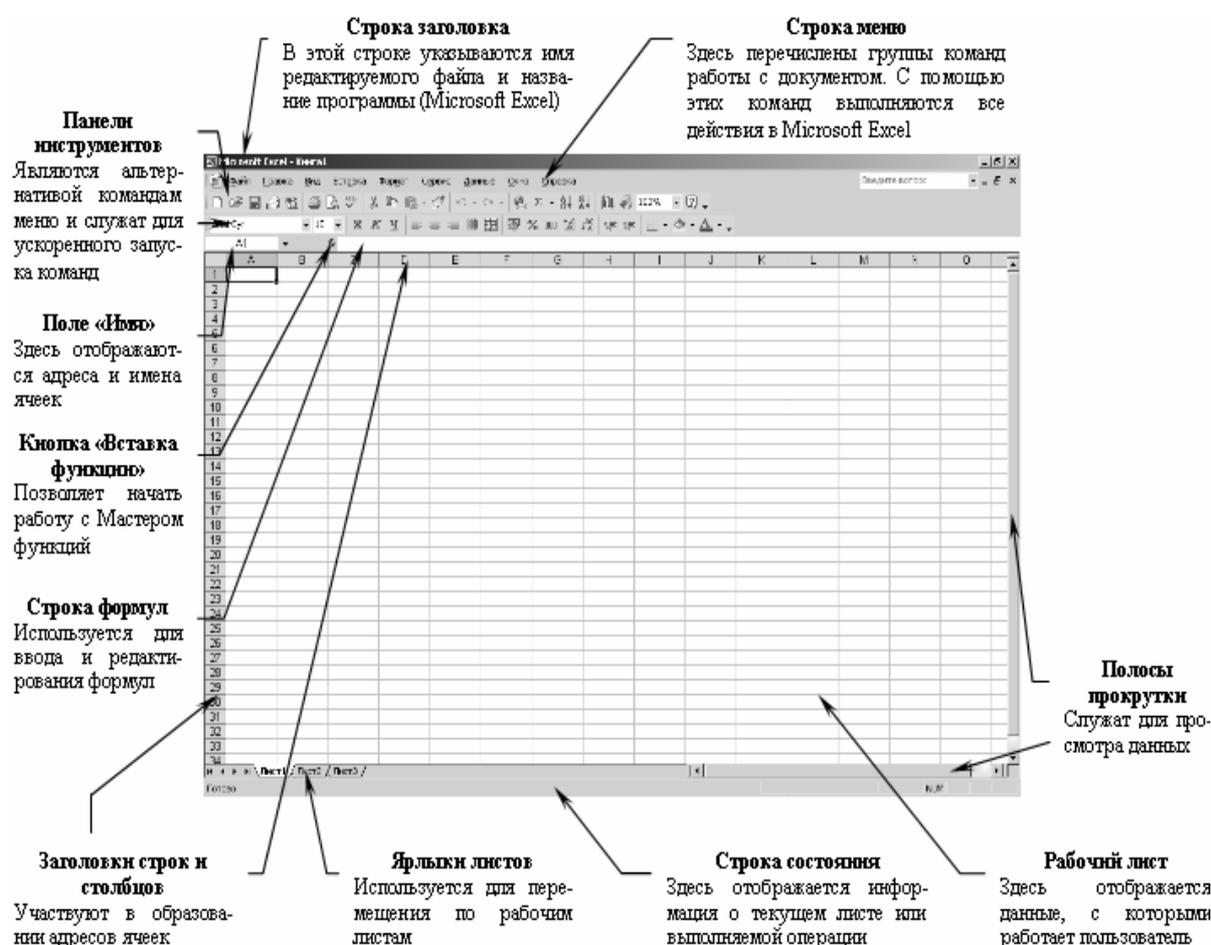


Рис. 5.1. Элементы окна Microsoft Excel

Рабочие листы имеют табличную структуру и могут содержать одну или несколько таблиц. Каждый рабочий лист имеет *название*, которое отображается на ярлычке листа, находящемся в его нижней части. Помимо рабочих листов, книга Microsoft Excel может содержать листы *диаграмм* и *модулей Visual Basic for Applications*.

Табличное пространство рабочего листа состоит из *строк* и *столбцов*. Столбцы озаглавлены латинскими буквами или двухбуквенными комбинациями (заголовки столбцов), строки последовательно нумеруются числами. Всего рабочий лист содержит 256 столбцов (от A до IV) и 65536 строк.

Минимальными элементами хранения данных являются *ячейки* таблицы. Обозначение отдельной ячейки сочетает в себе номера столбца и строки, пересечением которых она образована. Это сочетание называется *адресом* или *ссылкой* на ячейку (например, A1, DE234). Одна из ячеек всегда является активной и выделяется рамкой активной ячейки (*табличный курсор*). Ячейку можно активизировать (сделать текущей) с помощью мыши или клавиш управления курсором. Адрес текущей ячейки всегда отображается в *поле имени* с левой стороны *Строки формул* – над рабочим листом. Операции *ввода* и *редактирования* всегда производятся в активной ячейке.

На данные, расположенные в соседних ячейках, можно ссылаться в формулах, как на единое целое. Такую группу ячеек называют *диапазоном*. Диапазон ячеек обозначают, указывая через двоеточие номера ячеек, расположенных в левом верхнем и правом нижнем углах диапазона (например, A1:C15).

5.1. Ввод и редактирование данных

5.1.1. Перемещение по рабочему листу

Выполняется с помощью мыши (однократный щелчок левой кнопкой на нужной ячейке) или клавиш управления курсором (табл. 5.1):

Таблица 5.1

Сочетания клавиш для перемещения по рабочему листу

активизировать ячейку	Нажать
Слева	← или Shift + Tab
Справа	→ или Tab
Сверху, в предыдущей строке	↑ или Shift Enter
Снизу, в следующей строке	↓ или Enter
Выше на одну экранную страницу	Page Up
Ниже на одну экранную страницу	Page Down
У правой границы текущего диапазона	Ctrl + →
У левой границы текущего диапазона	Ctrl + ←
В начале рабочего листа	Ctrl + Home
На пересечении последних строки и столбца, которые содержат данные	Ctrl + End

При помощи полос прокрутки можно быстро просмотреть части рабочего листа, которые не поместились на экране, но в этом случае меняется вид рабочего листа, а текущей остается прежняя ячейка.

5.1.2. Выделение группы ячеек

Иногда при работе с данными возникает необходимость в выделении группы ячеек. Для этого MS Excel предоставляет следующие возможности.

Чтобы выделить несколько смежных ячеек, нужно щелкнуть на первой из них, затем нажать клавишу Shift и, не отпуская ее, щелкнуть на последней из выделяемых ячеек.

Можно выделить группу смежных ячеек, пользуясь только мышью. Для этого нужно нажать левую кнопку мыши на первой ячейке и, не отпуская ее, перетащить указатель мыши в последнюю ячейку.

Выделение группы смежных ячеек называется выделением *блока* или *диапазона ячеек*.

Для выделения несмежных ячеек, нужно щелкнуть на первой из них, а затем нажать клавишу Ctrl и, не отпуская ее, щелкать на других ячейках, которые необходимо выделить.

Для того чтобы выделить строку или столбец, нужно щелкнуть по заголовку соответствующей строки или столбца.

Для того чтобы выделить весь рабочий лист, следует щелкнуть на пересечении строки заголовков столбцов и столбца заголовков строк.

5.1.3. Ввод и редактирование данных

В ячейки рабочего листа можно вводить данные следующих типов: *текст*, *числа* или *формулы*. Для этого нужно выделить (активизировать) ячейку и набрать с клавиатуры данные. В процессе ввода Microsoft Excel автоматически распознает тип данных.

Все, что набирается с клавиатуры, появляется и в текущей ячейке, и в строке формул. Занести набранные данные в текущую ячейку можно, нажав клавишу Enter, или щелкнув мышью на другой ячейке. Отменить ввод данных можно нажатием клавиши Esc.

Редактировать данные в ячейках можно двумя способами:

1) можно выделить ячейку и, щелкнув в строке формул, отредактировать данные в строке формул;

2) редактировать данные прямо в ячейке. Для редактирования в ячейке нужно выполнить на ней двойной щелчок левой кнопкой мыши, а затем щелчком мыши поместить текстовый курсор на место, куда требуется внести изменения.

Чтобы удалить данные из ячейки, ее нужно выделить и нажать клавишу Del.

5.1.4. Ввод последовательностей данных в ячейки:

- 1) выделить первую из заполняемых ячеек;
- 2) ввести начальное значение для ряда значений;
- 3) ввести следующее значение в соседнюю ячейку, чтобы определить образец заполнения;
- 4) выделить ячейку или ячейки, содержащие начальные значения;
- 5) перетащить маркер заполнения (рис. 5.2) при нажатой левой кнопке мыши через заполняемые ячейки.



Рис. 5.2. Маркер заполнения

Для заполнения в возрастающем порядке перетащить маркер вниз или вправо. Для заполнения в убывающем порядке перетащить маркер вверх или влево.

Пример 1. Если требуется получить ряд 2, 3, 4, 5..., введите 2 и 3 в первые две ячейки. Если требуется получить ряд 2, 4, 6, 8..., введите 2 и 4. Если требуется получить ряд 2, 2, 2, 2..., то вторую ячейку можно оставить пустой. Чтобы задать тип ряда значений, перетащите маркер заполнения правой кнопкой мыши, а затем выберите соответствующую команду в контекстном меню.

Пример 2. Если начальное значение – дата «янв-2002», то для получения ряда «фев-2002», «мар-2002» и т.д. выберите команду *Заполнить по месяцам*, а для получения ряда «янв-2003», «янв-2004» и т.д. выберите команду *Заполнить по годам*.

Для того чтобы точно сформулировать условия заполнения ячеек, следует выполнить команду: меню *Правка* → *Заполнить* → *Прогрессия* – откроется окно диалога *Прогрессия* (рис. 5.3). Здесь выбрать тип прогрессии, величину шага и предельное значение. После щелчка на кнопке *ОК* Microsoft Excel автоматически заполнит ячейки в соответствии с заданными правилами.

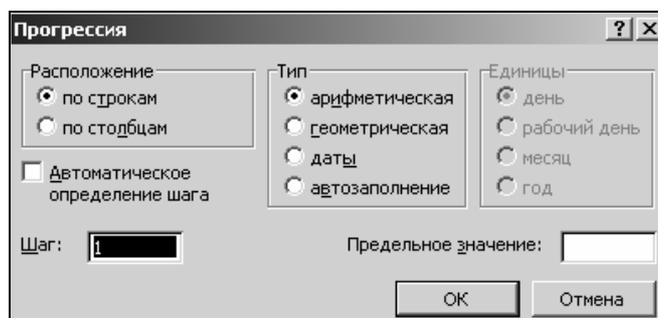


Рис. 5.3. Окно диалога *Прогрессия*

5.1.5. Настройка Автозаполнения:

- 1) выполнить команду меню *Сервис*→*Параметры* – откроется окно диалога *Параметры*, в этом окне выбрать вкладку *Списки* (рис. 5.4);
- 2) чтобы ввести новый список, выбрать *Новый список* из списка *Списки*;
- 3) ввести данные в поле *Элементы списка*, начиная с первого элемента. После ввода каждой записи нажимать клавишу *Enter*;
- 4) после того как список будет введен полностью, нажать кнопку *Добавить*.

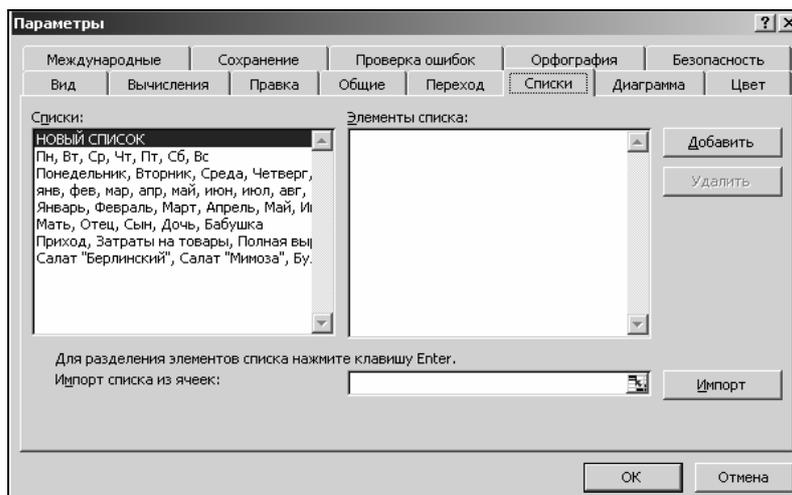


Рис. 5.4. Окно диалога *Параметры*, вкладка *Списки*

5.1.6. Переупорядочивание содержимого ячеек

Введенные данные, будь то текст, числа или формулы, не привязаны жестко к конкретным ячейкам. Данные можно копировать из одних ячеек в другие, добавлять и удалять ячейки, строки, столбцы и перемещать данные между ячейками.

Копирование и вставка данных в ячейки. Копировать данные в другое место рабочего листа можно: с помощью кнопок копирования и вставки на панели инструментов; с помощью соответствующих команд меню *Правка*; перетаскивая данные мышью.

Копирование с помощью мыши – самый быстрый способ, особенно если ячейки находятся на одном рабочем листе и расстояние между ними невелико.

Алгоритм:

- 1) выделить нужную ячейку или диапазон ячеек;
- 2) поместить указатель мыши на любом участке границы выделенного диапазона;

3) нажать и не отпускать клавишу `Ctrl`. При нажатии на клавишу `Ctrl`, если указатель мыши находится на границе диапазона, он помечается маленьким плюсом. Этот плюс указывает, что содержимое диапазона подлежит копированию, а не перемещению;

4) перетащить рамку диапазона на новое место, отпустить кнопку мыши и затем клавишу `Ctrl`. Копия выделенной информации будет вставлена в новый блок.

Копирование и вставка элементов ячеек. Копировать и вставлять можно не только ячейку целиком, но и отдельные элементы ее содержимого. Например, если в ячейке записана некоторая формула, то в новую ячейку можно перенести только результат вычислений.

Для того чтобы выборочно скопировать и вставить отдельные элементы ячейки, выполняют следующие действия:

- 1) выделить ячейку или диапазон ячеек;
- 2) скопировать выделенное в буфер обмена;
- 3) определить место вставки;
- 4) щелкнуть правой кнопкой мыши. В появившемся контекстном меню выбрать команду *Специальная вставка*. Откроется окно диалога *Специальная вставка* (рис. 5.5).
- 5) в этом окне определить параметры вставки данных путем выбора соответствующей опции;
- 6) щелкнуть на кнопке `OK`.

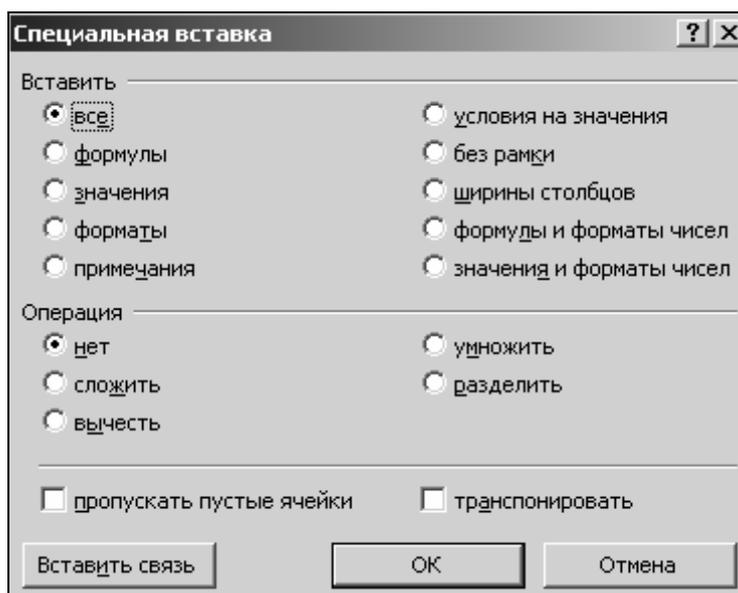


Рис. 5.5. Окно диалога *Специальная вставка*

Перемещение данных между ячейками. Переместить данные в другое место рабочего листа можно: с помощью кнопок *Вырезать* и *Вставить* на панели инструментов; с помощью соответствующих команд меню *Правка*; перетаскивая данные мышью.

Чтобы переместить содержимое ячейки с помощью мыши, достаточно выделить ячейку и перетащить ее рамку на новое место.

Вставка и удаление ячеек, строк и столбцов. При оформлении рабочих листов можно копировать ячейки в любое место рабочей книги. Если не хватает пространства для записи данных в конкретную область рабочей таблицы, можно добавлять или удалять ячейки, строки и столбцы. Когда происходит удаление ячеек рабочего листа, уничтожается их содержимое, а другие ячейки из правой или из нижней областей таблицы передвигаются на место удаленных. Вставить ячейку, строку или столбец можно с помощью команд *Ячейки*, *Строки*, *Столбцы* из меню *Вставка*, а удалить – с помощью команды *Удалить* из меню *Правка*.

Вставляя или удаляя строку или столбец, следует выделять их целиком, а не какую-то отдельную их часть. Выделить строку или столбец целиком можно, щелкнув на их заголовках.

Вставляя ячейку, строку или столбец, необходимо выделить ячейку, строку или столбец, расположенные непосредственно справа или внизу от позиции вставки. При вставке с помощью команд меню следует определить, какие из старых ячеек передвинуть, чтобы освободить пространство для новой, а в случае удаления – какие ячейки займут место удаленных. Вставлять, удалять и перемещать столбцы и строки можно так же, как и отдельные ячейки.

5.2. Работа с книгами Microsoft Excel

5.2.1. Перемещение по рабочей книге

Выбирать различные рабочие листы в рабочей книге можно следующими способами:

1) щелчком мыши на соответствующем ярлычке в нижней части рабочих листов;

2) с помощью клавиатуры: одновременное нажатие клавиш *Ctrl+Page Down* позволяет перейти на следующий лист, *Ctrl+Page Up* – на предыдущий.

Используя кнопки прокрутки ярлычков, находящихся в левом нижнем углу экрана , можно переместиться к ярлычку первого листа рабочей книги, перейти к ярлычку предыдущего или последующего листов, переместиться к ярлычку последнего листа (если в книге много листов).

5.2.2. Выделение рабочих листов

Можно выделить сразу несколько рабочих листов. Выделение выполняется аналогично выделению группы ячеек: выделение смежных листов выполняется щелчком мыши по первому и последнему листам при нажатой клавише Shift, а несмежных – щелчками мыши по нужным листам при нажатой клавише Ctrl.

Выделив несколько рабочих листов, можно вводить одни и те же данные одновременно в каждый из них. Данные появятся в соответствующих ячейках каждого из выделенных листов.

5.2.3. Манипулирование рабочими листами

Каждая новая рабочая книга состоит из трех рабочих листов с именами *Лист1*, *Лист2* и *Лист3*. Исходный состав рабочей книги можно изменить, добавляя, удаляя или переименовывая ее листы.

Вставка листов. Для вставки новых рабочих листов следует выполнить команду меню *Вставка→Лист* (добавится столько листов, сколько их было предварительно выделено).

Удаление листов. Для удаления рабочих листов следует выполнить команду меню *Правка→Удалить лист* (удалится столько листов, сколько их было предварительно выделено).

Переименование листов. Для переименования рабочего листа его нужно выделить, затем выполнить команду меню *Формат→Лист→Переименовать*, ввести новое имя поверх старого и нажать клавишу Enter или щелкнуть мышью вне ярлычка листа.

Перемещение и копирование листов внутри рабочей книги. Для выполнения этих операций можно использовать следующие действия:

- 1) перемещение выполняется обычным перетаскиванием ярлычка соответствующего листа на новое место;
- 2) копирование выполняется перетаскиванием ярлычка при нажатой клавише Ctrl.

5.3. Форматирование данных

5.3.1. Автоматическое форматирование данных

Этот вид форматирования позволяет быстро придать диапазону ячеек эстетичный вид. Microsoft Excel предлагает различные варианты форматирования, включая форматы для финансовых данных, учетных сведений, списков, а также цветные и трехмерные форматы.

Алгоритм:

- 1) выделить диапазон, который требуется отформатировать;
- 2) выполнить команду меню *Формат*→*Автоформат* – откроется окно диалога *Автоформат* (рис. 5.6).

3) в списке форматов выбрать наиболее подходящий. По умолчанию форматирование производится по всем элементам формата. Для частичного применения *Автоформата* нажать кнопку *Параметры* и снять флажки для элементов форматирования, которые не нужно применять.

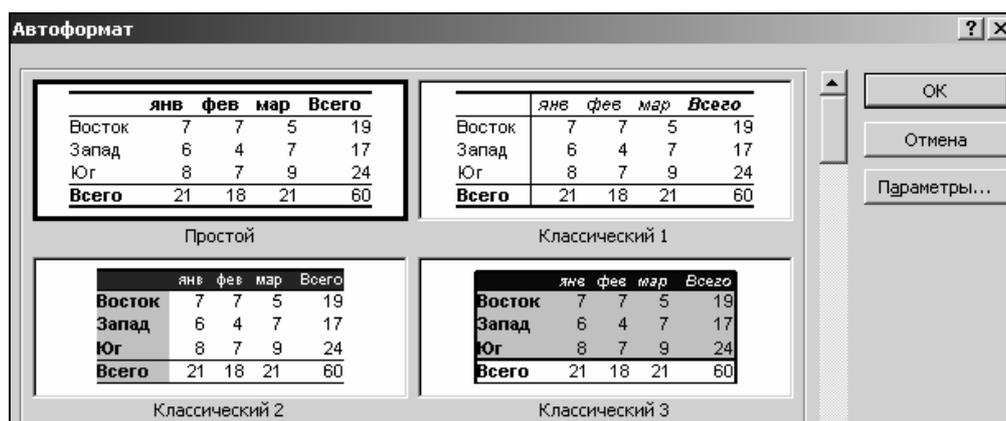


Рис. 5.6. Окно диалога *Автоформат*

Для удаления примененного *Автоформата* следует выполнить пункты 1) и 2), а затем выбрать формат с подписью *Нет*.

5.3.2. Копирование форматов в другие ячейки

Применить к данным формат, уже использованный в какой-то части рабочего листа, можно с помощью кнопки *Формат по образцу* .

Алгоритм:

- 1) выделить ячейку с нужным форматом;
- 2) щелкнуть на кнопке *Формат по образцу*;
- 3) указать область применения копируемого формата.

5.3.3. Форматирование с помощью панели инструментов *Форматирование*

Изменение формата чисел. Быстро задать стандартный формат для числа в ячейке можно при помощи следующих кнопок:

- 1) *Денежный формат* .
- 2) *Процентный формат* .
- 3) *Формат с разделителями* .

Каждый из этих форматов по умолчанию оставляет определенное количество десятичных знаков, которое может быть изменено кнопками *Уменьшить разрядность*  и *Увеличить разрядность* .

Изменение шрифта. Шрифты и их размер можно быстро менять, используя поля *Шрифт*  и *Размер шрифта*  на панели инструментов *Форматирование*. Каждый шрифт имеет три модификации: *Полужирный*, *Курсив* и *Подчеркнутый*, которые задаются соответственно кнопками , , .

Изменение выравнивания. По умолчанию Microsoft Excel автоматически выравнивает вводимый текст по левому, а числовые значения – по правому краю. Для изменения способа выравнивания данных в ячейках можно воспользоваться соответствующими кнопками на панели инструментов *Форматирование*:

- 1) *По левому краю* – ;
- 2) *По центру* – ;
- 3) *По правому краю* – .

Примечание. Если нужно, например, выровнять заголовок по центру нескольких столбцов, выделяются ячейки, относительно которых надо центрировать заголовок, и выполняется щелчок по кнопке выравнивания *Объединить и поместить в центре* .

Добавление рамок и изменение цветов. Отдельные ячейки или области листа можно оформить, выделив их рамками или выбрав различные цвета текста. Рамки могут выделять ячейки дополнительными линиями сверху, снизу, по бокам, а также вокруг ячеек. Ячейку можно закрасить подходящим цветом:

- 1) цвет текста можно изменить с помощью кнопки *Цвет шрифта* ;
- 2) границы ячеек можно установить с помощью кнопки *Границы* ;
- 3) закрасить ячейку можно с помощью кнопки *Цвет заливки* .

Алгоритм:

- 1) выделяется ячейка или диапазон ячеек;
- 2) выполняется щелчок по стрелке нужной кнопки;
- 3) в соответствующих палитрах щелчком мыши выбирается цвет или рамка.

Примечание. Изменять формат данных в ячейках можно также с помощью меню *Формат*→*Ячейки*→*окно Формат ячеек*. Вкладки этого диалогового окна предоставляют больше возможностей, чем кнопки на панели инструментов *Форматирование*. Здесь можно выбрать формат записи данных (количество знаков

после запятой, указание денежной единицы, способ записи даты и пр.), задавать направление текста и метод его выравнивания, определять шрифт и начертание символов, управлять отображением и видом рамок, задавать фоновый цвет.

5.3.4. Изменение высоты строк и ширины столбцов

При работе с длинными текстовыми заголовками, высокими шрифтами и записью чисел денежным стилем может оказаться, что не хватает стандартной ширины столбца или высоты строки. Поменять ширину столбцов и высоту строк можно с помощью мыши:

- перетащить правую границу заголовка столбца и/или нижнюю границу заголовка строки (перетаскиванием можно не только увеличивать, но и уменьшать ширину столбца и высоту строки);
- выполнить двойной щелчок на правой границе заголовка столбца и/или на нижней границе заголовка строки. Строка или столбец будут увеличены до размеров самого высокого символа и самого длинного текста в этих рядах ячеек.

5.4. Организация вычислений

При помощи Microsoft Excel удобно выполнять различные вычисления, оперируя данными, расположенными на рабочем листе. Результат вычислений определяется *формулами*, которые необходимо внести в ячейку рабочего листа. Microsoft Excel сам производит требуемые подсчеты и результат помещает в ячейку, содержащую формулу.

Формула может содержать числовые константы, ссылки на ячейки и функции Microsoft Excel, соединенные знаками арифметических операций: +, -, *, ^, /, %. Скобки позволяют изменять стандартный порядок выполнения действий. Если ячейка содержит формулу, то в рабочем листе отображается только текущий результат вычисления этой формулы. Для того чтобы увидеть саму формулу, надо выделить ячейку (сделать ее текущей) и посмотреть на запись, которая отображается в строке формул.

5.4.1. Автоматическое суммирование строк и столбцов

Автосуммирование выполняется с помощью кнопки *Автосумма*  на панели инструментов *Стандартная*.

Автосуммирование можно использовать для трех типов задач: обнаружить и просуммировать данные в строках или столбцах ближайшего к текущей ячейке диапазона; просуммировать данные в любом выделенном диапазоне ячеек; добавить итоговые суммы к ряду, содержащему частичные суммы.

Для того чтобы автоматически просуммировать данные в ближайшем к текущей ячейке диапазоне, нужно щелкнуть на кнопке *Автосумма* и убедиться, что требуемый диапазон выделен правильно (при необходимости выделение можно скорректировать) и нажать клавишу Enter. Просуммировать данные в любом диапазоне можно, выделив его, щелкнуть на кнопке *Автосумма*.

5.4.2. Составление элементарных формул

Элементарные формулы могут состоять только из арифметических операторов и адресов ячеек. В Microsoft Excel ввод формул в ячейках необходимо начинать со знака равенства =, а завершать нажатием клавиши Enter. Вместо чисел в формулах используются адреса ячеек, иначе говоря, *ссылки на ячейки*. Это означает, что результат расчета зависит от того, какие числа находятся в ячейках, участвующих в вычислении. Таким образом, ячейка, содержащая формулу, является *зависимой*. Значение в зависимой ячейке подлежит пересчету всякий раз, когда изменяются значения в ячейках, на которые указывают ссылки, входящие в формулу.

Ссылку на ячейку можно задать следующими способами:

- 1) ввести адрес ячейки вручную;
- 2) щелкнуть на нужной ячейке или выделить нужный диапазон, адрес которого нужно внести в формулу. Ячейка или диапазон при этом выделяются цветной пунктирной рамкой (рис. 5.7).

Для редактирования формулы следует дважды щелкнуть на соответствующей ячейке. При этом ячейки (диапазоны), от которых зависит значение формулы, выделяются на рабочем листе цветными рамками, а сами ссылки отображаются в ячейке и в строке формул тем же цветом. Это облегчает редактирование и проверку правильности формул.

	A	B	C
1	=B5+C5		
2			
3			
4			
5		5	3
6			

Рис. 5.7. Ввод формулы

Примеры записи формул:

- 1) = (B4 + D2) / F5;
- 2) = A1 * B2;
- 3) = C3 ^ 3.

5.4.3. Использование Мастера функций при составлении формул

Иногда в вычислениях приходится использовать формулы, содержащие числа и функции. *Функции* – заранее определенные формулы, которые выполняют вычисления по заданным величинам, называемым *аргумента-*

ми, и в указанном порядке. Эти функции позволяют выполнять как простые, так и сложные вычисления.

Примеры записи формул, содержащих функции:

1) =СУММ (С22:С26) – в этой формуле используется функция СУММ для того чтобы сложить значения в диапазоне ячеек С22:С26 (диапазон С22:С26 – аргумент). Эта формула аналогична формуле =(С22+С23+С24+ С25+С26);

2) =КОРЕНЬ(В10) – извлекает квадратный корень из числа, содержащегося в указанной ячейке;

3) =СЕГОДНЯ() – вставляет в ячейку текущую дату;

4) =ТДАТА() – вставляет в ячейку текущую дату и время.

Примечание. Функции СЕГОДНЯ() и ТДАТА() не требуют указания аргументов.

Некоторые функции, например, статистические или финансовые, используют несколько аргументов. В таких случаях аргументы отделяются друг от друга точкой с запятой.

Существуют различные типы аргументов: число, текст, логическое значение (*ИСТИНА* и *ЛОЖЬ*), массивы, значение ошибки (например, #Н/Д), или ссылки на ячейку. В качестве аргументов могут использоваться константы, формулы или функции.

Чтобы упростить ввод функций при создании формул, содержащих функции, рекомендуется использовать диалоговое окно *Мастер функций*.

Алгоритм:

1) выделить ячейку, в которой нужно разместить формулу, содержащую функцию;

2) щелкнуть по кнопке *Вставка функции*  в строке формул. Откроется диалоговое окно *Мастер функций* (рис. 5.8);

3) в списке *Категория* выбрать нужную категорию функций. В поле *Выберите функцию* появится список функций выбранной категории;

4) из списка функций выбрать нужную функцию и щелкнуть на кнопке ОК. Откроется окно *Аргументы функции* (рис. 5.9). При вставке функции в формулу диалоговое окно *Мастер функций* отображает имя функции, все ее аргументы, описание функции и каждого аргумента, текущий результат функции и всей формулы;

5) в окне *Аргументы функции* ввести требуемые аргументы в соответствующие поля либо путем ввода с клавиатуры, либо выделением нужных ячеек или диапазонов ячеек непосредственно на рабочем листе. Если

название аргумента указано полужирным шрифтом, то этот аргумент является *обязательными* и соответствующее поле должно быть заполнено. Аргументы, названия которых указаны обычным шрифтом, можно опустить;

б) щелкнуть на кнопке ОК.

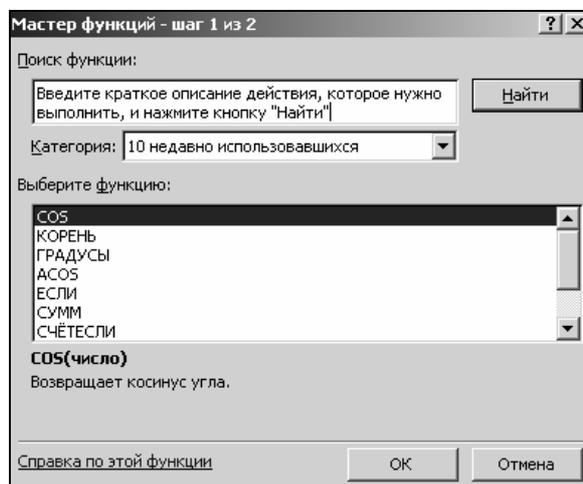


Рис. 5.8. Окно диалога *Мастер функций*

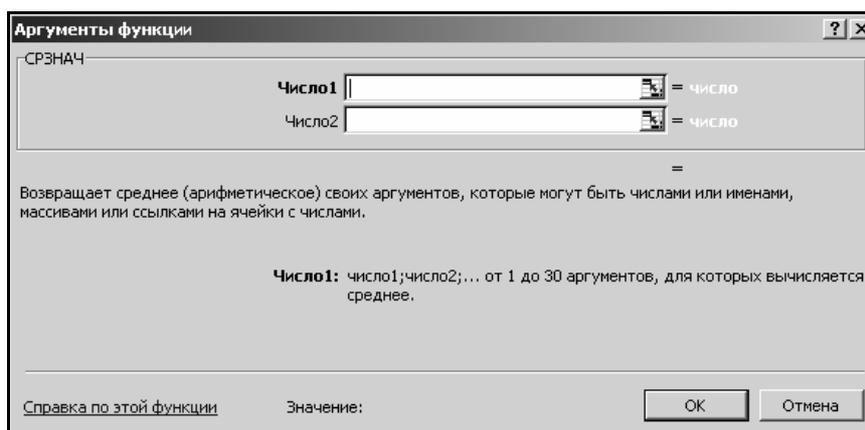


Рис. 5.9. Окно диалога *Аргументы функции*

Все диалоговые окна программы Microsoft Excel, которые требуют указания номеров или диапазонов ячеек, содержат кнопки, присоединенные к соответствующим полям. При щелчке на такой кнопке диалоговое окно сворачивается до минимально возможного размера, что облегчает выбор нужной ячейки (диапазона) с помощью мыши (рис. 5.10).

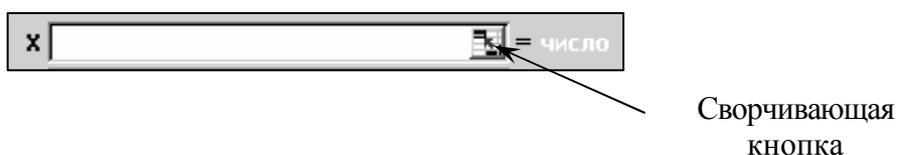


Рис. 5.10. Сворачивающая кнопка в окне *Аргументы функции*

5.4.4. Копирование формулы в диапазон ячеек

Копирование формулы в диапазон ячеек осуществляется так же, как и копирование данных. Для того чтобы внести формулу в диапазон, нужно выделить ячейку, содержащую формулу, и перетащить маркер заполнения вниз, вверх, вправо или влево на сколько требуется.

5.4.5. Создание формул с относительными и абсолютными адресами

Относительные адреса. *Относительный адрес* в формуле, например А1, основан на относительной позиции ячейки, содержащей формулу, и ячейки, на которую указывает адрес. При изменении позиции ячейки, содержащей формулу, изменяется и адрес, т.е. при копировании формулы адрес автоматически корректируется.

Пример 1. Пусть ячейка В2 содержит ссылку на ячейку А1. Тогда при копировании относительного адреса А1 из ячейки В2 в ячейку В3, она автоматически изменяется с = А1 на = А2 (рис. 5.11).

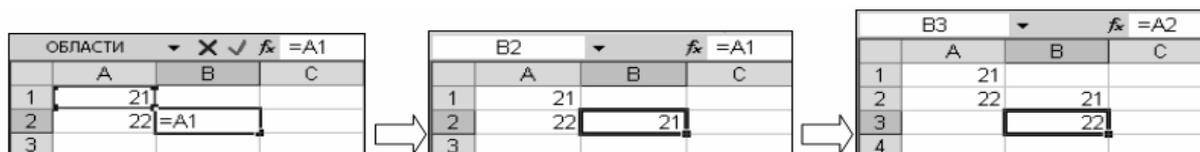


Рис. 5.11. Использование относительных ссылок

Абсолютные адреса. Запись *абсолютных адресов* содержит знаки доллара (например, \$A\$1). Абсолютный адрес ячейки в формуле всегда ссылается на ячейку, расположенную в определенном месте. При изменении позиции ячейки, содержащей формулу, абсолютный адрес не изменяется, т.е. при копировании формулы абсолютный адрес не корректируется.

Пример 2. Пусть ячейка В2 содержит ссылку на ячейку \$A\$1. Тогда при копировании абсолютного адреса \$A\$1 из ячейки В2 в ячейку В3, он остается прежним (рис. 5.12).

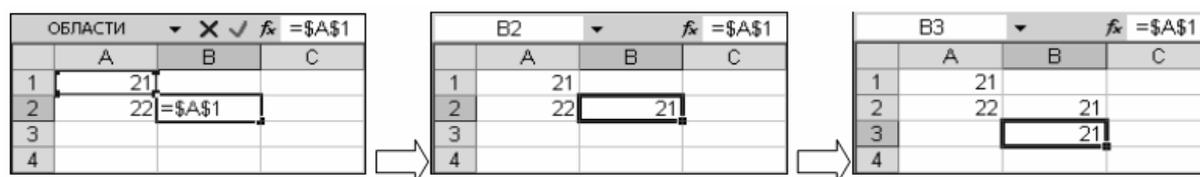


Рис. 5.12. Использование абсолютных ссылок

Смешанные адреса. В формулах можно также использовать *смешанные адреса*, которые задают столбец относительно, а строку абсолютно, или наоборот. При изменении позиции ячейки, содержащей формулу, относительный адрес изменяется, а абсолютный – нет, т.е. при копировании формулы относительная ссылка автоматически корректируется, а абсолютная ссылка не корректируется.

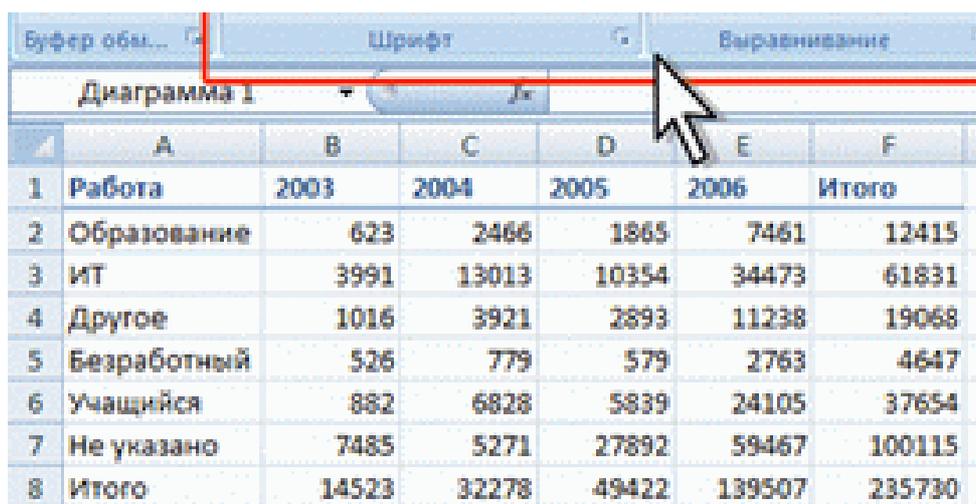
Пример 3. Абсолютный адрес столбцов приобретает вид \$A1, \$B1 и т.д. Абсолютный адрес строк приобретает вид A\$1, B\$1.

5.5. Диаграммы

Диаграммы являются средством наглядного представления данных и облегчают выполнение сравнений, выявление закономерностей и тенденций данных. Можно составить графики, показывающие изменения величин во времени, или диаграммы, определяющие, какая доля целого приходится на отдельные его части. В составленные диаграммы можно вносить исправления: переупорядочивать данные или добавлять ранее неучтенные. С помощью *Мастера диаграмм* можно легко показать изменения данных в динамике, что актуально для выступлений и докладов. При изменении данных в рабочей таблице диаграммы изменяются автоматически.

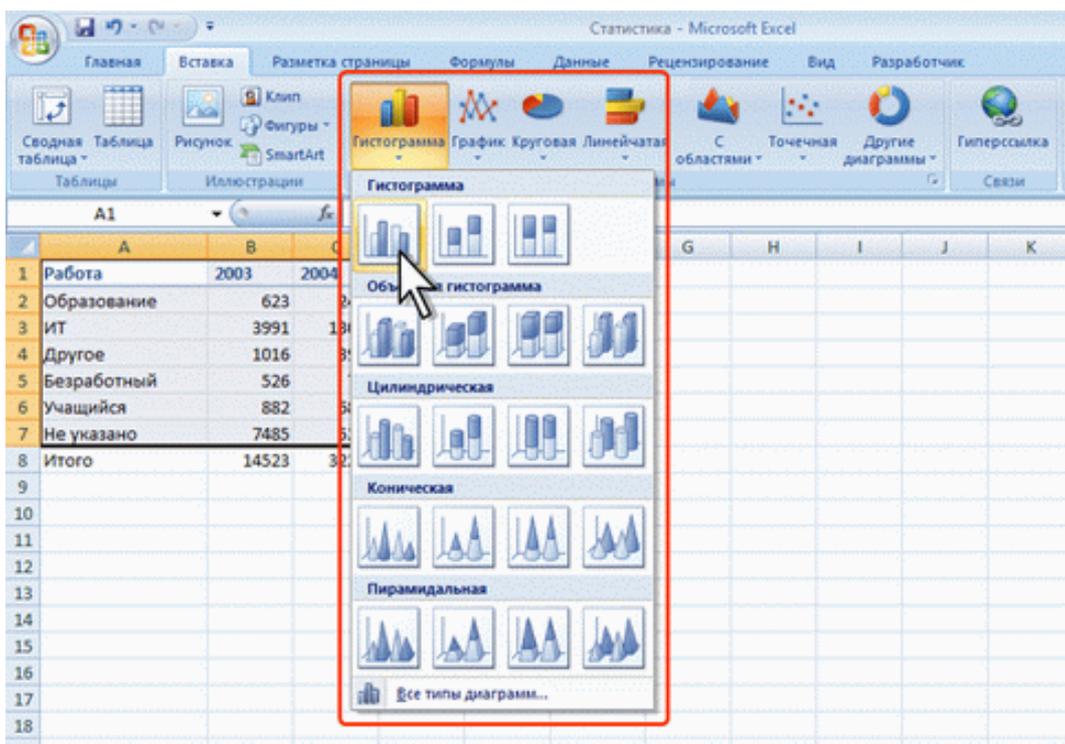
5.5.1. Построение диаграмм

Выделите фрагмент таблицы, для которого создается диаграмма.

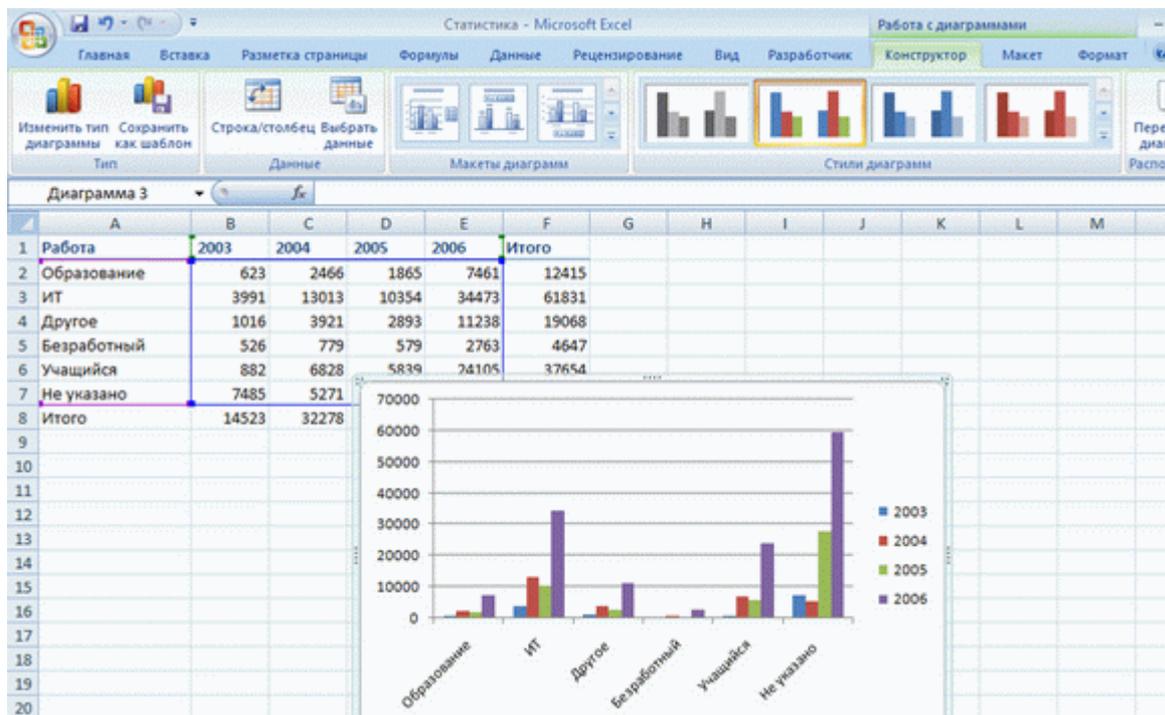


	A	B	C	D	E	F
1	Работа	2003	2004	2005	2006	Итого
2	Образование	623	2466	1865	7461	12415
3	ИТ	3991	13013	10354	34473	61831
4	Другое	1016	3921	2893	11238	19068
5	Безработный	526	779	579	2763	4647
6	Учащийся	882	6828	5839	24105	37654
7	Не указано	7485	5271	27892	59467	100115
8	Итого	14523	32278	49422	139507	235730

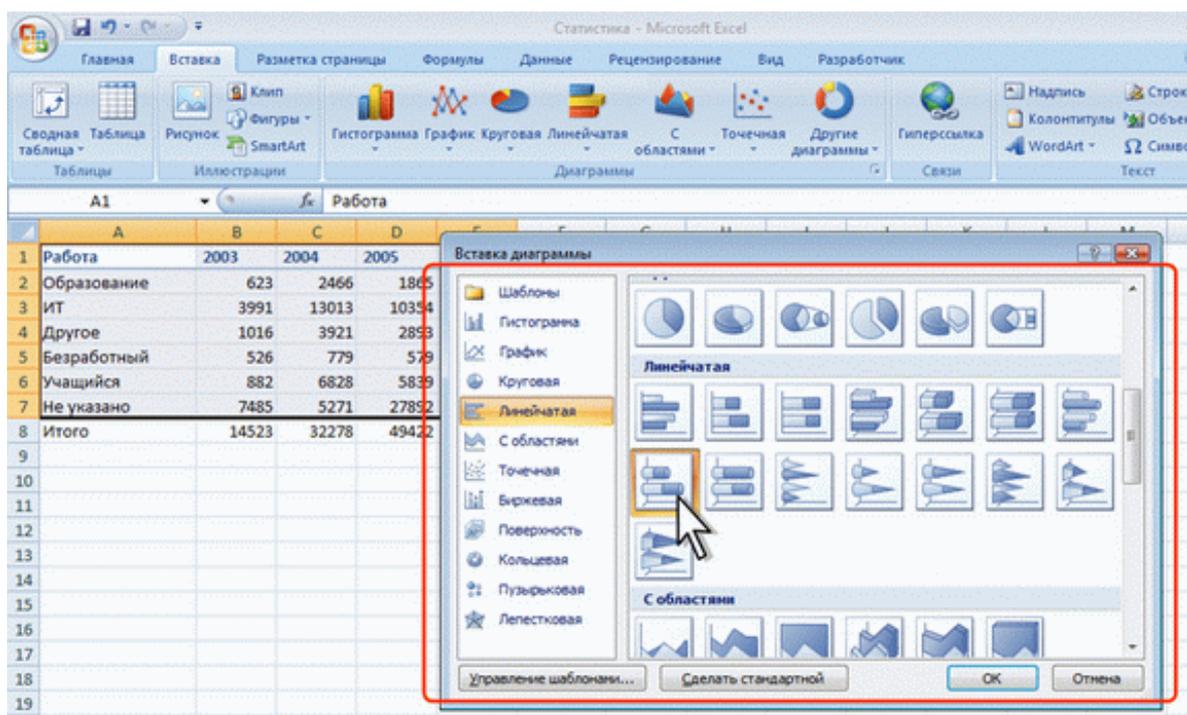
На вкладке *Вставка* в группе *Диаграммы* щелкните по кнопке с нужным типом диаграмм и в галерее выберите конкретный вид диаграммы.



На листе будет создана диаграмма выбранного вида.



Если в группе *Диаграммы* не отображается подходящий тип и вид диаграммы, щелкните по кнопке группы *Диаграммы* и выберите диаграмму в окне *Вставка диаграммы*.



Для создания диаграммы стандартного типа достаточно выделить фрагмент листа и нажать клавишу F11.

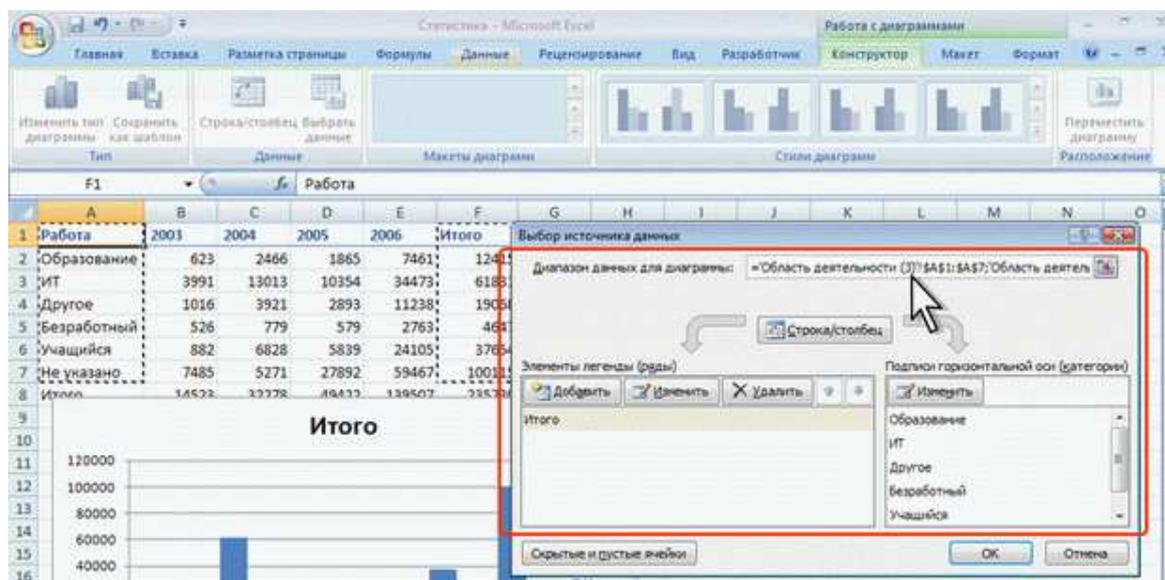
Для удаления диаграммы достаточно выделить ее и нажать клавишу Delete.

5.5.2. Изменение данных диаграммы

Замена источника данных:

1. В группе *Диапазон* вкладки *Работа с диаграммами/Конструктор* нажмите кнопку *Выбрать данные*.

2. В окне *Выбор источника данных* очистите поле *Диапазон данных* для диаграммы, а затем выделите на листе новый диапазон данных.



Изменение диапазона источника данных

В тех случаях, когда диаграмма расположена на листе с данными, изменить диапазон источника данных можно перетаскиванием маркеров диапазонов данных. Маркеры диапазонов отображаются на листе при выделении диаграммы. Как правило, выделяются три диапазона: в зеленой рамке – названия рядов диаграммы (в таблице на ячейки B1 : C1), в сиреневой рамке – названия категорий (в таблице на ячейки A2 : A7), в синей рамке – значения рядов данных (в таблице на ячейки B2 : C7).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Работа	2003	2004	2005	2006	Итого									
2	Образование	623	2466	1865	7461	12415									
3	ИТ	3991	13013	10354	34473	61831									
4	Другое	1016	3921	2893	11238	19068									
5	Безработный	526	779	579	2763	4647									
6	Учащийся	882	6828	5839	24105	37654									
7	Не указано	7485	5271	27892	59467	100115									

Для того чтобы изменить ряды данных, следует перетащить зеленую рамку на другие ячейки, а для добавления или удаления рядов данных следует перетащить зеленый маркер выделения.

Для того чтобы изменить категории, следует перетащить сиреневую рамку на другие ячейки, а для добавления или удаления категорий следует перетащить сиреневый маркер выделения.

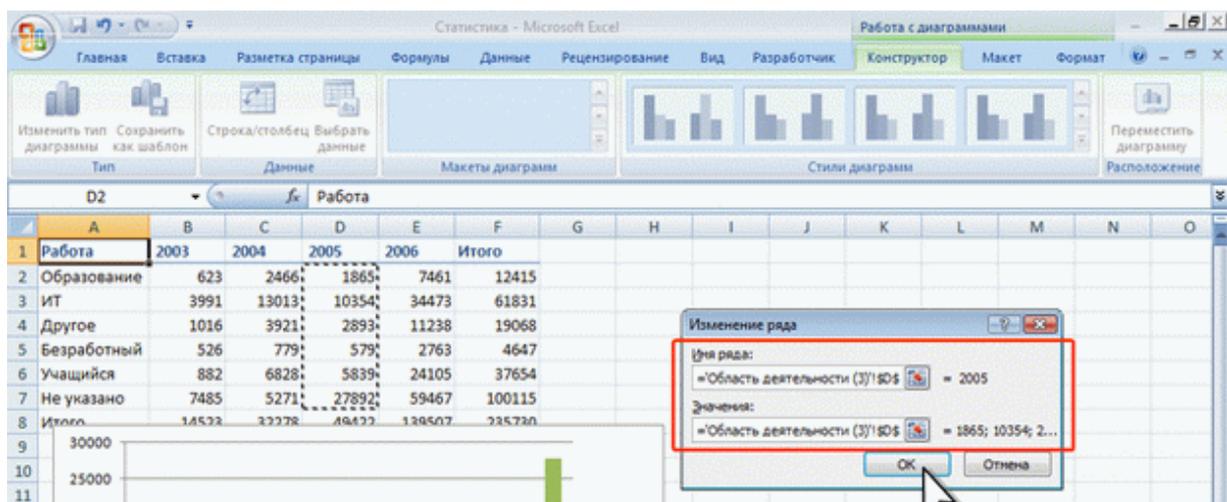
Для того чтобы изменить одновременно категории и ряды данных, следует перетащить синюю рамку на другие ячейки (зеленая и сиреневая рамки при этом переместятся автоматически), а для добавления или удаления одновременно категорий и рядов данных следует перетащить синий маркер выделения.

Для изменения рядов данных, подписей осей, легенды можно использовать окно *Выбор источника данных*.

1. В группе *Диапазон* вкладки *Работа с диаграммами/Конструктор* нажмите кнопку *Выбрать данные*.

2. Для добавления ряда данных в окне *Выбор источника данных* нажмите кнопку *Добавить*.

3. В окне *Изменение ряда* очистите поле *Имя ряда*, а затем выделите на листе ячейку, содержащую название ряда данных; очистите поле *Значение*, а затем на листе выделите ячейки, содержащие значения ряда данных.



4. Для удаления ряда данных в окне *Выбор источника* данных выделите название этого ряда и нажмите кнопку *Удалить*.

5.5.3. Редактирование диаграмм

Готовую диаграмму можно изменить. При щелчке на элементе диаграммы (ряды данных, оси координат, заголовок диаграммы, область построения и т.д.) он выделяется маркерами, а при наведении на него указателя мыши – комментируется *всплывающей подсказкой*.

Изменение типа диаграммы.

Алгоритм:

- 1) щелчком мыши выделить диаграмму;
- 2) вывести на экран вкладку *Работа с диаграммами*, если она не отображается на экране;
- 3) щелкнуть на присоединенной стрелке кнопки *Тип диаграммы* и в появившейся таблице типов выбрать нужный тип диаграммы. При этом тип построенной диаграммы автоматически изменяется на выбранный.

Тип диаграммы можно также изменить с помощью команды *Тип диаграммы* из контекстного меню. При этом открывается диалоговое окно *Тип диаграммы*, в котором можно выбрать иной тип и подтип диаграммы.

Форматирование сетки и легенды. Форматы некоторых диаграмм содержат *координатную сетку* и *легенду*. Сетку и легенду можно добавить к любому типу диаграмм. На диаграмме с сеткой легче оценить порядок величин, а наличие легенды помогает пояснить диаграмму и делает ее легко читаемой. Если же сетка и легенда мешают, то их можно удалить.

Операции с сеткой производятся на вкладке *Линии сетки* диалогового окна *Параметры диаграммы*, которое можно вывести на экран либо с помощью вкладки *Работа с диаграммами* → *Параметры диаграммы*, либо с помощью контекстного меню области диаграммы.

Добавить или удалить легенду, а также управлять ее размещением в поле диаграммы, можно на вкладке *Легенда* диалогового окна *Параметры диаграммы*. Переместить легенду и изменить ее размеры можно также с помощью мыши.

5.5.4. Оформление диаграмм

Если необходимо выделить на диаграмме отдельный элемент данных или придать какой-либо диаграмме особый вид, можно сделать это, внося в диаграмму дополнительные линии, стрелки и выделяя элементы диаграммы различными цветами.

Маркеры отдельных данных можно выделить стрелочкой или линией, а также изменить цвет ряда данных или отдельных их значений. В диаграмму можно также вносить текстовые поля, менять шрифт и другие атрибуты уже имеющихся текстовых фрагментов.

Изменение цвета элементов диаграммы. Если пользователя не устраивает цветовая гамма на диаграмме и нужно представить данные другими цветами, то цвета любых элементов диаграммы (сетки, фона легенды и текста, рамок, линий, маркеров рядов) можно легко изменить с помощью кнопки *Цвет заливки*  из панели инструментов *Форматирование* или соответствующей команды из контекстного меню. Изменять формат объектов можно и при помощи команд меню *Формат*.

Примечание. Имя команды форматирования в контекстном меню зависит от типа выбранного объекта. Для легенды она называется *Формат легенды*, для маркеров ряда данных или сетки эта команда будет называться *Формат рядов данных* или *Формат линий сетки* соответственно, для всей диаграммы – *Формат области диаграммы* и т.д. Выполнение такой команды открывает диалоговое окно форматирования элемента диаграммы, различные вкладки которого позволяют изменять не только цвета, но и другие параметры отображения выбранного элемента данных.

Изменение перспективы объемной диаграммы. Работая с объемными диаграммами, можно изменять большее количество параметров, нежели для плоских диаграмм. Для объемных диаграмм можно задать ориентацию в пространстве, выбрать наилучшую перспективу изображения и подходящий ракурс. Меняя угол поворота диаграммы, можно сделать лучше видимыми элементы диаграммы, находящиеся на заднем плане, или даже переместить их на передний план. Меняя перспективу, возвышение или угол поворота диаграммы, можно достичь наиболее наглядного изображения данных.

Перспективы объемной диаграммы можно изменять, воспользовавшись командой *Объемный вид* из меню *Диаграмма*. В открывшемся окне *Формат трехмерной проекции*, варьируя различные параметры, можно изменять положение в пространстве точки, из которой видна диаграмма. Здесь же есть окно предварительного просмотра, в котором можно увидеть, как будет меняться вид диаграммы в зависимости от устанавливаемых параметров. Размер элементов диаграммы и промежутки между ними можно изменять с помощью контекстного меню, выбрав в нем команду *Формат рядов данных*. В открывшемся окне *Формат ряда данных* нужно выбрать вкладку *Параметры*. Здесь можно изменять глубину и ширину зазора между элементами диаграммы и глубину самой диаграммы. Результат изменений можно увидеть тут же в окне предварительного просмотра.

5.6. Управление данными

5.6.1. Сортировка данных

При работе с Microsoft Excel часто приходится сортировать и обрабатывать списки данных. В списке заголовки столбцов определяют *поля*, а строки содержат *записи*. В каждом поле содержится информация определенного типа, например фамилии, имена и т.д., а запись состоит из описания элемента списка. Совокупность данных в виде таблицы полей и записей называется *базой данных*. В Microsoft Excel понятия список и БД взаимозаменяемы.

Прежде чем вывести на экран или напечатать данные списка в определенном порядке, необходимо про-

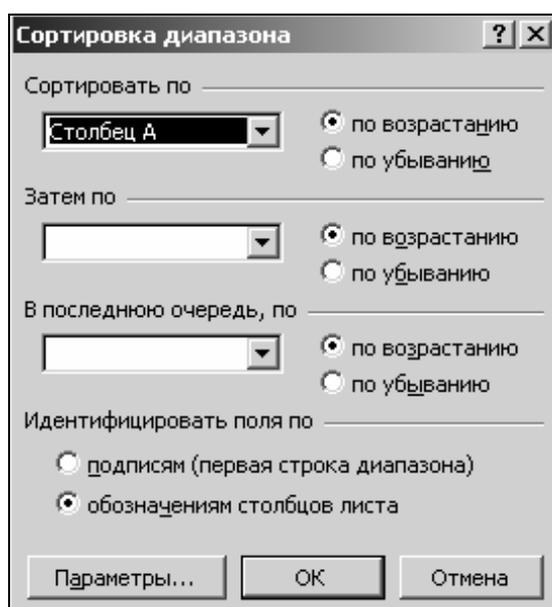


Рис. 5.13. Диалоговое окно *Сортировка диапазона*

произвести сортировку данных. Для каждого столбца списка данных можно задать свой способ сортировки. Имя столбца (поля), по которому проводится сортировка, называется *ключом сортировки*.

Сортировка диапазона данных по возрастанию или убыванию:

1) выделить любую ячейку этого диапазона. Весь диапазон, включающий выделенную ячейку, автоматически подлежит сортировке;

2) меню *Данные*→*Сортировка* – откроется окно диалога *Сортировка диапазона* (рис. 5.13);

3) в поле *Сортировать по* в раскрывающемся списке выбрать ключ сортировки и указать вид сортировки *по возрастанию* (от *А* до *Я* или от *0* до *9*) или *по убыванию* (от *Я* до *А* или от *9* до *0*) с помощью соответствующих переключателей;

4) ОК.

Примечание. Простейшую сортировку по возрастанию или по убыванию можно также выполнять с помощью соответствующих кнопок на панели инструментов *Стандартная* ( и .

Сортировка строк по двум или трем ключам. Иногда приходится использовать более одного ключа сортировки. Лучше всего, если в сортируемом списке будут заголовки столбцов.

Алгоритм:

1) выделить ячейку в списке, который требуется отсортировать;

2) меню *Данные* → *Сортировка* – откроется окно диалога *Сортировка диапазона* (см. рис. 5.13);

3) указать ключи сортировки в полях *Сортировать по* и *Затем по*, если выполняется сортировка по двум критериям. Если нужно выполнить сортировку по трем критериям, то дополнительно указывается ключ сортировки в поле *В последнюю очередь, по*;

4) выбрать остальные параметры сортировки, ОК.

5.6.2. Подбор параметра

Когда нужно определить, насколько следует изменить переменную, чтобы результат формулы, в которую она входит, равнялся заданной величине, используют функцию *Подбор параметра* из меню *Сервис*. Эта функция позволяет исследовать уравнения и формулы, исходя из итогового результата. Иными словами, задается требуемый результат, выбирается изменяемый параметр формулы и запускается программа поиска значения параметра, при котором будет достигнут указанный результат.

Пример 1:

– составить формулу, вычисляющую размер платежей по кредиту фиксированного размера в зависимости от величины процентной ставки.

– определить максимально допустимый размер кредита по заданной величине выплат и при фиксированном проценте.

Алгоритм:

1) Для расчета величины ежемесячных выплат по кредиту используется функция *ПЛТ*. Ее синтаксис: *ПЛТ (ставка; кпер; пс; бс; тип)*, где *ставка* – процентная ставка по ссуде (в примере 8,5%), *кпер* – общее число выплат по ссуде (в примере 360 месяцев), *пс* – приведенная к текущему

моменту стоимость, или общая сумма, которая на текущий момент равноценна ряду будущих платежей, называемая также основной суммой (в примере 120 000 р. Величина в данном случае включается в формулу со знаком «-», т.к. мы исходим из того, что на момент расчета никакие выплаты не осуществлялись). Аргументы *bc* и *тип* для данной функции обязательными не являются (рис. 5.14).

2) Функция *ПЛТ* помещает в ячейку *B5* величину ежемесячных выплат для заданного кредита при процентной ставке 8,5% годовых.

3) Пусть имеется возможность ежемесячно выплачивать по кредиту сумму 900 р. при процентной ставке 8,5%:

– выделить ячейку, для которой задается величина выплат (в примере – ячейка *B5*);

– выполнить команду меню *Сервис*→*Подбор параметра* – откроется окно диалога *Подбор параметра* (рис. 5.15);

B5		fx =ПЛТ(B1/12;B2;-B3)	
	A	B	
1	Проценты	8,5%	
2	Срок кредита (в месяцах)	360	
3	Кредит	120 000р.	
4			
5	Величина ежемесячных выплат	923р.	

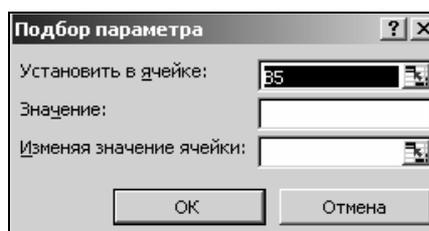


Рис. 5.14. Использование функции ПЛТ

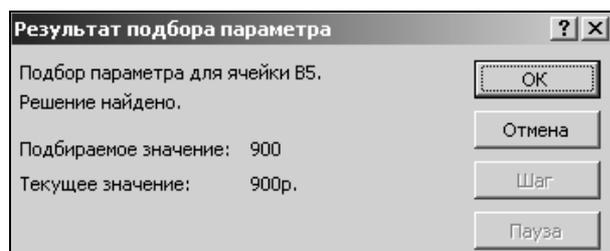
Рис. 5.15. Окно диалога *Подбор параметра*

– в поле *Значение* ввести заданную величину выплат (в примере 900 р.)

– ОК – откроется окно *Результат подбора параметра*, сообщающее, что решение найдено (рис. 5.16);

– в поле *Изменяя значение ячейки* ввести адрес ячейки, для которой выполняется подбор (в примере – ячейка *B3*);

– ОК – новые значения будут введены в соответствующие ячейки (рис. 5.17).



B5		fx =ПЛТ(B1/12;B2;-B3)	
	A	B	
1	Проценты	8,5%	
2	Срок кредита (в месяцах)	360	
3	Кредит	117 048р.	
4			
5	Величина ежемесячных выплат	900р.	

Рис. 5.16. Окно диалога *Результат подбора параметра*

Рис. 5.17. Результат работы подбора параметра

5.6.3. Таблицы подстановки данных

Введя формулы в ячейки рабочего листа, можно провести их анализ и изучить область возможных значений этих формул. *Таблицы подстановки данных* позволяют оперативно вычислять все значения в каждой операции. Эти таблицы представляют собой диапазоны ячеек, показывающие результаты подстановки различных значений в одну или несколько формул.

Например, есть несколько допустимых комбинаций данных, которые нужно сравнить. Пользователю может потребоваться сравнить размеры выплат по кредиту для различных процентных ставок или для различных сроков кредита. Или потребуется оценить влияние роста различных показателей торговли на текущий доход. Вместо того чтобы подбирать параметры и поочередно следить за изменением соответствующих величин, можно составить таблицу данных и сравнить сразу несколько результатов. В таблицах подстановки данных варьируются одна или две переменные, а количество строк таблицы может быть произвольным. Например, можно получить размер платежей по кредиту в зависимости от величины процентной ставки, колеблющейся между 6 и 12%, или найти влияние процента роста торговли (на 2, 3, 4 или 5%) на текущий доход.

Существуют два типа таблиц подстановки данных: *таблицы подстановки с одной переменной* и *таблицы подстановки с двумя переменными*. Названия говорят сами за себя. Таблицы данных с одной переменной позволяют исследовать влияние различных значений одной переменной на результат одной или нескольких формул. В таблицах с двумя переменными анализируется зависимость результата одной формулы от изменения двух входящих в нее переменных.

Создание таблицы подстановки с одной переменной. Предположим, следует сформировать таблицу подстановки с одной переменной, чтобы введенные значения были расположены либо в столбце (ориентированные по столбцу), либо в строке (ориентированные по строке). Формулы, используемые в таблицах подстановки с одной переменной, должны ссылаться на одну и ту же на ячейку ввода.

Примечание. Ячейка ввода – это ячейка, в которую подставляются все значения из таблицы данных. Ячейкой ввода может быть любая ячейка листа. Хотя ячейка ввода не обязана входить в таблицу данных, формулы в таблице данных должны ссылаться на ячейку ввода.

Алгоритм:

1) либо в отдельный столбец, либо в отдельную строку ввести список значений, которые следует подставлять в ячейку ввода;

- 2) выполнить одно из следующих действий:
 - если значения в таблице подстановки ориентированы по столбцу, ввести формулу в ячейку, расположенную на одну строку выше и на одну ячейку правее первого значения. Правее первой формулы можно ввести любые другие формулы;
 - если значения в таблице подстановки ориентированы по строке, ввести формулу в ячейку, расположенную на один столбец левее и на одну строку ниже первого значения. В том же столбце, но ниже можно ввести любые другие формулы.
- 3) выделить диапазон ячеек, содержащий формулы и значения подстановки;
- 4) выполнить команду меню *Данные*→*Таблица подстановки*→окно диалога *Таблица подстановки* (рис. 5.18);
- 5) выполнить одно из следующих действий:
 - б) если значения в таблице расположены по столбцам, ввести ссылку на ячейку ввода в поле *Подставлять значения по строкам в*;
 - 7) если значения в таблице расположены по строкам, ввести ссылку на ячейку ввода в поле *Подставлять значения по столбцам в*;
- 8) ОК – в столбец, находящийся справа от столбца подставляемых значений, или в строку, находящуюся ниже строки подставляемых значений, будут помещены результаты заданной формулы для различных аргументов.

Создание таблицы подстановки с двумя переменными. С помощью таблиц подстановки данных с двумя переменными можно исследовать влияние одновременно двух переменных на результат формулы. Формула должна ссылаться на две различные ячейки ввода.

Алгоритм:

- 1) в ячейку листа ввести формулу, которая ссылается на две ячейки ввода;
- 2) в том же столбце ниже формулы ввести значения подстановки для первой переменной;
- 3) ввести значения подстановки для второй переменной правее формулы в той же строке;
- 4) выделить диапазон ячеек, содержащий формулу и оба набора данных подстановки;
- 5) выполнить команду меню *Данные*→*Таблица подстановки*→окно диалога *Таблица подстановки* (см. рис. 5.17);

- 6) в поле *Подставлять значения по столбцам в* ввести ссылку на ячейку ввода для значений подстановки в строке;
- 7) в поле *Подставлять значения по строкам в* введите ссылку на ячейку ввода для значений подстановки в столбце;
- 8) ОК – результаты формул будут занесены в таблицу подстановки.

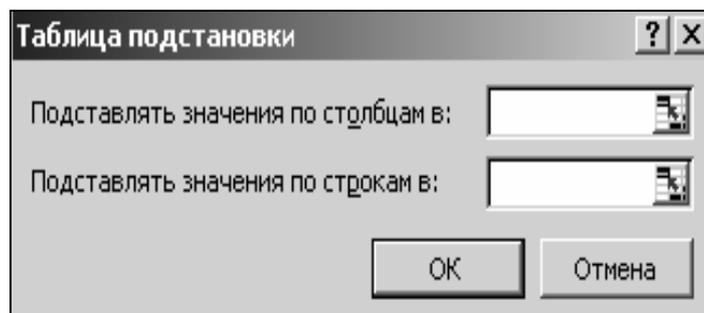


Рис. 5.17. Окно диалога *Таблица подстановки*

6. МЕТОДЫ И АЛГОРИТМЫ ЧИСЛЕННОГО РЕШЕНИЯ УРАВНЕНИЙ

6.1. Основы теории погрешностей

На общую погрешность решения задачи влияет целый ряд факторов. Рассмотрим основные из них.

Пусть R – точное значение результата решения некоторой задачи. Из-за того, что построенная математическая модель нетождественна реальному объекту, а также по причине неточности исходных данных, вместо R будет получен результат, который обозначим R_1 . Образовавшаяся таким образом погрешность $\varepsilon_1 = R - R_1$ уже не может быть устранена в ходе последующих вычислений (так называемая неустранимая погрешность). Приступив к решению задачи в рамках математической модели, выбирают приближенный метод (например, численный) и еще не приступив к вычислениям, допускают новую погрешность, приводящую к получению результата R_2 . Погрешность $\varepsilon_2 = R_1 - R_2$ называется погрешностью метода. Реализовав метод получают R_3 . Погрешность $\varepsilon_3 = R_2 - R_3$ называется вычислительной погрешностью. В случае использования ЭВМ происходит принудительное округление, которое связано с конечностью разрядной сетки машины. Полная погрешность получается как сумма всех погрешностей

$$\varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_3.$$

При решении конкретных задач те или иные виды погрешностей могут либо отсутствовать совсем, либо влиять на окончательный результат незначительно. Тем не менее, для исчерпывающего представления о точности результата в каждом случае необходим полный анализ погрешностей всех видов.

6.2. Приближенное решение нелинейных уравнений

6.2.1. Отделение корней уравнения

При разработке алгоритмов решения конкретных инженерных задач часто приходится прибегать к методам численного анализа. Рассмотрим наиболее простые численные методы решения некоторых классов задач, которые используются в данной работе. Для рассматриваемых методов приводятся реализующие их подпрограммы с необходимыми инструкциями. Использование некоторых методов иллюстрируется примерами.

Корнем уравнения $f(x) = 0$ называется такое значение $x = \xi$ аргумента функции $f(x)$ при котором это уравнение обращается в тождество: $f(\xi) = 0$.

Корень уравнения геометрически представляет собой абсциссу точки пересечения, касания (рис. 6.1) или другой общей точки (рис. 6.2) графика функции $y = f(x)$ и оси OX .

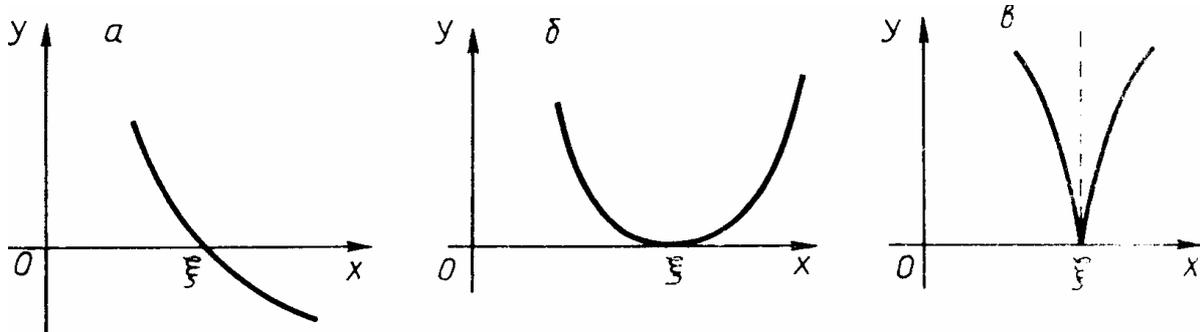


Рис. 6.1. Графики функции $y = f(x)$

Отделить корень уравнения – значит найти такой конечный промежуток, внутри которого имеется единственный корень данного уравнения. Отделение корней уравнения можно выполнить аналитически, исследовав данную функцию или графически, построив график функции $y = f(x)$, по которому можно судить о том, в каких промежутках находятся точки пересечения его с осью OX .

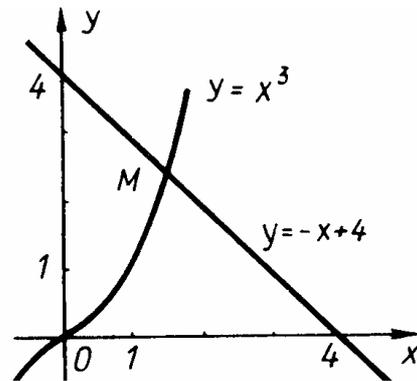


Рис. 6.2. Пример отделения корня методом разбиения функции на 2

6.2.2. Уточнение корня методом половинного деления

Допустим, что искомый корень уравнения отделен, т.е. найден отрезок $[a, b]$, на котором имеется один и только один корень уравнения. Любую точку этого отрезка можно принять за приближенное значение корня. Погрешность такого приближения не превосходит длины $[a, b]$. Следовательно, задача отыскания приближенного значения корня с заданной точностью ϵ сводится к нахождению отрезка $[a, b]$, содержащего только один корень уравнения. Эту задачу обычно называют задачей уточнения корня.

Задача уточнения корня сводится к делению отрезка пополам, с каждым шагом выделяя новый кусок отрезка $[a, c]$ или $[c, b]$ до тех пор, пока значение не будет $x \approx 0$. Примерная схема алгоритма решения уравнения $f(x) = 0$ методом половинного деления представлена на рисунке 6.3.

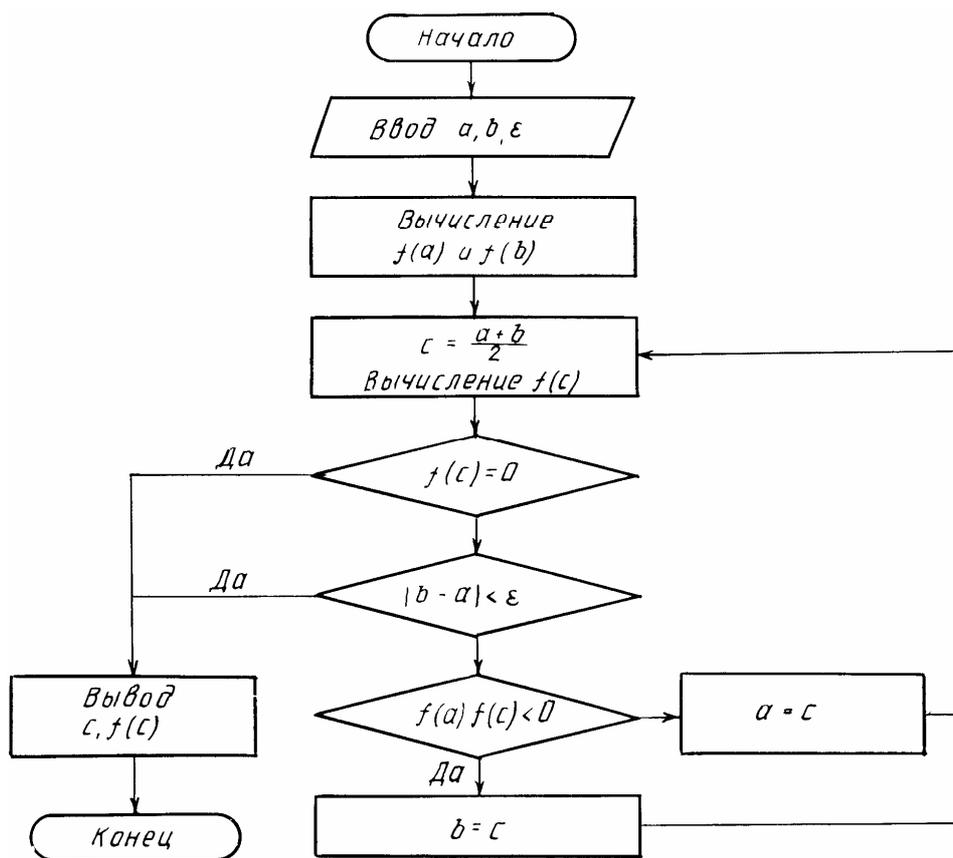


Рис. 6.3. Блок-схема алгоритма решения нелинейного уравнения методом половинного деления

6.2.3. Метод хорд (секущих)

Метод хорд, или метод секущих, приближенного решения уравнения $y = f(x)$ имеет следующую геометрическую иллюстрацию: вместо точки пересечения оси OX и графика функции $f(x)$, входящей в это уравнение, рассматривается точка пересечения данной оси и отрезка прямой, соединяющей концы дуги графика (рис. 6.4). Если известно $(n - 1)$ -е приближение, то n -е вычисляется по формуле

$$x_n = \frac{bf(x_{n-1}) - x_{n-1}f(b)}{f(x_{n-1}) - f(b)} \quad \text{при } f(b)f''(x) > 0$$

где $n = 1, 2, 3, \dots$,

$$\text{или } x_n = \frac{af(x_{n-1}) - x_{n-1}f(a)}{f(x_{n-1}) - f(a)} \quad \text{при } f(a)f''(x) > 0.$$

Оценка абсолютной погрешности определяется формулой

$$|\xi - x_n| \leq \frac{|f(x_n)|}{\mu}, \quad \mu = \max |f'(x)|, \quad f'(x) \neq 0.$$

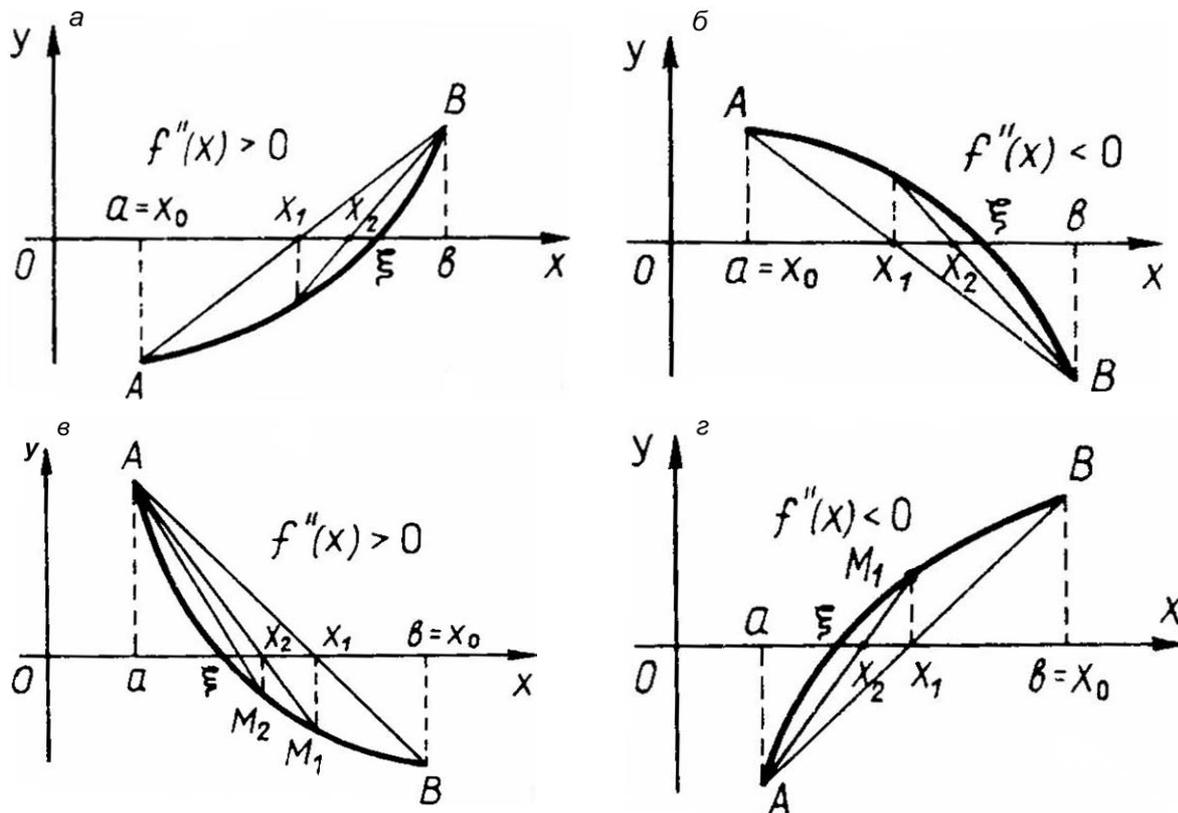


Рис. 6.4. Графики, иллюстрирующие метод хорд

6.2.4. Метод касательных

Метод касательных (или метод Ньютона) отличается от метода хорд тем, что здесь рассматривается не секущая, соединяющая концы дуги графика, а касательная к графику. Точка пересечения касательной с осью OX дает приближенное значение корня (рис. 6.5).

В методе касательных $(n + 1)$ -е приближение вычисляется по формуле

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad (n = 1, 2, 3, \dots),$$

в которой за нулевое приближение x_0 принимается такое значение из отрезка $[a, b]$, для которого выполняется условие при $f(x_0)f''(x) > 0$.

Оценка погрешности, как и в методе хорд, определяется формулой

$$|\xi - x_n| \leq \frac{|f(x_n)|}{\mu}, \quad \text{где } \mu = \max |f'(x)|, \quad f'(x) \neq 0.$$

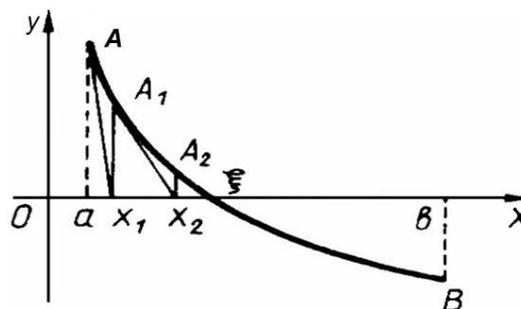


Рис. 6.5. График, иллюстрирующий метод касательных

6.2.5. Метод итераций

Если каким-либо способом получено приближенное значение x_0 корня уравнения $f(x) = 0$, то уточнение корня можно осуществить методом последовательных приближений или методом итераций (рис. 6.6). Для этого уравнение $f(x) = 0$ представляют в виде

$$x = \varphi(x),$$

что всегда можно сделать и притом многими способами, например

$$x = x + cf(x),$$

где c – произвольная постоянная,

$$x_n = \varphi(x_{n-1}).$$

Процесс последовательного вычисления чисел x_n ($n = 1, 2, 3, \dots$) по этой формуле называется методом последовательных приближений или методом итераций. Процесс итераций сходится, если выполнено условие $|\varphi'(x)| \leq q < 1$ на отрезке $[a, b]$, содержащем корень.

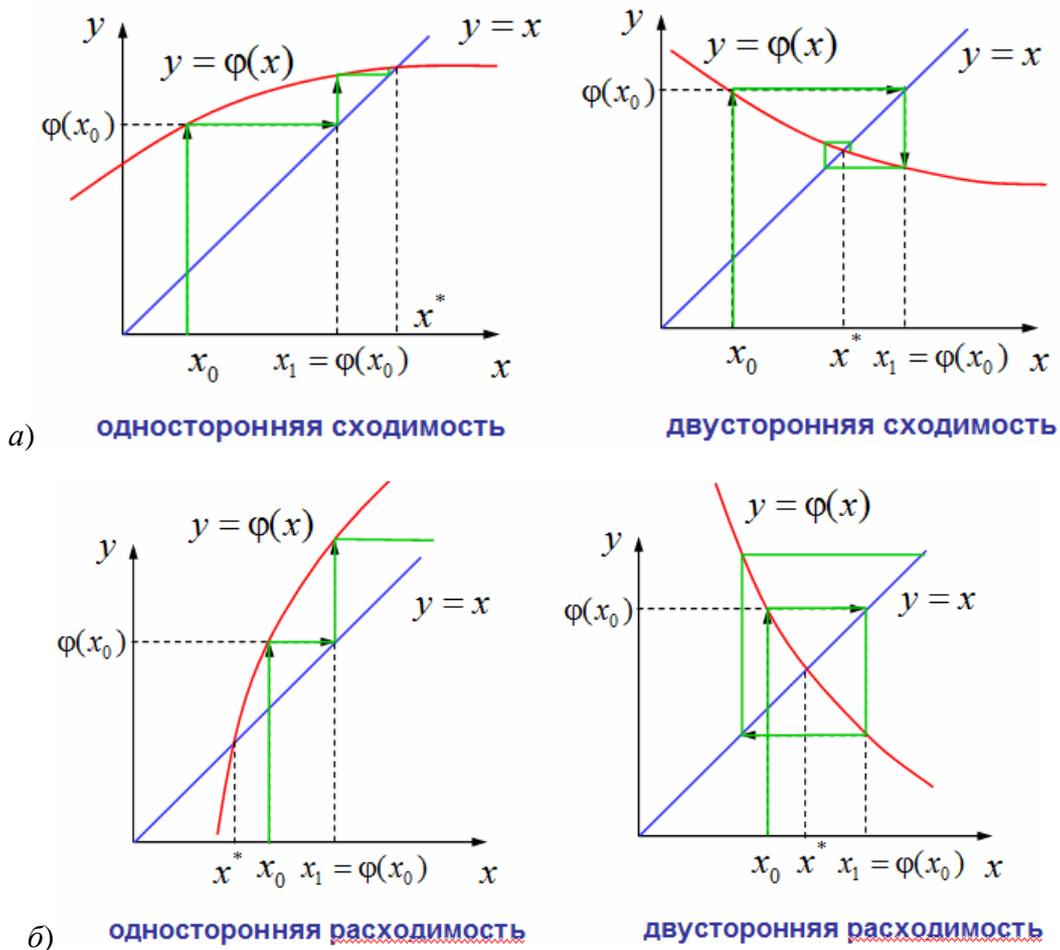


Рис. 6.6. Графики, иллюстрирующие условия итерационных процессов:
а – сходимости; б – расходимости

6.3. Методы решения систем линейных уравнений

Инженеру часто приходится решать алгебраические и трансцендентные уравнения, что может представлять собой самостоятельную задачу или являться частью более сложных задач. В обоих случаях практическая ценность метода в значительной мере определяется быстротой и эффективностью полученного решения. Выбор подходящего метода для решения уравнений зависит от характера рассматриваемой задачи. Задачи, сводящиеся к решению алгебраических и трансцендентных уравнений, можно классифицировать по числу уравнений (рис. 6.7) и в зависимости от предлагаемого характера и числа решений.

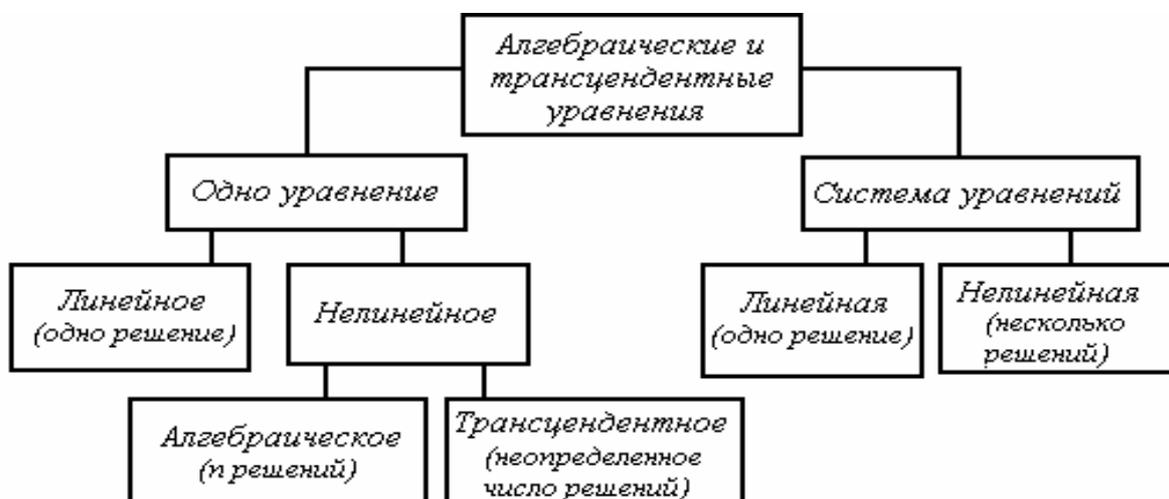


Рис. 6.7. Классификация уравнений

Одно уравнение будем называть *линейным*, *алгебраическим* или *трансцендентным* в зависимости от того, имеет ли оно одно решение, n решений или неопределенное число решений. Систему уравнений будем называть *линейной* или *нелинейной* в зависимости от математической природы входящих в нее уравнений.

Способы решения систем линейных уравнений делятся на две группы:

1) *точные методы*, представляющие собой конечные алгоритмы для вычисления корней системы (решение систем с помощью обратной матрицы, правило Крамера, метод Гаусса и др.),

2) *итерационные методы*, позволяющие получить решение системы с заданной точностью путем сходящихся итерационных процессов (метод итерации, метод Зейделя и др.).

6.4. Приближенное вычисление определенных интегралов

6.4.1. Формулы прямоугольников

Формула левых прямоугольников (рис. 6.8, а)

$$\int_a^b f(x)dx \approx h \sum_{k=0}^{n-1} y_k = h(y_0 + y_1 + y_2 + \dots + y_{n-1});$$

формула правых прямоугольников (рис. 6.8, б)

$$\int_a^b f(x)dx \approx h \sum_{k=1}^n y_k = h(y_1 + y_2 + \dots + y_n),$$

где $h = (b - a)/n$, $y_k = f(x_k)$, $x_k = a + kh$, ($k = 0, 1, 2, \dots, n$).

Абсолютная погрешность метода прямоугольников определяется неравенством

$$|R_n(f)| \leq \frac{(b-a)^2 M}{2n}, \quad \text{где } M = \max |f'(x)|.$$

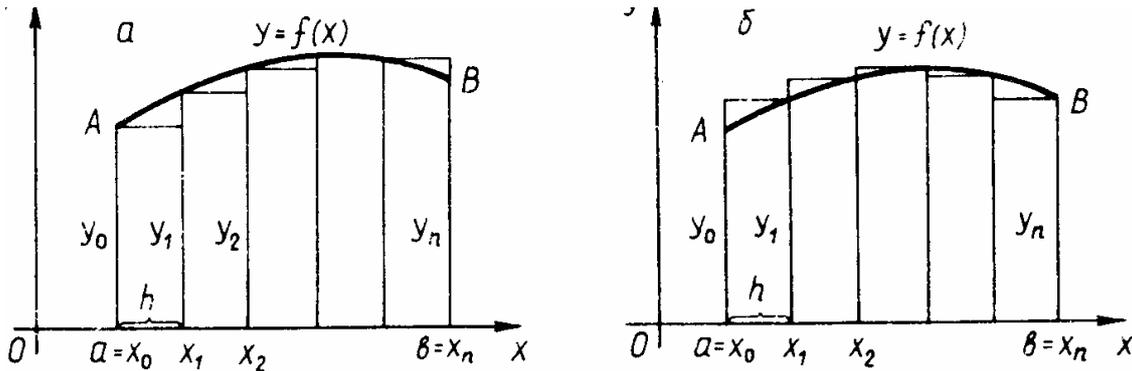


Рис. 6.8. Метод прямоугольников

6.4.2. Формула трапеций

$$\int_a^b f(x)dx \approx h(y_0/2 + y_1 + y_2 + \dots + y_n/2),$$

где $h = (b - a)/n$, $y_k = f(x_k)$, $x_k = a + kh$, ($k = 0, 1, 2, \dots, n$).

Правая часть этой формулы выражает площадь фигуры, состоящей из трапеций, высота каждой из которых равна h (рис. 6.9). Остаточный член приближенной формулы трапеций

$$|R_n| \leq \frac{(b-a)^3 M}{12n}, \quad \text{где } M = \max |f''(x)|.$$

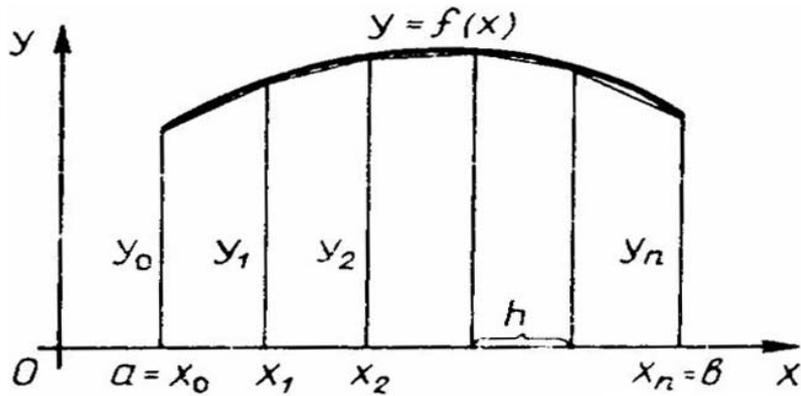


Рис. 6.9. Метод трапеций

6.4.3. Формула парабол

Формула парабол (или формула Симпсона) имеет вид

$$\int_a^b f(x) dx \approx \frac{h}{3} (y_0 + 4(y_1 + y_3 + \dots + y_{2n-1}) + 2(y_2 + y_4 + \dots + y_{2n-2}) + y_{2n}),$$

где $h = (b-a)/(2n)$, $y_k = f(x_k)$, $x_k = a + kh$, $(k = 0, 1, 2, \dots, 2n)$.

Правая часть формулы парабол выражает площадь фигуры, составленной из параболических трапеций $x_0M_0M_2x_2$, $x_2M_2M_4x_4$ (рис. 6.10).

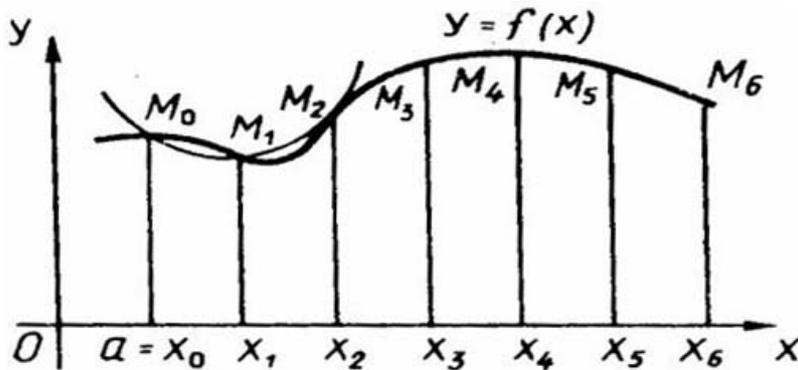


Рис. 6.10. Метод парабол

Дуга $M_0M_1M_2$ графика подынтегральной функции здесь заменена дугой параболы, проходящей через точки M_0 M_1 M_2 . Аналогичная замена произведена и остальных дуг. Для остаточного члена формулы парабол выполняется неравенство

$$|R_n| \leq \frac{(b-a)^5 M}{180(2n)^4}, \quad \text{где } M = \max |f^{(IV)}(x)|.$$

6.5. Приближенное решение дифференциальных уравнений

6.5.1. Интегрирование дифференциальных уравнений с помощью рядов

Решения многих дифференциальных уравнений не выражаются в элементарных функциях. В этих случаях пользуются приближенными методами интегрирования дифференциальных уравнений. Одним из таких методов является представление решения уравнения в виде степенного ряда; сумма конечного числа членов этого ряда будет приближенно равна искомому решению. Указанный степенной ряд находят способом неопределенных коэффициентов или способом, основанным на применении ряда Тейлора.

Способ неопределенных коэффициентов особенно удобен в применении к линейным уравнениям и состоит в следующем. Если все коэффициенты уравнения и свободный член разлагаются в ряды по степеням, сходящиеся в интервале $(a - h, a + h)$, то искомое решение $y = y(x)$ также представляется степенным рядом сходящимся в том же интервале

$$y(x) = C_0 + C_1(x - a) + C_2(x - a)^2 + \dots + C_n(x - a)^n.$$

Подставляя в уравнение функцию $y(x)$ и ее производные, приравнивают коэффициенты при одинаковых степенях. Из полученных при этом уравнений и заданных начальных условий находят коэффициенты. Способ, основанный на применении ряда Тейлора – Маклорена, заключается в последовательном дифференцировании данного уравнения

$$y(x) = y(a) + y'(a)(x - a) + \frac{y''(a)}{2!}(x - a)^2 + \dots + \frac{y^{(n)}(a)}{n!}(x - a)^n.$$

Это дает возможность найти значения производных, входящих в выражения для коэффициентов ряда, являющегося решением уравнения.

6.5.2. Метод Эйлера

Пусть требуется решить задачу Коши: найти решение дифференциального уравнения

$$y' = f(x, y),$$

удовлетворяющее начальному условию $y(x_0) = y_0$.

При численном решении этого уравнения задача ставится так: в точках $x_0, x_1, x_2, \dots, x_n$ найти приближения y_k ($k = 0, 1, 2, \dots, n$) для значений точного решения $y(x_k)$. Разность $\Delta x_k = x_{k-1} - x_k$ называется шагом сетки. Во многих случаях величину Δx_k принимают постоянной h , тогда

$$x_k = x_{k-1} + kh \quad (k = 0, 1, 2, \dots, n).$$

Метод Эйлера основан на непосредственной замене производной разностным отношением по приближенной формуле

$$\Delta y / \Delta x = f(x, y),$$

где $\Delta y = y(x+h) - y(x)$, $\Delta x = (x+h) - x = h$.

Приближенное значение y_k в точке $x_k = x_0 + kh$ вычисляется по формуле

$$y_{k+1} = y_k + hf(x_k, y_k) \quad (k = 0, 1, 2, \dots, n).$$

6.5.3. Метод Рунге – Кутты

Схема четвертого порядка точности:

$$y_{n+1} = y_n + (k_1 + 2k_2 + 2k_3 + k_4) / 6,$$

$$k_1 = h_n \varphi(x_n, y_n), \quad k_2 = h_n \varphi(x_n + h_n / 2, y_n + k_1 / 2),$$

$$k_3 = h_n \varphi(x_n + h_n / 2, y_n + k_2 / 2), \quad k_4 = h_n \varphi(x_n + h_n, y_n + k_3).$$

В общем виде методы Рунге – Кутты можно описать следующим разностным уравнением:

$$\frac{y_{n+1} - y_n}{h} = \sum_{i=1}^m \sigma_i k_i,$$

где $k_i = h \varphi \left(x_n + a_i \cdot h, y_n + \sum_{j=1}^{i-1} b_{ij} k_j \right)$, $i = 2, 3, \dots, m$, $k_1 = h \varphi(x_n, y_n)$.

Погрешность метода Рунге – Кутты $R_S \leq \frac{x_l - x_0}{2880} \max(h_n^4) \max |\varphi^{(IV)}|$.

6.6. Интерполирование функций

В задачах математического моделирования очень часто возникает необходимость заменить используемую в расчетах функциональную зависимость $y(x)$ приближенной функцией $\varphi(x, \bar{a})$, по которой легко вычисляются значения исходной функции и которая в определенном смысле близка к $y(x)$. После такой замены все расчеты выполняют, используя зависимость $\varphi(x, \bar{a})$, причем близость $y(x)$ и $\varphi(x, \bar{a})$ обеспечивается подбором свободных параметров

$$\bar{a} = \{a_0, a_1, \dots, a_n\}. \quad (6.2)$$

Интерполяция может быть линейной или нелинейной, в соответствии с характером зависимости функции $\varphi(x, \bar{a})$ от параметров \bar{a} .

6.6.1. Линейная интерполяция

При линейной интерполяции функция $\varphi(x, \bar{a})$ имеет вид

$$\varphi(x, \bar{a}) = \sum_{j=0}^n a_j \varphi_j(x), \quad (6.3)$$

где все функции $\varphi_j(x)$ линейно независимы.

Подставляя (6.3) в (6.2), получим систему линейных уравнений для определения a_j :

$$\sum_{j=0}^n a_j \varphi_j(x_i) = y_i, \quad 0 \leq i \leq n. \quad (6.4)$$

Единственность решения обеспечивается требованием неравенства нулю определителя системы (6.4):

$$\Delta = \begin{pmatrix} \varphi_0(x_0) & \varphi_1(x_0) \dots & \varphi_n(x_0) \\ \cdot & \dots & \cdot \\ \varphi_0(x_n) & \varphi_1(x_n) \dots & \varphi_n(x_n) \end{pmatrix} \neq 0 \quad (6.5)$$

при $x_k \neq x_l$ (т.е. при любых несовпадающих узлах).

Для практических вычислений удобно использовать многочлен $P_n(x)$ в форме интерполяционного полинома Лагранжа или Ньютона.

6.6.2. Интерполяционный многочлен Лагранжа

Интерполяционным многочленом Лагранжа называется многочлен вида

$$\begin{aligned} y = P_n(x) &= \sum_{k=0}^n y_k \frac{(x-x_0)(x-x_1)\dots(x-x_{k-1})(x-x_{k+1})\dots(x-x_n)}{(x_k-x_0)(x_k-x_1)\dots(x_k-x_{k-1})(x_k-x_{k+1})\dots(x_k-x_n)} = \\ &= \sum_{k=0}^n y_k \prod_{\substack{j=0 \\ j \neq k}}^n \frac{x-x_j}{x_k-x_j}. \end{aligned}$$

Этот многочлен удовлетворяет условиям

$$P_n(x) = y_k, \quad k = 0, 1, \dots, n, \quad (6.6)$$

где x_k – узлы (или полюсы) интерполяции; y_k – заданные числа.

Производя интерполирование функции $f(x)$ по формуле Лагранжа, заменяют эту функцию полиномом $P_n(x)$, совпадающим с ней в $(n+1)$ данных точках отрезка $[a, b]$. В остальных точках этого отрезка разность

$R_n(x) = f(x) - P_n(x)$ отлична от нуля и представляет собой погрешность метода. Термин «интерполяция» впервые употребил Д. Валлис (1656 г.) при составлении астрономических и математических таблиц.

6.6.3. Интерполяционный многочлен Ньютона

$$P_n(x) = P_n(x_0) + (x - x_0)P(x_0, x_1) + (x - x_0)(x - x_1)P(x_0, x_1, x_2) + \dots \\ + (x - x_0)(x - x_1)\dots(x - x_{n-1})P(x_0, x_1, x_2, \dots, x_n).$$

Таким образом, выражают многочлен n -й степени через его значения в $(n + 1)$ узлах $x_0, x_1, x_2, \dots, x_n$. Ввиду того, что значения интерполяционного полинома в этих узлах совпадают со значениями интерполируемой функции, разделенные разности выражаются через узловые значения функции. В результате получается полином, называемый интерполяционным многочленом Ньютона:

$$y(x) \approx y(x_0) + \sum_{k=1}^n (x - x_0)(x - x_1)\dots(x - x_{k-1})y(x_0, \dots, x_k).$$

При вычислениях по этой формуле точность расчетов удобно оценивать, наблюдая за тем, насколько быстро убывают члены ряда. Если это происходит достаточно быстро, можно оставлять только те члены, которые больше заданной погрешности расчетов. Безразличен порядок нумерации узлов, что очень удобно при подключении новых узлов для построения полинома более высокого порядка.

Погрешность многочлена Ньютона оценивают по формуле

$$(y(x) - P_n(x)) < \sqrt{\frac{2n}{\pi}} M_{n+1} \left(\frac{h}{2}\right)^{n+1},$$

где $M_{n+1} = \max(y^{(n+1)}(\xi))$ – максимальное значение производной интерполируемой функции на отрезке между наименьшим и наибольшим из значений $x_0, x_1, x_2, \dots, x_n$, а $h = x_{i+1} - x_i$ (предполагается, что шаг постоянен).

Эта формула свидетельствует, что с уменьшением расстояния между узлами (шага) h погрешность представления функций полиномом Ньютона убывает.

Трудность использования формулы на практике состоит в том, что производные интерполируемой функции обычно неизвестны, поэтому для определения погрешности удобнее воспользоваться оценкой первого отброшенного члена.

6.6.4. Нелинейная интерполяция

Для табулирования быстроменяющихся функций требуется весьма малый шаг, т.е. возникает необходимость создавать таблицы очень больших объемов, что в ряде случаев неприемлемо. Оказывается, что преобразованием переменных $\eta = \eta(y)$ и $\xi = \xi(x)$ можно добиться того, чтобы в новых переменных график $\eta(\xi)$ был близок к прямой хотя бы на отдельных участках. В этом случае интерполяцию проводят в переменных (η, ξ) , а затем обратным интерполированием находят $y_i = y(\eta_i)$.

Преобразования $\eta(y)$ и $\xi(x)$ должны быть достаточно простыми (логарифмическая, экспоненциальная, тригонометрические и некоторые другие функции). При этом надо заботиться о том, чтобы и обратное преобразование $y(\eta)$ оказалось несложным.

Во многих задачах теплофизики, гидродинамики, оптики (особенно в задачах переноса излучения) и других областей науки и техники часто встречается степенная зависимость функции от своих аргументов. В этом случае удобны преобразования типа логарифмирования.

Пример 1. Получить формулу для нелинейной двухточечной интерполяции функции $y(x)$, если переменные можно преобразовать по формулам $\eta = \ln(y)$ и $\xi = x$.

Составим интерполяционный полином Ньютона на двухточечном шаблоне:

$$\eta = \eta_0 + \frac{\eta_1 - \eta_0}{\xi_1 - \xi_0} (\xi - \xi_0).$$

В исходных переменных имеем

$$\ln(y) = \ln(y_0) + \frac{\ln(y_1) - \ln(y_0)}{x_1 - x_0} (x - x_0),$$

и окончательно $y = y_0 (y_1 / y_0)^{(x-x_0)/(x_1-x_0)}$.

6.6.5. Интерполяция сплайнами

Слово «сплайн» переводится как «гибкая линейка». Такую «линейку» можно использовать для проведения кривых через заданную совокупность точек, изгибая и придерживая ее так, чтобы ребро проходило через все точки на плоскости. Равновесие гибкой линейки описывается уравне-

нием $\psi''(x) = 0$, т.е. интерполяционный полином на участке между каждой парой соседних точек имеет третью степень:

$$\psi(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3,$$

$$x_{i-1} \leq x \leq x_i, \quad 0 \leq i \leq N.$$

В узлах значения многочлена и интерполируемой функции совпадают:

$$\psi(x_{i-1}) = y_{i-1} = a_i,$$

$$\psi(x_i) = y_i = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3,$$

$$h_i = x_i - x_{i-1}, \quad 1 \leq i \leq N.$$

Число таких уравнений меньше числа неизвестных в два раза. Недостающие уравнения получают, приравнявая во внутренних узлах первые и вторые производные, вычисляемые по коэффициентам на соседних участках:

$$\psi'(x) = b_i + 2c_i(x - x_{i-1}) + 3d_i(x - x_{i-1})^2,$$

$$\psi''(x) = 2c_i + 6d_i(x - x_{i-1}), \quad x_{i-1} \leq x \leq x_i,$$

$$b_{i+1} = b_i + 2c_i h_i + 3d_i h_i^2,$$

$$c_{i+1} = c_i + 3d_i h_i, \quad 1 \leq i \leq N - 1.$$

Недостающие условия можно получить, полагая, например, что вторая производная равна нулю на концах участка интерполирования:

$$\psi''(x_0) = 0, \quad c_1 = 0,$$

$$\psi''(x_N) = 0, \quad c_N + 3d_N h_N = 0,$$

Уравнения позволяют определить все $4N$ неизвестных коэффициента: $a_i, b_i, c_i, d_i (1 \leq i \leq N)$.

6.6.6. Многомерная интерполяция

В различных приложениях широко используют двумерные и трехмерные таблицы. Например, теплофизические свойства различных веществ зависят от температуры и давления, а оптические характеристики – еще и от длины волны излучения.

При многомерной интерполяции из-за громоздкости таблиц необходимо брать достаточно большие шаги по аргументам, т.е. сетка узлов, на которой строят таблицу, получается довольно грубой. Поэтому требу-

ется вводить преобразование переменных $\eta = \eta(y)$, $\xi = \xi(x)$, $\varphi = \varphi(z)$, подбирая подходящие формулы. При удачном выборе таких формул можно использовать в новых переменных интерполяционный полином невысокой степени.

Осуществляя многомерную интерполяцию, следует помнить, что расположение узлов не может быть произвольным. Например, при интерполяции полиномом первой степени $P_1(x, y)$ узлы не должны лежать на одной прямой в плоскости. Действительно, определитель системы трех уравнений

$$z_i = a + bx_i + cy_i, \quad 1 \leq i \leq 3$$

записывается в виде

$$\Delta = x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2).$$

Условие размещения трех точек на одной прямой выглядит следующим образом:

$$\frac{x_2 - x_1}{y_2 - y_1} = \frac{x_3 - x_1}{y_3 - y_1}.$$

После простых преобразований имеем

$$x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2) = 0,$$

т.е. если узлы лежат на одной прямой, то определитель Δ обращается в нуль и построить полином $P_1(x, y)$ вида z_i невозможно.

Проверять условия подобного типа достаточно сложно, поэтому на практике стремятся строить регулярные сетки, как правило, прямоугольные и равномерные, когда узлы являются точками пересечения двух взаимно перпендикулярных систем параллельных прямых. На этой сетке проводят простую последовательную интерполяцию: сначала по строкам, а затем по столбцам.

При последовательной интерполяции завышается степень интерполяционного полинома. При треугольной конфигурации расположения узлов степень многочлена будет минимальной. Многочлен n -й степени в форме Ньютона в этом случае можно представить как обобщение одномерного варианта записи:

$$P_n(x, y) = \sum_{i=0}^n \sum_{j=0}^{n-1} z(x_0, \dots, x_i, y_0, \dots, y_j) \prod_{p=0}^{i-1} (x - x_p) \prod_{q=0}^{j-1} (y - y_q).$$

6.7. Регрессия

Термин «регрессия» ввел в употребление Ф. Гальтон – один из создателей математической статистики. Сопоставляя рост детей и их родителей, он обнаружил, что соответствие между ростом отцов и детей выражено слабо. Оно оказалось меньшим, чем он ожидал. Гальтон объяснил это явление наследственностью не только от родителей, но и от более отдаленных предков. По его предположениям, рост определяется наполовину родителями, на четверть – дедушкой и бабушкой, на одну восьмую – прадедом и прабабкой и т.д. Неважно, насколько был прав Гальтон. Главное в том, что он обратил внимание на движение назад по генеалогическому дереву и назвал его регрессией, позаимствовав понятие движения назад, противоположное прогрессу – движению вперед.

Впоследствии слово «регрессия» заняло в статистике заметное место, хотя, как это часто бывает в любом языке, в том числе и в языке науки, в него теперь вкладывают другой смысл – оно означает статистическую связь между случайными величинами.

6.7.1. Линейная регрессия

При обработке экспериментальных данных химики часто строят различные графики, пользуясь декартовой (прямоугольной) системой координат. Иногда зависимости между измеряемыми величинами x и y носят линейный характер, поэтому экспериментальные точки группируются около некоторой прямой линии (рис. 6.11).

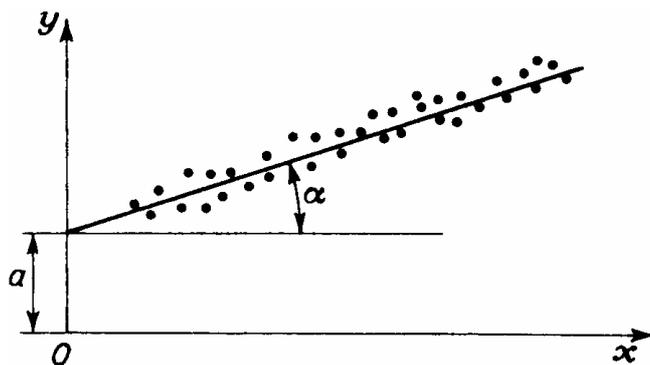


Рис. 6.11. Зависимость между измеряемыми величинами

Уравнение прямой линии имеет вид

$$y = a + bx.$$

Здесь a – длина отрезка от начала координат до точки пересечения прямой с осью Oy , b – тангенс угла наклона α прямой к оси Ox ($b = \operatorname{tg} \alpha$).

Прямую линию стараются провести так, чтобы сумма квадратов отклонений расчетных значений y_p от экспериментальных значений $y_э$ была минимальной для всех n рассматриваемых опытов:

$$s = \sum_{i=1}^n (y_p - y_э)^2 \rightarrow \min .$$

Другими словами, n – это общее число опытов, в которых измерялись значения величин x и y ; s – сумму квадратов отклонений.

Уравнение $y = a + bx$, в которое подставлены найденные значения коэффициентов, принято называть *уравнением линейной регрессии*. Пользуясь этим уравнением, можно, не проводя дальнейших опытов, рассчитать для заданного x соответствующее значение y . Таким образом, с помощью уравнения регрессии можно прогнозировать величину y . Разумеется, такой прогноз не будет абсолютно точным. Близость прогнозируемого значения y к фактическому значению зависит в основном от точности, с которой выполняется эксперимент, и от того, насколько существующая зависимость между y и x близка в действительности к линейной.

Расчеты коэффициентов линейной регрессии всегда выполняются по одним и тем же формулам и требуют при ручном счете значительных затрат времени. Чтобы избавить исследователей от этой рутинной работы, применяют ПК.

По программе вычисляются не только коэффициенты уравнения регрессии, но и среднее квадратичное отклонение расчетных значений y от экспериментальных. Среднее квадратичное отклонение σ вычисляется по формуле

$$\sigma = \sqrt{\frac{s}{n-1}} .$$

Рассмотрим применение программы для выполнения практических расчетов.

Известно, что с повышением температуры растворимость солей в воде увеличивается. Напомним, что растворимостью называется концентрация соли в насыщенном растворе.

Экспериментальные данные по растворимости соли при различных температурах приведены в таблице 6.1.

Таблица 6.1

Температура (x), °С	10	20	30	40	50	60	70	80
Растворимость (y), г/л	42	50	62	67	75	86	92	103

Обработав экспериментальные данные с помощью компьютера, получаем уравнение регрессии $y = 33,71 + 0,85x$ и среднее квадратичное отклонение $\sigma = 1,46$. Программа оказалась достаточно простой и удобной для использования, но, к сожалению, область ее применения ограничена только линейными зависимостями. Однако нужно иметь в виду, что некоторые нелинейные зависимости можно преобразовать в линейные.

Например, гиперболическая связь $y = a_0 + a_1/x$ линеаризуется заменой переменной $z = 1/x$, тогда $y = a_0 + a_1z$.

Показательная связь $y = a \cdot e^{a_1x}$ линеаризуется путем логарифмирования: $v = \ln y = \ln a_0 + a_1x$.

Степенная связь после логарифмирования линеаризуется заменой $\ln y = p$, $\ln a_0 = c$, $\ln x = z$, тогда $p = c + a_1z$.

Логарифмическая связь $y = a_0 + a_1 \ln x$ линеаризуется заменой $z = \ln x$, тогда $y = a_0 + a_1z$.

Комбинированная связь $y = 1 / (a_0 + a_1 e^{-x})$ линеаризуется заменой $v = 1/y$ и $z = e^{-x}$, тогда $v = a_0 + a_1z$.

Широкое использование линейных зависимостей и связей, легко приводимых к линейным, объясняется следующим. Линейные связи просты и требуют относительно малого объема вычислений, и методика их установления более глубоко разработана.

Рассмотрим пример линеаризации. Как известно, зависимость константы скорости химической реакции от температуры имеет экспоненциальный характер:

$$k = k_0 e^{-\frac{E}{RT}}.$$

Прологарифмировав это выражение, получим

$$\ln k = \ln k_0 - \frac{E}{R} \frac{1}{T}.$$

Введем следующие обозначения:

$$y = \ln k; \quad a = \ln k_0; \quad b = -\frac{E}{R}; \quad x = \frac{1}{T}.$$

С учетом этих обозначений получаем линеаризованное уравнение регрессии $y = a + bx$. Теперь снова можно воспользоваться нашей программой. Вычислив коэффициенты a и b , можно рассчитать предэкспоненциальный множитель k_0 и энергию активации E .

Таким образом, благодаря линеаризации область применения программы значительно расширяется.

6.7.2. Корреляция

Желая установить связь между двумя величинами, исследователь наносит их значения на график. Каждой паре значений величин x и y на графике соответствует точка. Обычно такие точки не ложатся на одну линию, а занимают на плоскости чертежа некоторую область, образуя так называемую диаграмму рассеивания (рис. 6.12). Причины рассеивания могут быть самыми различными. Среди них можно назвать погрешности измерений, влияние неучтенных факторов и т.д. Даже если один и тот же опыт в химической лаборатории проводится несколько раз, результаты измерений обычно отличаются друг от друга.

Связь между двумя величинами, характеризующими химико-технологический процесс, можно выразить уравнением регрессии. Однако оно не позволяет с абсолютной точностью предсказать значение y , если известно значение x . Прогнозирование y осуществляется только с некоторой вероятностью. Такая связь между двумя величинами называется корреляционной. Она существенно отличается от связи детерминистической, описываемой строгими физико-химическими формулами.

Благодаря рассеянию, точки на графике располагаются в некоторой конечной области, обычно имеющей форму эллипса. Чем теснее связь между рассматриваемыми величинами, тем более вытянут эллипс вдоль одной из своих осей (см. рис. 6.12, *а* и *б*). Наоборот, если связь между x и y отсутствует, то разброс точек на диаграмме рассеивания по форме приближается к кругу (см. рис. 6.12, *в*).

Корреляционную связь между величинами называют положительной, если при увеличении одной из них возрастает другая величина. Отрицательной называют такую корреляционную связь, когда при увеличении значений одной из величин другая уменьшается.

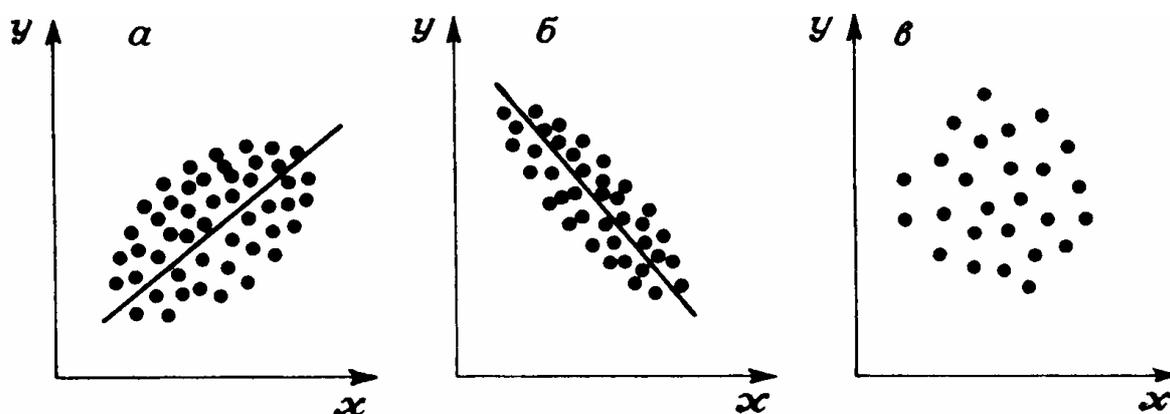


Рис. 6.12. Диаграммы рассеивания

Видя на графике корреляционную связь между величинами, мы можем дать ей качественную оценку. Для количественной оценки линейной корреляции используется коэффициент парной корреляции r_{xy} , который рассчитывается по формуле

$$r_{xy} = \frac{s_1 s_5 - s_2 s_3}{\sqrt{(s_1 s_4 - s_2^2)(s_1 s_6 - s_3^2)}};$$

где $s_1 = n$; $s_2 = \sum_{i=1}^n x_i$; $s_3 = \sum_{i=1}^n y_i$; $s_4 = \sum_{i=1}^n x_i^2$; $s_5 = \sum_{i=1}^n x_i y_i$; $s_6 = \sum_{i=1}^n y_i^2$.

Этот коэффициент может принимать следующие значения:

1) $r_{xy} = 0$, что свидетельствует об отсутствии корреляционной связи между x и y ;

2) $r_{xy} = 1$, в данном случае существует строгая положительная детерминистическая связь;

3) $r_{xy} = -1$; между x и y существует строгая отрицательная детерминистическая связь;

4) $-1 < r_{xy} < +1$, это наиболее распространенный случай: корреляционная связь может быть как положительной, так и отрицательной, и характеризоваться различной степенью тесноты связи.

Чем ближе абсолютное значение коэффициента корреляции $|r_{xy}|$ к единице, тем сильнее линейная связь между x и y . Следует отметить, что r_{xy} одновременно отражает степень случайности и криволинейности связи между величинами x и y . Например, зависимость y от x может быть близкой к функциональной, но существенно нелинейной. В этом случае коэффициент корреляции будет значительно меньше единицы.

Таким образом, тесноту связи между двумя величинами можно объективно оценить только после рассмотрения графика «корреляционного поля» и вычисления коэффициента парной корреляции.

7. ВВЕДЕНИЕ В ТЕХНОЛОГИЮ БАЗ ДАННЫХ

База данных – это организованная структура, предназначенная для хранения информации. Это утверждение легко пояснить, если, например, рассмотреть БД крупного банка. В ней есть все необходимые сведения о клиентах, их адресах, кредитной истории, состоянии расчетных счетов, финансовых операциях и т.д. Доступ к этой базе имеется у достаточно большого количества сотрудников банка, но среди них вряд ли найдется такое лицо, которое имеет доступ *ко всей базе полностью* и при этом способно единолично вносить в нее произвольные изменения. Кроме данных, база содержит *методы и средства*, позволяющие каждому из сотрудников оперировать только с теми данными, которые входят в его компетенцию. В результате взаимодействия данных, содержащихся в базе, с методами, доступными конкретным сотрудникам, образуется информация, которую они потребляют и на основании которой в пределах собственной компетенции производят ввод и редактирование данных.

С понятием *БД* тесно связано понятие *системы управления базой данных*. Это комплекс программных средств, предназначенных для создания структуры новой базы, наполнения ее содержимым, редактирования содержимого и визуализации информации. Под *визуализацией информации* понимается отбор отображаемых данных в соответствии с заданным критерием, их упорядочение, оформление и последующая выдача на устройство вывода или по каналам связи.

В мире существует множество СУБД. Несмотря на то что они могут по-разному работать с разными объектами и предоставляют пользователю различные функции и средства, большинство СУБД опираются на единый устоявшийся комплекс основных понятий. Это дает нам возможность рассмотреть одну систему и обобщить ее понятия, приемы и методы на весь класс СУБД. В качестве такого учебного объекта мы выберем СУБД Microsoft Access, входящую в пакет Microsoft Office. В тех случаях, когда конкретные приемы операций зависят от используемой версии программы, мы будем опираться на Microsoft Access, хотя в основном речь будет идти о таких обобщенных понятиях и методах, для которых различия между конкретными версиями программ второстепенны.

7.1. Структура простейшей базы данных

Сразу поясним, что если в базе нет никаких данных (*пустая база*), то это все равно полноценная БД. Этот факт имеет методическое значение.

Хотя данных в базе и нет, но информация в ней все-таки есть – это *структура базы*. Она определяет методы занесения данных и хранения их в базе. Простейший «некомпьютерный» вариант БД – деловой ежедневник, в котором каждому календарному дню выделено по странице. Даже если в нем не записано ни строки, он не перестает быть ежедневником, поскольку имеет структуру, четко отличающую его от записных книжек, рабочих тетрадей и прочей писчебумажной продукции.

Базы данных могут содержать различные объекты, но основными объектами любой БД являются ее таблицы. Простейшая БД содержит хотя бы одну таблицу. Соответственно, структура простейшей БД тождественно равна структуре ее таблицы.

Известно, что структуру двумерной таблицы образуют столбцы и строки. Их аналогами в структуре простейшей БД являются *поля* и *записи* (рис. 7.1). Если записей в таблице пока нет, значит ее структура образована только набором полей. Изменив состав полей базовой таблицы (или их свойства), мы изменяем структуру БД и, соответственно, получаем новую базу данных.

Книги в продаже : таблица				
	Наименование книги	Автор	Цена	Поставщик
	Банковские технологии	Эшли	134,00р.	ООО "Звезда"
	Предпринимательское право	Несбит	64,00р.	ЗАО "Букинист"
	Электронная коммерция	Новиков	78,00р.	ООО "Раритет"
	Справочник предпринимателя		60,00р.	ООО "Парк"
			0,00р.	

Рис. 7.1. Простейшая таблица базы данных

7.2. Свойства полей базы данных (на примере СУБД MS ACCESS)

Поля БД определяют групповые свойства данных, записываемые в ячейки, принадлежащие каждому из полей. Перечислим основные свойства полей таблиц БД:

1. **Имя поля** – определяет, как следует обращаться к данным этого поля при автоматических операциях с базой (по умолчанию имена полей используются в качестве заголовков столбцов таблиц).

2. **Тип поля** – определяет тип данных, которые могут содержаться в данном поле.

3. **Размер поля** – определяет предельную длину (в символах) данных, которые могут размещаться в данном поле.

4. **Формат поля** – определяет способ форматирования данных в ячейках, принадлежащих полю.

5. **Маска ввода** – определяет форму, в которой вводятся данные в поле (средство автоматизации ввода данных).

6. **Подпись** – определяет заголовок столбца таблицы для данного поля (если подпись не указана, то в качестве заголовка столбца используется свойство имя поля).

7. **Значение по умолчанию** – то значение, которое вводится в ячейки поля автоматически (средство автоматизации ввода данных).

8. **Условие на значение** – ограничение, используемое для проверки правильности ввода данных (средство автоматизации ввода данных, которое используется, как правило, для данных, имеющих числовой тип, денежный тип или тип даты).

9. **Сообщение об ошибке** – текстовое сообщение, которое выдается автоматически при попытке ввода в поле ошибочных данных (проверка ошибочности выполняется автоматически, если задано свойство «Условие на значение»).

10. **Обязательное поле** – свойство, определяющее обязательность заполнения данного поля при наполнении базы.

11. **Пустые строки** – свойство, разрешающее ввод пустых строковых данных.

12. **Индексированное поле** – если поле обладает этим свойством, все операции, связанные с поиском или сортировкой записей по значению, хранящемуся в данном поле, существенно ускоряются. Кроме того, для индексированных полей можно сделать так, что значения в записях будут проверяться по этому полю на наличие повторов, что позволяет автоматически исключить дублирование данных.

Следует помнить, что поскольку в разных полях могут содержаться данные разного типа, то и наборы свойств у полей могут различаться в зависимости от типа данных.

Так, например, список вышеуказанных свойств полей относится в основном к полям текстового типа. Поля других типов могут иметь или не иметь эти свойства, но могут добавлять к ним и свои. Например, для данных, представляющих действительные числа, важным свойством является

количество знаков после десятичной запятой. С другой стороны, для полей, используемых для хранения рисунков, звукозаписей, видеоклипов и других объектов *OLE*, большинство вышеуказанных свойств не имеют смысла.

7.3. Типы данных

Базы данных MS Access работают со следующими типами данных:

1. **Текстовый** – тип данных, используемый для хранения обычного неформатированного текста ограниченного размера (до 255 символов).

2. **Поле Мемо** – специальный тип данных для хранения больших объемов текста (до 65 535 символов). Физически текст не хранится в поле. В поле хранится указатель на него.

3. **Числовой** – тип данных для хранения действительных чисел.

4. **Дата/время** – тип данных для хранения календарных дат и текущего времени.

5. **Денежный** – тип данных для хранения денежных сумм.

6. **Счетчик** – специальный тип данных для уникальных (не повторяющихся в поле) натуральных чисел с автоматическим наращиванием. Естественное использование – для порядковой нумерации записей.

7. **Логический** – тип для хранения логических данных, которые могут принимать только два значения, например, *Да* или *Нет*.

8. **Поле объекта OLE** – специальный тип данных, предназначенный для хранения объектов, вставляемых внедрением или связыванием (OLE). Реально такие данные в таблице не хранятся, в таблице хранятся только указатели на них.

9. **Гиперссылка** – специальное поле для хранения адресов URL Web-объектов Интернета. При щелчке на ссылке автоматически происходит запуск браузера и воспроизведение объекта в его окне.

10. **Мастер подстановок** – это неспециальный тип данных. Это объект, настройкой которого можно автоматизировать ввод данных в поле так, чтобы не вводить их вручную, а выбирать из раскрывающегося списка.

7.4. Объекты базы данных

Система управления базой данных MS Access позволяет создавать и использовать объекты следующих типов.

Таблицы. Является основным объектом БД, т.к. 1) в таблицах хранятся все данные, имеющиеся в базе; 2) таблицы хранят структуру базы (поля, их типы и свойства).

Запросы. Эти объекты служат для извлечения данных из таблиц и предоставления их пользователю в удобном виде. С помощью запросов выполняют такие операции, как отбор данных, их сортировку и фильтрацию. С помощью запросов можно выполнять преобразование данных по заданному алгоритму, создавать новые таблицы, выполнять автоматическое наполнение таблиц данными, импортированными из других источников, выполнять простейшие вычисления в таблицах и др.

Начинающие пользователи не сразу понимают роль запросов, поскольку все те же операции можно делать и с таблицами. Да, действительно, это так, но есть соображения удобства (в первую очередь быстродействия) и соображения безопасности.

Из соображений безопасности, чем меньше доступа к базовым таблицам имеют конечные пользователи, тем лучше. Во-первых, снижается риск того, что неумелыми действиями они повредят данные в таблицах. Во-вторых, предоставив разным пользователям разные запросы, можно эффективно разграничить их доступ к данным в строгом соответствии с кругом персональных обязанностей. В банках, например, одни сотрудники имеют доступ к таблицам данных о клиентах, другие к их расчетным счетам, третьи – к таблицам активов банка. Если и есть специальные службы, имеющие доступ ко всем информационным ресурсам банка (с целью контроля и анализа), то они лишены средств для внесения изменений – все сделано так, чтобы один человек не мог совершить фиктивную операцию, независимо с того, какую должность он занимает. В базе данных, имеющей правильно организованную структуру, для совершения противоправных действий необходим сговор нескольких участников, а такие действия пресекаются не программными, а традиционными средствами обеспечения безопасности.

Особенность запросов состоит в том, что они черпают данные из базовых таблиц и создают на их основе временную *результующую таблицу*. Если хотят подчеркнуть факт «временности» этой таблицы, то ее еще называют *моментальным снимком*. Когда мы работаем с *основными таблицами* базы, мы физически имеем дело с жестким диском, т.е. с очень медленным устройством. Когда же на основании запроса мы получаем *результующую таблицу*, то имеем дело с электронной таблицей, не имеющей аналога на жестком диске, – это только *образ* отобранных полей и записей. Разумеется, работа с «образом» происходит гораздо быстрее и эффективнее – это еще одно основание для того, чтобы широко использовать запросы.

Формы. Это средства для ввода данных. Смысл их в том, чтобы предоставить пользователю средства для заполнения только тех полей, которые ему заполнять положено. Одновременно с этим в форме можно разместить специальные элементы управления (счетчики, раскрывающиеся списки, переключатели, флажки и др.) для автоматизации ввода. Преимущества форм раскрываются особенно наглядно, когда происходит ввод данных с заполненных бланков. В этом случае форму делают графически средствами так, чтобы она повторяла оформление бланка, – это заметно упрощает работу наборщика. С помощью форм данные не только вводятся, но и отображаются.

Отчеты. Предназначены для формирования выходного документа, предназначенного для вывода на печать. В связи с этим отчеты отличаются тем, что в них приняты специальные меры для группирования выводимых данных и для вывода специальных элементов оформления, характерных для печатных документов (верхний и нижний колонтитулы, номера страниц, служебная информация о времени создания отчета и т.п.).

Страницы. Это специальные объекты БД, реализованные в СУБД Microsoft Access, XP. Правда, более корректно их называть *страницами доступа к данным*. Физически это особый объект, выполненный в коде HTML, размещаемый на Web-странице и передаваемый клиенту вместе с ней. Сам по себе этот объект не является БД, но содержит компоненты, через которые осуществляется связь переданной Web-страницы с базой данных, остающейся на сервере. Пользуясь этими компонентами, посетитель Web-узла может просматривать записи базы в полях страницы доступа. Таким образом, страницы доступа к данным осуществляют интерфейс между клиентом, сервером и БД, размещенной на сервере. Эта БД необязательно должна быть базой данных Microsoft Access. Страницы доступа, созданные средствами Microsoft Access, позволяют работать также с БД Microsoft SQL Server.

Макросы. Состоят из последовательности внутренних команд СУБД и являются одним из средств автоматизации работы с БД.

Модули. Содержат программы на Visual Basic for Applications, которые могут разрабатываться пользователем для реализации нестандартных процедур при создании приложения.

7.5. Проектирование баз данных. Режимы работы с базами данных

Обычно с БД работают две категории исполнителей. Первая категория – *проектировщики*. Их задача состоит в разработке структуры таблиц БД и согласовании ее с заказчиком. Кроме таблиц проектировщики разрабатывают и другие объекты БД, предназначенные, с одной стороны, для автоматизации работы с базой, а с другой стороны – для ограничения функциональных возможностей работы с базой (если это необходимо из соображений безопасности). Проектировщики не наполняют базу конкретными данными (заказчик может считать их конфиденциальными и не предоставлять посторонним лицам). Исключение составляет экспериментальное наполнение модельными данными на этапе отладки объектов базы.

Вторая категория исполнителей, работающих с БД, – *пользователи*. Они получают исходную БД от проектировщиков и занимаются ее наполнением и обслуживанием. В общем случае пользователи не имеют средств доступа к управлению структурой базы – только к данным, да и то не ко всем, а к тем, работа с которыми предусмотрена на их конкретном рабочем месте.

Соответственно, СУБД имеет два режима работы: *проектировочный* и *пользовательский*. Первый режим предназначен для создания или изменения структуры базы и создания ее объектов. Во втором режиме происходит использование ранее подготовленных объектов для наполнения базы или получения данных из нее.

7.6. Взаимодействие заказчика базы данных с разработчиком

7.6.1. Разработка технического задания

Техническое задание на проектирование БД должен предоставить заказчик. Для этого он должен владеть соответствующей терминологией и знать, хотя бы в общих чертах, технические возможности основных СУБД. К сожалению, на практике такое положение встречается не всегда. Поэтому обычно используют следующие подходы:

- разработчик демонстрирует заказчику работу аналогичной БД, после чего заказчик высказывает свои пожелания, и они согласовывают спецификацию отличий;
- если аналога нет, разработчик выясняет круг задач и потребностей заказчика, после чего помогает ему подготовить техническое задание.

При подготовке технического задания составляют:

- список исходных данных, с которыми работает заказчик;
- список выходных данных, которые необходимы заказчику для управления структурой своего предприятия;
- список выходных данных, которые заказчик должен предоставлять в другие организации (в вышестоящие структуры, в органы статистического учета, прочие административные и контролирующие организации).

При этом очень важно не ограничиваться взаимодействием с головным подразделением заказчика, а провести обследование всех служб и подразделений, которые могут оказаться поставщиками данных в базу или их потребителями. К возможным пожеланиям заказчика следует готовиться на этапе проектирования до создания базы.

7.6.2. Основные этапы проектирования БД

Основные этапы проектирования БД представлены на рисунке 7.2.

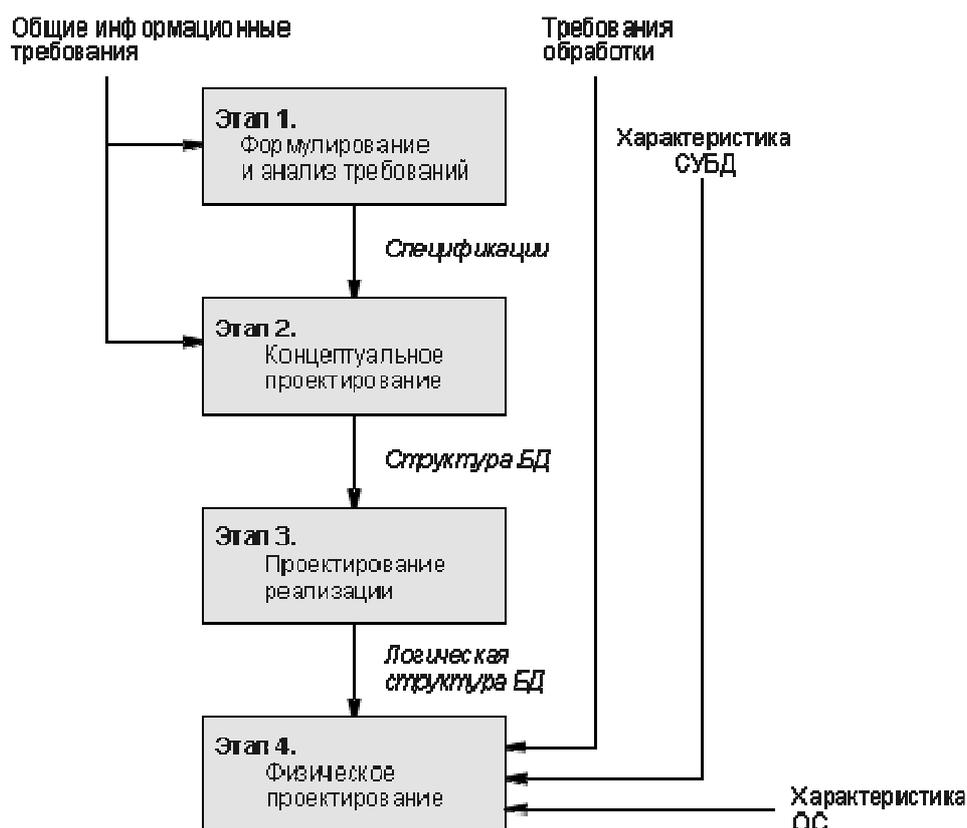


Рис. 7.2. Этапы проектирования БД

Этап 1. Цели первого этапа состоят в следующем:

- 1) изучение области деятельности моделируемой организации;
- 2) определение информационных потребностей организации;

- 3) документирование требований к БД в форме, доступной как конечному пользователю, так и проектировщику базы данных;
- 4) собранные и документированные требования должны включать ограничения, обуславливающие безопасность и надежность.

Этап 2. Этап концептуального проектирования связан с описанием и синтезом разнообразных информационных требований пользователей в первоначальный проект БД. Результатом этого этапа является высокоуровневое представление информационных требований, описание которых может быть выполнено с помощью модели данных высокого уровня «**сущность-связь**». Модель «сущность-связь» (entity – relationship, ER) была задумана как средство представления *схемы предметной области*, не зависящей от СУБД.

Этап 3. Проектированием реализации будем называть этап между концептуальным и физическим проектированием, в результате выполнения которого получается СУБД-ориентированная схема БД, удовлетворяющая всем требованиям пользователей, начиная с требований целостности и непротиворечивости проектируемой БД и кончая показателями эффективности функционирования при ее расширении и усложнении. Вначале анализируется содержание требований обработки данных, затем формируются предварительные типы записей, допускаемых СУБД. Для отображения связей типа M:N в реляционной модели строится новое отношение. Далее схема может быть усовершенствована использованием такого средства СУБД, как индексация.

Этап 4. Основными составляющими в проектировании физической структуры БД являются проектирование формата хранимой записи, кластеризация хранимых записей и выбор метода доступа. Кластеризация – это выявление групп объектов с общими признаками и расположение их в смежных блоках внешнего запоминающего устройства. Проектирование физической БД прошло несколько этапов развития от проектирования файлов до проектирования физических структур интегрированных БД в рамках существующих в настоящее время СУБД. Современные СУБД используют физические структуры и методы доступа, опирающиеся на технологию проектирования файлов. Разработчика конкретной информационной системы обычно не интересует физическое хранение БД и методы доступа, т.к. этим управляет СУБД.

7.6.3. Разработка схемы данных

Выяснив основную часть данных, которые заказчик потребляет или поставляет, можно приступить к созданию структуры базы, т.е. структуры ее основных таблиц.

1. Работа начинается с составления генерального списка полей – он может насчитывать сотни позиций.

2. В соответствии с тем, какие данные размещаются в каждом поле, определяют наиболее подходящий тип для каждого поля.

3. Далее распределяют поля генерального списка по базовым таблицам. На первом этапе распределение производят по функциональному признаку. Принцип разделения очень прост: обеспечить, чтобы ввод данных в одну таблицу производился в рамках одного подразделения, а еще лучше – на одном рабочем месте.

Наметив столько таблиц, сколько подразделений (рабочих мест) охватывает БД, приступают к дальнейшему делению таблиц. Критерием необходимости деления является факт множественного повтора данных в соседних записях. (Выполняется нормализация таблиц).

4. В каждой из таблиц намечают *ключевое поле*. В качестве такового выбирают поле, данные в котором повторяться не могут. Например, для таблицы данных о студентах таким полем может служить индивидуальный шифр студента. Для таблицы со сведениями о постоянных подписчиках на услуги, предоставляемые фирмой, ключевым является номер контракта. Для складской таблицы ключевым может быть артикул товара.

Если в списке полей таблицы нет такого поля, которое можно было бы использовать как ключевое, его можно создать искусственно. Например, можно ввести дополнительное поле типа *Счетчик* – оно не может содержать повторяющиеся данные по определению.

Есть еще прием объединения двух неуникальных полей в одно уникальное (составной ключ). Так, например, в таблице расписания занятий время занятия – неуникальное поле, поскольку в одно и то же время занятия идут у разных групп. Номер аудитории тоже не уникальное поле, поскольку в одной аудитории могут заниматься разные группы. Но если объединить время занятия с номером аудитории, то такое составное поле станет уникальным, ведь в одно и то же время в одной аудитории две группы заниматься не могут. С таким искусственным полем может быть связан номер группы студентов, занимающихся в данное время в данной аудитории, и фамилия преподавателя, который этим занятием руководит.

5. Вычерчивают на бумаге линии связи между таблицами. С одной стороны, линия связи всегда маркируется знаком «1», с другой – либо зна-

ком «1» (связь *один к одному*), либо значком «бесконечность» (связь *один ко многим*). Понятно, что если связываются ключевые поля, то это всегда связь *один ко одному*, а если ключевое поле связано с неключевым, то это связь *один ко многим*. Такой чертеж называется *схемой данных* (рис. 7.3).

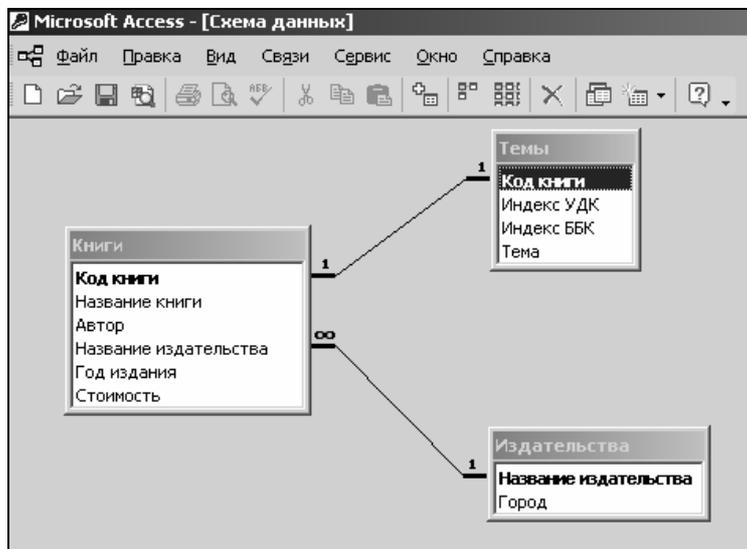


Рис. 7.3. Схема связей между таблицами

6. Разработкой схемы данных заканчивается «бумажный» этап работы над техническим предложением. Эту схему заказчик согласовывает с исполнителем, и лишь после этого исполнитель может приступить к непосредственному созданию БД.

От успеха этапа предварительного проектирования зависит, насколько БД станет удобной и будут ли с ней работать пользователи. Если отмечается, что пользователи базы отказываются ее эксплуатировать и предпочитают работать традиционными методами, это говорит о недостаточной квалификации разработчика базы.

7.7. Работа с системой управления базой данных Microsoft Access

Система управления базой данных Microsoft Access предоставляет несколько средств создания каждого из основных объектов базы. Эти средства можно классифицировать как:

- ручные (разработка объектов в режиме *Конструктора*);
- автоматизированные (разработка с помощью программ-мастеров);
- автоматические – средства ускоренной разработки простейших объектов.

Ручные средства наиболее трудоемки, но обеспечивают максимальную гибкость; автоматизированные и автоматические средства являются наиболее производительными, но и наименее гибкими.

7.7.1. Работа с таблицами

Создание таблиц. Работа с любыми объектами начинается с окна *База данных* (см. рис. 7. 4). На его левой панели сосредоточены элементы управления для вызова всех семи типов объектов программы. Создание таблиц начинается с выбора элемента управления *Таблицы*.

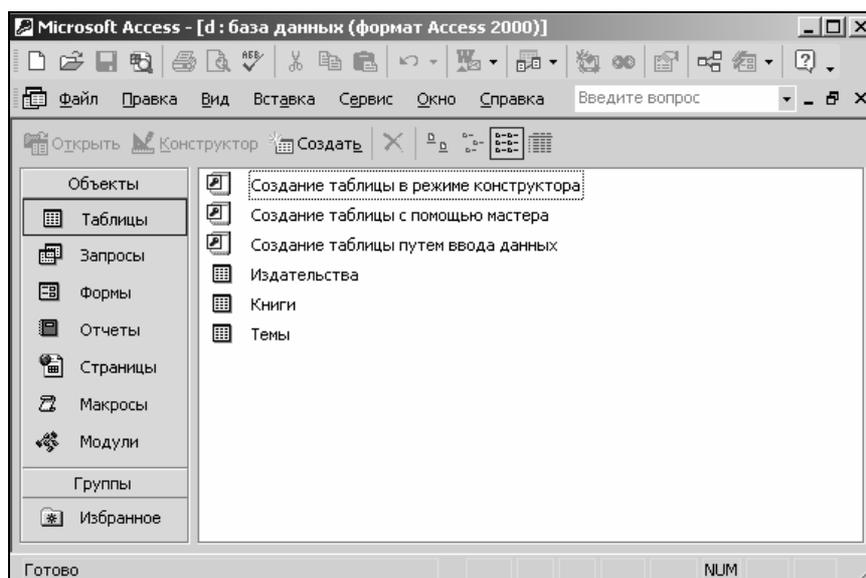


Рис. 7.4. Окно *База данных* является исходным элементом управления программы Microsoft Access

На правой панели представлен список таблиц, уже имеющихся в составе базы, и приведены элементы управления для создания новой таблицы. Чтобы создать таблицу вручную, следует использовать значок *Создание таблицы в режиме конструктора*.

Окно *Конструктора таблиц* представлено на рис. 7.5.

То, что мы видим в этом режиме, фактически является графическим бланком для создания и редактирования структуры таблиц. В первом столбце вводят имена полей. Если свойство *Подпись* для поля не задано, то *Имя поля* станет одновременно и именем столбца будущей таблицы. Тип для каждого поля выбирают из раскрывающегося списка, открываемого кнопкой выбора типа поля. Эта кнопка – скрытый элемент управления. Она отображается только после щелчка на поле бланка. (Это надо иметь в виду – в Microsoft Access очень много таких скрытых элементов управления, которые не отображаются, пока ввод данных не начат).

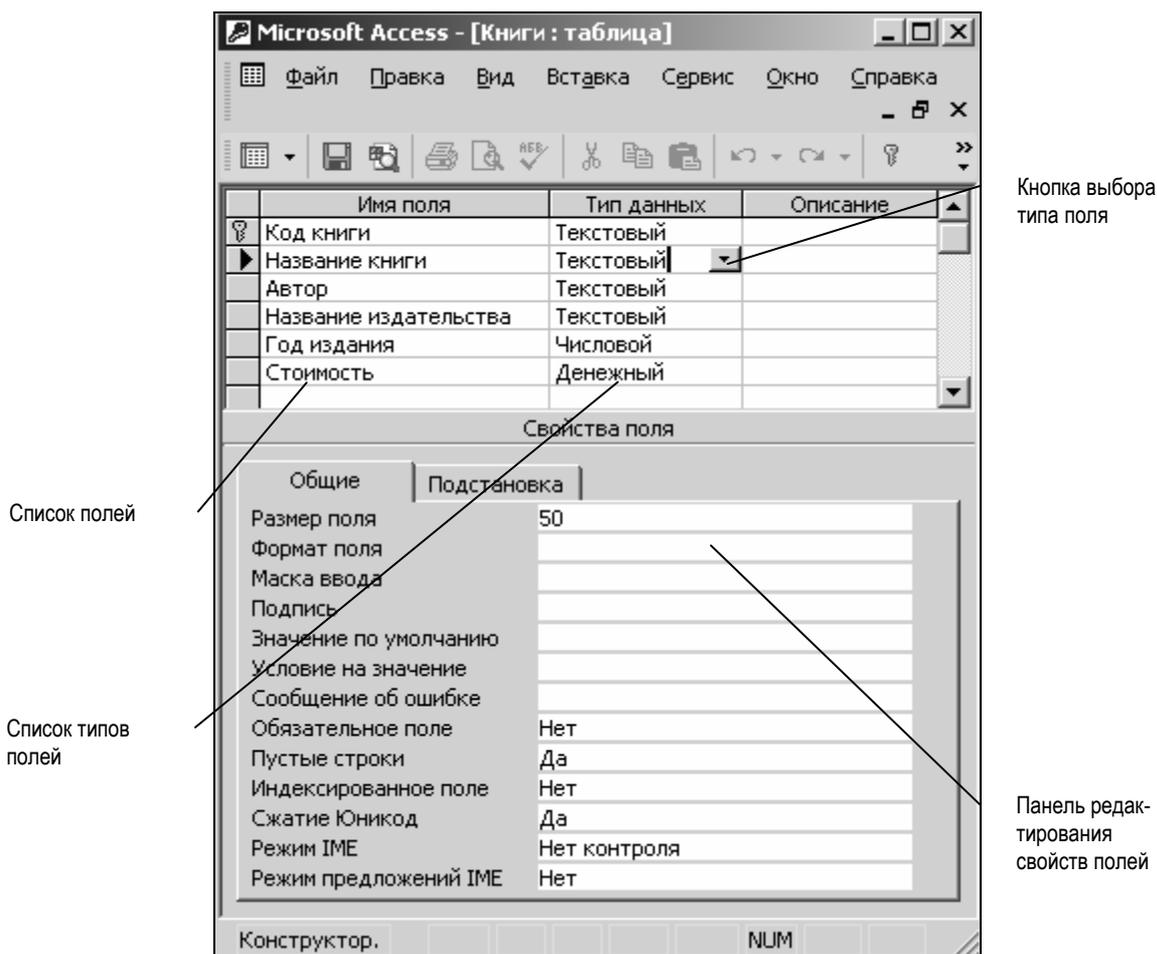


Рис. 7.5. Проектирование структуры таблицы

Нижняя часть бланка содержит список свойств поля, выделенного в верхней части. Некоторые из свойств уже заданы по умолчанию. Свойства полей не являются обязательными. Их можно настраивать по желанию, а можно и не трогать.

При создании таблицы целесообразно (хотя и необязательно) задать ключевое поле. Это поможет впоследствии, при организации связей между таблицами. Для задания ключевого поля достаточно щелкнуть на его имени правой кнопкой мыши и в открывшемся контекстном меню выбрать пункт *Ключевое поле*.

Создание составного ключа выполняется точно так же, но предварительно надо выделить те поля, которые войдут в составной ключ. Групповое выделение выполняют при нажатой клавише Shift щелчками на квадратных маркерах слева от имен полей.

Закончив создание структуры таблицы, бланк закрывают (при этом система выдает запрос на сохранение таблицы), после чего дают таблице имя, и с этого момента она доступна в числе прочих таблиц в основном окне *База данных*. Оттуда ее и можно открыть в случае необходимости.

Созданную таблицу открывают в окне *База данных* двойным щелчком на ее значке. Новая таблица не имеет записей – только названия столбцов, характеризующие структуру таблицы (рис. 7.6).



Рис. 7.6. Пример новой таблицы

Заполнение таблицы данными производится обычным порядком. Курсор ввода устанавливается в нужную ячейку указателем мыши. Переход к следующей ячейке можно выполнить клавишей Tab. Переход к очередной записи выполняется после заполнения последней ячейки.

В нижней части таблицы расположена *Панель кнопок перехода*. Ее элементами управления удобно пользоваться при навигации по таблице, имеющей большое число записей.

Пользователям Microsoft Access доставляет неудобство тот факт, что данные не всегда умещаются в ячейках таблицы. Шириной столбцов можно управлять методом перетаскивания их границ. Удобно использовать автоматическое форматирование столбцов «по содержимому». Для этого надо установить указатель мыши на правую границу столбца (в строке заголовков столбцов), дождаться, когда указатель сменит форму, и выполнить двойной щелчок.

После наполнения таблицы данными сохранять их не надо – все сохраняется автоматически. Однако если при работе с таблицей произошло редактирование ее макета (например, изменялась ширина столбцов), СУБД попросит подтвердить сохранение этих изменений.

Если возникнет необходимость изменить структуру таблицы (состав полей или их свойства), таблицу надо открыть в режиме *Конструктора*. Для этого ее следует выделить в окне *База данных* и щелкнуть на кнопке *Конструктор*.

Создание межтабличных связей. У связи два основных назначения: 1) обеспечение целостности данных; 2) автоматизация задач обслуживания базы.

Связь между таблицами позволяет:

- либо исключить возможность удаления или изменения данных в ключевом поле главной таблицы, если с этим полем связаны какие-либо поля других таблиц;
- либо сделать так, что при удалении (или изменении) данных в ключевом поле главной таблицы автоматически (и абсолютно корректно) произойдет удаление или изменение соответствующих данных в полях связанных таблиц.

Вся необходимая работа по созданию связей происходит в специальном окне *Схема данных*. Это окно открывают соответствующей кнопкой на панели инструментов  или командой *Сервис* → *Схема данных*.

Образовавшаяся межтабличная связь отображается в окне *Схема данных* в виде линии, соединяющей два поля разных таблиц. При этом одна из таблиц считается *главной*, а другая – *связанной*. Главная – это та таблица, которая участвует в связи своим ключевым полем (название этого поля на схеме данных отображается полужирным шрифтом).

Для настройки свойств связи надо в окне *Схема данных* выделить линию, соединяющую поля двух таблиц, щелкнуть на ней правой кнопкой мыши и открыть контекстное меню связи, после чего выбрать в нем пункт *Изменить связь* – откроется диалоговое окно *Изменение связи*. В нем показаны названия связанных таблиц и имена полей, участвующих в связи (здесь же их можно изменить), а также приведены элементы управления для обеспечения условий целостности данных.

Если установлен только флажок *Обеспечение целостности данных*, то удалять данные из ключевого поля главной таблицы нельзя. Если вместе с ним включены флажки *Каскадное обновление связанных полей* и *Каскадное удаление связанных записей*, то соответственно операции *редактирования* и *удаления* данных в ключевом поле главной таблицы разрешены, но сопровождаются автоматическими изменениями в связанной таблице.

Таким образом, смысл создания реляционных связей между таблицами состоит, с одной стороны, в защите данных, а с другой – в автоматизации внесения изменений сразу в несколько таблиц при изменениях в одной таблице.

7.7.2. Работа с запросами

Если структура БД предприятия хорошо продумана, то исполнители, работающие с базой, не должны выполнять никаких действий с таблицами. Таблицы – слишком ценные объекты базы, чтобы с ними имел дело кто-либо, кроме разработчика базы.

Если исполнителю надо получить данные из базы, он должен использовать специальные объекты – *запросы*. Если запрос уже существует, надо открыть панель *Запросы* в окне *База данных*, выбрать его и открыть двойным щелчком на значке – откроется *результатирующая таблица*, в которой исполнитель найдет то, что его интересует.

Поля результирующей таблицы могут представлять набор из полей разных таблиц, а ее записи могут содержать отфильтрованные и отсортированные записи таблиц, на основе которых формировался запрос. Лишь в тех случаях, когда исполнитель не находит нужных данных в результирующей таблице, возникает необходимость готовить новый запрос – это задача разработчика базы.

Простые запросы лучше готовить вручную, с помощью *Конструктора*. Как и в случае с таблицами, для этого есть специальный значок в окне *База данных*. Он называется *Создание запроса в режиме конструктора* и открывает специальный бланк, называемый *бланком запроса по образцу*.

Бланк запроса по образцу представлен на рис. 7.7.

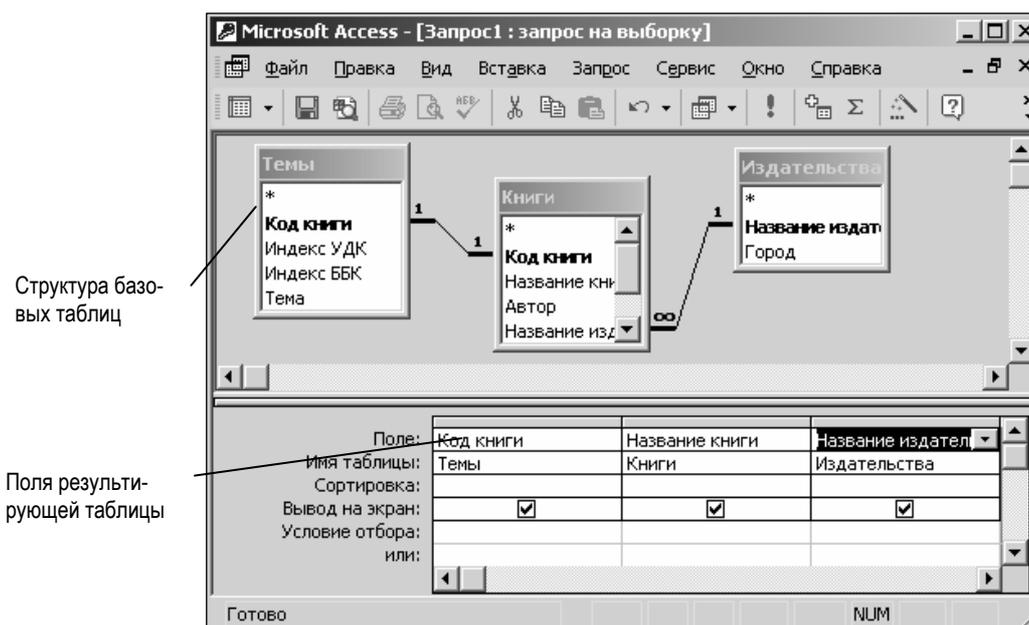


Рис. 7.7. Бланк запроса по образцу

Он состоит из двух областей. В верхней области отображается структура таблиц, к которым запрос адресован, а нижняя разбита на столбцы – по одному столбцу на каждое поле будущей результирующей таблицы. В бланке запроса определяются включаемые поля, условия отбора и порядок сортировки.

Порядок формирования запроса по образцу: после того, как был открыт бланк запроса, следует добавление таблиц, из которых выбираются данные. Добавить необходимые таблицы можно либо с помощью окна *Добавление таблицы* (после добавления нужных таблиц это окно следует закрыть), либо с помощью контекстного меню верхней половины бланка. Затем в этих таблицах выбирают двойными щелчками те поля, которые должны войти в результирующую таблицу. При этом автоматически заполняются столбцы в нижней части бланка. Сформировав структуру запроса, его закрывают, дают ему имя и в дальнейшем запускают двойным щелчком на значке в окне *База данных*.

Порядок действий, рассмотренный выше, позволяет создать простейший запрос, называемый *запросом на выборку*. Запросы этого типа возвращают данные из одной или нескольких таблиц и отображают их в виде таблицы, записи в которой можно обновлять (с некоторыми ограничениями). Запросы на выборку можно также использовать для группировки записей и вычисления сумм, средних значений, подсчета записей и нахождения других типов итоговых значений.

Упорядочение записей в результирующей таблице. По умолчанию данные отображаются в порядке, установленном для первичного ключа первой выбранной таблицы. Если первичного ключа нет, данные отображаются в порядке отбора записей. Для задания другого порядка сортировки в нижней части бланка имеется специальная строка *Сортировка*. При щелчке на этой строке открывается кнопка раскрывающегося списка, в котором можно выбрать метод сортировки: по возрастанию или по убыванию. Текстовые данные сортируются по возрастанию от А до Z (от А до Я), а числовые – от 0 до 9. В результирующей таблице данные будут отсортированы по тому полю, для которого задан порядок сортировки.

Возможна многоуровневая сортировка сразу по нескольким полям. В этом случае данные сначала сортируются по тому полю, которое в бланке запроса по образцу находится левее, затем по следующему полю, для которого включена сортировка, и так далее слева направо. Соответственно, при формировании запроса надо располагать поля результирующей таблицы с учетом будущей сортировки. В крайнем случае, если запрос уже сформирован, и надо изменить порядок следования столбцов, пользуются следующим приемом:

- выделяют столбец щелчком на его заголовке (кнопку мыши отпускают);
- еще раз щелкают на заголовке уже выделенного столбца (но кнопку не отпускают);
- перетаскивают столбец на новое место.

Управление отображением данных в результирующей таблице. В нижней части бланка *запроса по образцу* имеется строка *Вывод на экран*. По умолчанию предполагается, что все поля, включенные в запрос, должны выводиться на экран, но это не всегда целесообразно. Например, бывают случаи, когда некое поле необходимо включить в запрос только потому, что оно является полем сортировки, но, в то же время, нежелательно, чтобы пользователь базы видел его содержание. В таких случаях отображение содержимого на экране подавляют сбросом флажка *Вывод на экран*.

Использование условия отбора. Дополнительным средством, обеспечивающим отбор данных по заданному критерию, является так называемое *Условие отбора*. Соответствующая строка имеется в нижней части бланка запроса по образцу. Для того чтобы задать условие отбора для поля в бланке запроса, введите выражение в ячейку *Условие отбора* для данного поля.

Приведем примеры типичных выражений, используемых в запросах (табл. 7.1).

Таблица 7.1

Наиболее типичные выражения, используемые в запросах

Поле	Условие	Результат
Фамилия	«Иванов»	Извлекает записи, содержащие в поле <i>Фамилия</i> строку <i>Иванов</i>
Страна	«Испания»	Извлекает записи, содержащие в поле <i>Страна</i> значение <i>Испания</i>
Страна	«Германия» or «Испания»	Извлекает записи, содержащие в поле <i>Страна</i> названия <i>Германия</i> или <i>Испания</i>
Страна	not «Венесуэла»	Извлекает записи, содержащие в поле <i>Страна</i> любое название, кроме <i>Венесуэла</i>
Страна	Like [А-Д]	Извлекает записи с названиями стран, которые начинаются на А-Д
<i>Примечание</i> – Приведенные в этих строках условия, используемые для отбора данных, являются текстовыми условиями!		
Цена	>=30.00	Извлекает записи, содержащие в поле <i>Цена</i> , значение, не меньшее 30.00
Цена	Between 30.00 and 40.00	Извлекает записи, содержащие в поле <i>Цена</i> , значения от 30.00 до 40.00
Дата заказа	Between 3/1/99 and 31/12/99	Извлекает записи, содержащие в поле <i>Дата, заказа</i> даты от 3/1/99 до 31/12/99
Телефон	(919) 555-????	Извлекает записи, содержащие номера телефонов, с кодом области 919, тремя начальными цифрами 555 и любыми четырьмя цифрами
<i>Примечание</i> – Приведенные в этих строках условия, используемые для отбора данных, являются числовыми условиями!		

Математические операторы, используемые в числовых запросах.

Оператор	Значение
=	Равно
<	Меньше
>	Больше
<=	Меньше или равно (не больше)
>=	Больше или равно (не меньше)
<>	Не равно
between	Между

Использование условий для дат. Математические операторы используются в выражениях условий, использующих поля дат. Даты вводятся в любом принятом формате. Например:

#23/6/99#

23/6/99

23-Июн-99.

В условиях для дат можно использовать функцию *Date()* для отбора записей, содержащих текущую дату или попадающих в определенный временной интервал относительно текущей даты. При вводе выражения *Date()* в строке *Условие отбора* поле записей будет иметь значение, соответствующее текущей дате. Можно использовать такие выражения, как *Between 15/6/99 and Date()*, – для включения всех записей с датами между 15 июня 1999 г. и текущей датой. Можно также использовать такие выражения, как *Between Date() and Date()+30*, – для выбора всех записей со значением даты между текущей датой и 30 следующими днями.

Использование условий с логической операцией AND (И). Для выбора подмножества данных, удовлетворяющих нескольким условиям, в соответствующие поля бланка запроса необходимо ввести условия в определенном формате. Такие запросы используют логические операции *AND (И)*, поскольку для выбора записи должны удовлетворяться первое и второе условия. Например:

Поле:	Город	Область	Почтовый индекс	Страна	Телефон
Имя таблицы:	Покупатели	Покупатели	Покупатели	Покупатели	Покупатели
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>				
Условие отбора:	«Мадрид»			«Испания»	
или:					

В этом запросе выбираются записи, которые имеют значение *Мадрид* в поле *Город* и *Испания* в поле *Страна*.

Использование условий с логической операцией OR (ИЛИ). Когда выбираются записи, удовлетворяющие хотя бы одному условию, используют логическую операцию *OR (ИЛИ)*, т.к. для выбора записи должно выполняться одно или другое условие. Для ввода нескольких условий, объединенных операцией *ИЛИ*, можно использовать произвольное количество дополнительных строк под строкой *Условие отбора* бланка запроса. Например:

Поле:	Город	Область	Почтовый индекс	Страна	Телефон
Имя таблицы:	Покупатели	Покупатели	Покупатели	Покупатели	Покупатели
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>				
Условие отбора:				«Германия»	
или:				«Испания»	
				«Франция»	

В этом запросе выбираются все записи, содержащие в поле *Страна*, либо *Германия*, либо *Испания*, либо *Франция*.

Комбинация операций AND (И) и OR (ИЛИ). В одном и том же запросе можно использовать комбинацию логических операций *AND (И)* и *OR (ИЛИ)*. Это позволяет создавать сложные вопросы. Например:

Поле:	Город	Область	Почтовый индекс	Страна	Телефон
Имя таблицы:	Покупатели	Покупатели	Покупатели	Покупатели	Покупатели
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>				
Условие отбора:		«Аризона»		«США»	
или:		«Оклахома»		«США»	
		«Квебек»		«Канада»	
				«Мексика»	

В этом примере запроса извлекаются данные, содержащие *Аризона* в поле *Область* и *США* в поле *Страна*; или *Оклахома* в поле *Область* и *США* в поле *Страна*; или *Квебек* в поле *Область* и *Канада* в поле *Страна*; или *Мексика* в поле *Страна*.

Другие виды запросов. Помимо запросов на выборку существуют и другие виды запросов, некоторые из которых выполняются на базе предварительно созданного запроса на выборку. К ним относятся, прежде всего:

1. **Запросы с параметром.** Такие запросы запрашивают необходимые условия при каждом выполнении. Например:

Поле:	Код покупателя	Дата	Комментарии
Имя таблицы:	Телефонные звонки	Телефонные звонки	Телефонные звонки
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		Between [Введите начальную дату] and [Введите конечную дату]	
или:			

Текст, введенный в квадратных скобках, называется параметром.

В этом запросе выбираются все записи, содержащие информацию о телефонных звонках, поступивших от покупателей между датами, вводимыми пользователем с клавиатуры каждый раз при выполнении этого запроса.

2. Запросы на изменение. Запросом на изменение называют запрос, который за одну операцию изменяет или перемещает несколько записей. Существует четыре типа запросов на изменение.

а) *на удаление записи.* Удаляет группу записей из одной или нескольких таблиц. Например, запрос на удаление позволяет удалить записи о товарах, поставки которых прекращены или на которые нет заказов. С помощью запроса на удаление можно удалять только всю запись, а не отдельные поля внутри нее.

б) *на обновление записи.* Вносит общие изменения в группу записей одной или нескольких таблиц. Например, на 10% поднимаются цены на все молочные продукты или на 5% увеличивается зарплата сотрудников определенной категории. Запрос на обновление записей позволяет изменять данные в существующих таблицах.

в) *на добавление записей.* Запрос на добавление добавляет группу записей из одной или нескольких таблиц в конец одной или нескольких таблиц. Например, появилось несколько новых клиентов, а также БД, содержащая сведения о них. Для того чтобы не вводить все данные вручную, их можно добавить в таблицу «Клиенты».

г) *на создание таблицы.* Запрос на создание таблицы создает новую таблицу на основе всех или части данных из одной или нескольких таблиц. Запрос на создание таблицы полезен при создании таблицы для экспорта в другие БД Microsoft Access или при создании архивной таблицы, содержащей старые записи.

3. Перекрестные запросы. Перекрестные запросы используют для расчетов и представления данных в структуре, облегчающей их анализ. Перекрестный запрос подсчитывает сумму, среднее, число значений или

выполняет другие статистические расчеты, после чего результаты группируются в виде таблицы по двум наборам данных, один из которых определяет заголовки столбцов, а другой заголовки строк.

7.7.3. Работа с формами

Формы позволяют пользователям вводить данные в таблицы базы без непосредственного доступа к самим таблицам, также они позволяют выводить результаты работы запросов красиво оформленными. В связи с таким разделением существует два вида формирования структуры форм: на основе таблицы и на основе запроса, хотя возможен и комбинированный подход.

Автоформы. В отличие от таблиц и запросов, формы удобнее готовить с помощью средств автоматизации. Полностью автоматическими являются средства, называемые *автоформами*. Существует три вида автоформ: *в столбец*, *ленточные* и *табличные*.

Автоформа в столбец отображает все поля одной записи – она удобна для ввода и редактирования данных. *Ленточная автоформа* отображает одновременно группу записей – ее удобно использовать для оформления вывода данных. *Табличная автоформа* по внешнему виду ничем не отличается от таблицы, на которой она основана.

Для создания автоформы следует открыть панель *Формы* в окне *База данных* и воспользоваться командной кнопкой *Создать* на панели инструментов этого окна. В открывшемся диалоговом окне *Новая форма* выбирают тип автоформы и таблицу (или запрос), на которой она основывается. После щелчка на кнопке ОК автоформа формируется автоматически и немедленно готова к работе, т.е. к вводу или отображению данных.

Следует обратить внимание на то, что автоформа основывается только на одном объекте. Иные средства создания форм позволяют закладывать в основу структуры формы поля нескольких таблиц или запросов. Если форма основывается только на одном объекте, она называется *простой формой*. Если форма основывается на полях из нескольких связанных таблиц, то она называется *сложной* и представляет собой композицию из нескольких форм.

Создание форм с помощью мастера. Автоматизированные средства предоставляет *Мастер форм* – специальное программное средство, создающее структуру формы в режиме диалога с разработчиком. *Мастер форм* можно запустить из окна *База данных* щелчком на значке *Создание формы с помощью мастера* на панели *Формы* (рис. 7.8):

1. На первом этапе работы *Мастера форм* выбирают таблицы и поля, которые войдут в будущую форму.

2. На втором этапе выбирается внешний вид формы.
3. На третьем этапе выбирается стиль оформления формы.
4. На последнем этапе выполняется сохранение формы под заданным именем. Здесь же можно включить переключатель *Изменить макет формы*, который открывает только что созданную форму в режиме *Конструктора*.

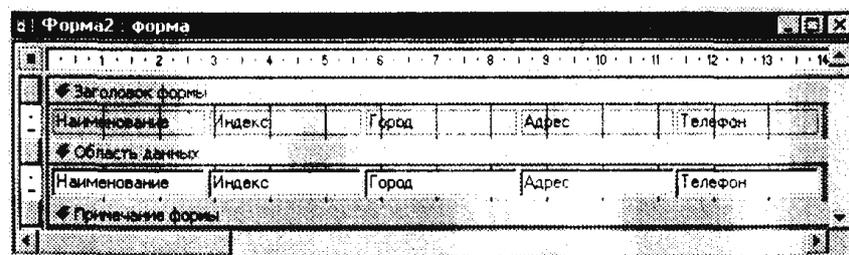


Рис. 7.8. Структура формы, созданной мастером

Разработка форм вручную. Формы можно разрабатывать вручную, открыв пустую форму и добавив в нее объекты (надписи, текстовые поля, графику и т.д.). Для этого нужно открыть панель *Формы* в окне *База данных* и щелкнуть по кнопке *Создать* на панели инструментов, в диалоговом окне *Новая форма* выбирать из списка *Конструктор* и таблицу (или запрос), на которой будет основана форма (если данные находятся в нескольких таблицах, можно выбрать первичную таблицу или запрос, выбирающий поля из всех нужных таблиц), щелкнуть по кнопке *ОК*, после чего на экране отобразится форма в режиме *Конструктора*. Форма имеет три основных раздела: *область заголовка*, *область данных* и *область примечания*. Линии, разделяющие разделы, перетаскиваются по вертикали с помощью мыши. Это позволяет изменять размеры разделов так, как требуется.

Разделы заголовка и примечания имеют чисто оформительское назначение – их содержимое напрямую не связано с таблицей или запросом, на котором основана форма. Раздел данных имеет содержательное значение – в нем представлены элементы управления, с помощью которых выполняется отображение данных или их ввод. Разработчик формы может разместить здесь дополнительные элементы управления для автоматизации ввода данных (переключатели, флажки, списки и другие, типичные для приложений Windows).

Элементы управления формы. Элементы управления, которыми может пользоваться разработчик, представлены на *Панели элементов*. Ее открывают щелчком на соответствующей кнопке панели инструментов Microsoft Access  или командой *Вид→Панель элементов*. Вид и назначение кнопок, расположенных на панели элементов, представлены в таблице 7.2.

Кнопки панели элементов

Кнопка	Назначение
Выбор объектов 	Выбирает, перемещает, изменяет размеры объектов и редактирует их
Мастера 	Кнопка включения/отключения автоматического вызова Мастеров создания элементов управления. Существуют Мастера по созданию группы, поля со списком, списка и командной строки
Надпись 	Добавляет надписи, заголовки, инструкции или описательный текст
Поле 	Используется для отображения, ввода или изменения данных, содержащихся в источнике записей, вывода результатов вычислений, а также приема данных, вводимых пользователем
Группа переключателей 	Служит для создания группы элементов (флажков, переключателей или выключателей), представляющих набор альтернативных значений, из которых выбирается одно значение
Выключатель  Переключатель  Флажок 	Предназначены для отображения логических значений. Выбор (включение) элемента приводит к вводу в соответствующее логическое поле значения «Да», «Истина» или «Вкл» (определяется значением свойства поля <i>Формат поля</i>). Повторный выбор элемента изменяет значение на противоположное: «Нет», «Ложь» или «Выкл»
Поле со списком 	Создает комбинированный список, который позволяет выбирать значение из списка или вводить его вручную
Список 	Предназначен для создания списка возможных значений. Список можно создать, явно вводя данные, либо указав источник данных – таблицу или запрос
Кнопка 	Обычно используется для запуска закрепленного за ней макроса или программы на языке Visual Basic
Рисунок 	Предназначен для размещения в форме неизменяемого рисунка
Свободная рамка объекта 	Используется для размещения объекта из приложения, поддерживающего технологию OLE. В качестве объекта может быть электронная таблица, рисунок, диаграмма, звуковой файл и т.д.
Присоединенная рамка объекта 	Используется для включения в форму OLE-объектов в виде ссылок
Разрыв страницы 	Позволяет вставлять разрыв страницы в многостраничной форме
Вкладка 	Позволяет создать в форме несколько вкладок, каждая из которых может содержать другие элементы управления
Подчиненная форма/отчет 	Предназначен для внедрения в форму некоторой другой (подчиненной) формы
Линия  Прямоугольник 	Предназначены для создания соответствующих геометрических фигур
Другие элементы 	Кнопка, после нажатия которой открывается список всех установленных в системе элементов управления ActiveX

Выбор элемента управления выполняется одним щелчком на его значке в *Панели элементов*, после чего следующим щелчком в поле формы отмечается место, где он должен быть размещен. Вместе с элементом в поле формы вставляется его *присоединенная надпись*. По умолчанию эта надпись стандартная, например, для переключателей это *Переключатель 1*, *Переключатель 2* и т.д. Редактированием свойства элемента управления можно дать элементу управления более содержательную подпись. Доступ к свойствам открывается через контекстное меню соответствующего элемента.

Дизайн формы. В первую очередь все элементы управления форм должны быть аккуратно выровнены. Это обеспечивается командой *Формат*→*Выровнять*. Если нужно равномерно распределить элементы управления по полю формы, используют средства меню *Формат*→*Интервал по горизонтали* или *Формат*→*Интервал по вертикали*.

Возможно ручное изменение размеров и положения элементов управления. При работе вручную используют перетаскивание маркеров, которые видны вокруг элемента управления в тот момент, когда он выделен. Особый статус имеет маркер левого верхнего угла. Перетаскивание с помощью этого маркера позволяет отделить присоединенную надпись от элемента.

Существенную помощь при разработке дизайна формы оказывает вспомогательная сетка. Ее отображение включают командой *Вид*→*Сетка*. Автоматическую привязку элементов к узлам сетки включают командой *Формат*→*Привязать к сетке*.

Управление последовательностью перехода. Физически последовательность перехода – это порядок перехода к следующему полю по окончании работы с предыдущим. Обычно предполагается, что ввод данных в форму должен происходить по элементам управления слева направо и сверху вниз. Последовательность перехода легко проверяется с помощью клавиши Tab. Если при последовательных нажатиях этой клавиши фокус ввода «мечется» по всей форме, значит, последовательность перехода не рациональна, и ее надо править.

Для управления последовательностью перехода служит диалоговое окно *Последовательность перехода*. В нем представлен список элементов управления формы. Порядок элементов в списке соответствует текущему порядку перехода. Изменение порядка перехода выполняется перетаскиванием в два приема:

- щелчком на кнопке маркера слева от названия выделяется элемент управления (кнопка мыши отпускается);
- после повторного щелчка с перетаскиванием элемент перемещается на новое место.

Закончив разработку макета формы, ее следует закрыть и сохранить под заданным именем. После открытия формы в окне *База данных* с ней можно работать: просматривать или редактировать данные из базовой таблицы. Проверку последовательности перехода выполняют клавишей Tab.

7.7.4. Работа с отчетами

Отчеты во многом похожи на формы и страницы доступа к данным, но имеют иное функциональное назначение – они служат для форматированного вывода данных на печатающие устройства и, соответственно, при этом должны учитывать параметры принтера и параметры используемой бумаги. Большая часть того, что было сказано о формах, относится и к отчетам. Здесь также существуют средства автоматического, автоматизированного и ручного проектирования. Средства автоматического проектирования реализованы автоотчетами (*База данных*→*Создать*→*Новый отчет*→*Автоотчет в столбце*). Кроме *автоотчетов в столбце* существуют *ленточные автоотчеты*.

Средством автоматизированного создания отчетов является *Мастер отчетов*. Он запускается двойным щелчком на значке *Создание отчета с помощью мастера* в окне *База данных*. *Мастер отчетов* работает в шесть этапов. При его работе выполняется выбор базовых таблиц или запросов, на которых отчет базируется, выбор полей, отображаемых в отчете, выбор полей группировки, выбор полей и методов сортировки, выбор формы печатного макета и стиля оформления.

Структура готового отчета отличается от структуры формы только увеличенным количеством разделов. Кроме разделов заголовка, примечания и данных, отчет может содержать разделы верхнего и нижнего колонтитулов. Если отчет занимает более одной страницы, эти разделы необходимы для печати служебной информации, например, номеров страниц. Чем больше страниц занимает отчет, тем важнее роль данных, выводимых на печать через эти разделы. Если для каких-то полей отчета применена группировка, количество разделов отчета увеличивается, поскольку оформление заголовков групп выполняется в отдельных разделах.

Редактирование структуры отчета выполняют в режиме *Конструктора* (режим запускается кнопкой *Конструктор* в окне *База данных*). Приемы редактирования те же, что и для форм. Элементы управления в данном случае выполняют функции элементов оформления, поскольку печатный отчет – неинтерактивный объект. Размещение элементов управления выполняют с помощью *Панели элементов* (*Вид*→*Панель элементов*), которая по составу практически не отличается от *Панели элементов*

формы. Важной особенностью отчетов является наличие средства для вставки в область верхнего или нижнего колонтитула текущего номера страницы и полного количества страниц. Эту операцию выполняют с помощью диалогового окна *Номера страниц (Вставка→Номера страниц)*.

7.7.5. Работа со страницами доступа к данным

Страницы (страницы доступа к данным) – новый объект БД, вошедший в Microsoft Access. Как и формы, этот объект служит для обеспечения доступа к данным, содержащимся в базе, но здесь речь идет об удаленном доступе, например о доступе через Интернет или через корпоративную сеть Intranet.

С помощью страниц доступа к данным решается вопрос передачи данных из базы удаленному потребителю. Обычно БД имеют очень большие размеры, и напрямую передавать их через медленные каналы связи непрактично. В то же время, большинство современных Web-браузеров пока не имеют функций для работы с БД, размещенными на серверах. Таким образом, страницы доступа выполняют посредническую функцию. Они имеют небольшой размер, содержат удобные элементы управления для навигации в БД, могут быть записаны в формате кода *HTML*, переданы по медленным каналам связи и воспроизведены в стандартном браузере. В связи с тем, что по формату они являются Web-документами, их нетрудно встроить в любой Web-документ, например, разместить на Web-странице.

От прочих объектов БД страницы доступа отличаются тем, что имеют двойную природу. Прочие объекты базы являются *внутренними*. Так, например, мы не можем выделить ни таблицу, ни запрос, ни форму в виде самостоятельного файла. Эти объекты размещаются где-то внутри файла БД, но операционная система компьютера работать с ними не может, поскольку это не файлы. С ними работает лишь сама СУБД. Страница же представлена двумя объектами: внутренним объектом базы (его можно редактировать) и внешним объектом – файлом в формате *HTML*. Запись этого файла происходит при сохранении спроектированной страницы доступа.

Создание страницы доступа к данным. Для страниц доступа, как и для форм, важную роль играет внешний вид, поэтому создавать их удобно с помощью мастера. *Мастер страниц* запускается щелчком на значке *Создание страницы доступа к данным с помощью мастера*.

1. На первом этапе работы *Мастера форм* выбирают таблицы (или запросы), в их составе – поля, к которым должна обеспечить доступ страница.
2. Второй этап работы мастера предназначен для управления группировкой данных. Эта возможность предусмотрена для доступа к базам,

содержащим большие объемы данных. Если значения в некотором поле часто повторяются, имеет смысл объединить соответствующие им записи в группу. Так, например, для страницы, обеспечивающей доступ к таблице «Новые книги», в отдельные группы могут быть собраны записи, относящиеся к экономической и юридической литературе.

3. В результате группировки образуется иерархическая структура. Она может иметь несколько уровней вложения. Так, например, внутри группы *Юридическая литература* могут быть созданы вложенные группы *Частное право*, *Публичное право* и т.д.

4. Вторая страница мастера предоставляет элементы управления для выбора полей, по которым производится группировка, и управления глубиной уровней группировки. Если просмотреть в режиме *Конструктора* страницу, имеющую уровни группировки, то можно убедиться, что для каждого уровня группировки в структуре объекта образуется отдельный раздел, т.е. различные уровни группировки могут быть дополнены различными элементами управления экранной Web-формы.

5. На третьем этапе выбирается метод упорядочения отображаемых данных. Возможно задание до четырех полей сортировки, причем сортировка возможна как по возрастанию, так и по убыванию.

6. На последнем этапе выполняется сохранение страницы под заданным именем. Здесь же можно перейти в режим *Конструктора*, включив переключатель *Изменить макет страницы*. В случае изменения макета к странице можно применить одну из тем оформления, входящих в состав пакета MS Office. Темы оформления представляют собой совокупности стилей оформления текстов, фоновых узоров и специфических элементов оформления страницы (маркеров, линий и т.д.).

Редактирование страницы доступа к данным. Редактирование созданной страницы доступа выполняется в режиме *Конструктора* теми же приемами, которые были описаны для форм. Основными отличиями являются:

- наличие большего количества разделов (связано с возможностью группировки);
- расширенный состав элементов управления на *Панели элементов (Вид→Панель элементов)*;
- иной механизм перетаскивания элементов управления и присоединенных надписей (элементы управления перетаскиваются вместе с присоединенными надписями, но присоединенные надписи перетаскиваются отдельно от элементов управления).

8. МАТЕМАТИЧЕСКИЙ ПАКЕТ MСAD

8.1. Детали интерфейса

В верхней части окна *Mscad* видны шесть характерных элементов интерфейса:

- строка заголовка – строка с именем системы и текущего документа, а также кнопками управления окном системы;
- строка меню – строка, открывающая доступ к пунктам меню с различными командами;
- панель инструментов – панель с кнопками (значками), обеспечивающими быстрое исполнение наиболее важных команд при работе с системой;
- панель форматирования – панель с кнопками (значками), обеспечивающими быстрое форматирование текстовых и формульных блоков в документах;
- панель вывода палитр математических знаков – панель с кнопками (значками), выводящими палитры специальных математических знаков и греческих букв;
- координатная линейка – линейка с делениями, позволяющая (если это нужно) точно располагать блоки по горизонтали.

Курсор ввода и линия раздела страниц. Обычно в окне редактирования можно увидеть два важных объекта – курсор ввода в виде красного крестика и вертикальную линию, отделяющую текущую страницу от соседней (справа). Положение линии определяется заданными по умолчанию параметрами страницы документа. Курсор ввода устанавливается мышью.

Строка меню. В системе *Mscad_2000* представлены следующие меню:

- *Файл* – работа с файлами, сетью Internet и электронной почтой;
- *Правка* – редактирование документов;
- *Просмотр* – изменение способов представления документа и скрытие/отображение элементов интерфейса;
- *Вставка* – вставка объектов и их шаблонов (включая графику);
- *Форматирование* – изменение формата объектов;
- *Математика* – управление процессом вычислений;
- *Символика* – выбор операций символьного процессора;
- *Окно* – управление окнами системы;
- *Помощь* – работа со справочной БД о системе, центром ресурсов и электронными книгами.

Меню Mscad – контекстные, т.е. число позиций в них и их назначение зависят от состояния системы. Указанные выше меню характерны для рабочего состояния, когда происходит редактирование документа.

Для активизации строки меню без применения мыши достаточно нажать клавишу Alt. Затем, нажимая клавиши перемещения курсора, можно перемещать световое выделение по позициям меню. Выбрав нужное меню, для его раскрытия достаточно нажать клавишу Enter. Можно также нажать клавишу Alt и одновременно клавишу, которая подчеркнута в названии нужного меню.

Панель инструментов содержит несколько групп кнопок управления, каждая из которых дублирует наиболее важные команды меню. При остановке указателя мыши на любой из кнопок появляется подсказка с именем этой кнопки. Рассмотрим назначение кнопок на панели инструментов.

Кнопки операций с файлами:

New (Новый) – создание нового документа с очисткой окна редактирования;

Open (Открыть) – загрузка ранее созданного документа с выбором его файла из диалогового окна;

Save (Сохранить) – запись текущего документа с его текущим именем.

Печать и контроль документов:

Print (Печать) – распечатка документа на принтере;

Print Preview (Предпросмотр) – предварительный просмотр документа;

Check Spelling (Правописание) – проверка орфографии в документе.

Кнопки операций редактирования:

Cut (Вырезать) – перенос выделенной части документа в буфер обмена с очисткой этой части документа;

Copy (Копировать) – перенос выделенной части документа в буфер обмена с сохранением выделенной части документа;

Paste (Вставить) – перенос содержимого буфера обмена в окно редактирования на место, в котором находится курсор ввода;

Undo (Отменить) – отмена предшествующей операции редактирования;

Redo (Восстановить) – повторение ранее отмененной операции.

Кнопки размещения блоков (текстовых, формульных, графических):

Align Across (Привязка через) – блоки выравниваются по горизонтали, располагаясь слева направо в одной строке;



Align Down (Привязать) – блоки выравниваются по вертикали, располагаясь сверху вниз в одном столбце.



Кнопки операций с выражениями:

Insert Function (Вставить функцию) – вставить функцию из списка, появляющегося в диалоговом окне;



Insert Unit (Вставить юнит) – вставка размерных единиц;



Calculate (Подсчет) – вычисление выделенного выражения. Если документы большие, то при их изменениях не всегда выгодно запускать вычисления с самого начала. Эта кнопка позволяет инициировать вычисления только для выделенных блоков, что может сократить время вычислений.



Доступ к дополнительным возможностям Mathcad:

Insert Giperlink (Вставить гиперссылку) – обеспечивает создание гиперссылки;



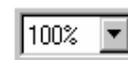
Component Wizard (Вставить компонент) – открывает окно мастера, дающего удобный доступ ко всем компонентам системы;



Run MathConnex (Запуск MathConnex) – запуск системы для симулирования блочно заданных устройств.



Для оперативного изменения масштаба отображения символов в текущем документе на панели инструментов имеется раскрывающийся список *Zoom* (Масштаб).



Кнопки управления ресурсами:

Resource Center (Центр ресурсов) – обеспечивает доступ к центру ресурсов. Центр ресурсов – это мощная БД, объединяющая в себе встроенные в систему электронные книги, обучающую систему, справочную систему, многочисленные примеры применения – «шпаргалки», средства общения с фирмой-разработчиком системы. Кроме того, центр ресурсов предоставляет выход в Internet, средства регистрации и доступа к Web-библиотеке Mathcad и т.д.;



Help (Помощь) – обеспечивает доступ к ресурсам справочной БД системы.



Панель форматирования. Для выбора различных вариантов отображения текстовых блоков и символов служат три раскрывающихся списка:

Style (Стиль) – выбор стиля отображения текстовых блоков;



Font (Шрифт) – выбор шрифта для символов;



Font Size (Размер шрифта) – выбор размера символов.



Для изменения начертания символов предназначены три кнопки:

Bold (Жирный) – полужирное начертание;



Italic (Курсив) – курсивное начертание;



Underline (Подчеркнутый) – подчеркнутое начертание.



Кнопки задания режима выравнивания текста:

Align Left (Привязка влево) – выравнивание текстов по левой границе;



Align Center (Привязка к центру) – выравнивание текстов по центру;



Align Right (Привязка вправо) – выравнивание текстов по правой границе.



Кнопки для создания списков:

Bullets (Маркеры) – создание маркированного списка;



Numbering (Номера) – создание нумерованного списка.



Строка состояния находится в нижней части системы. Она является элементом оперативной помощи по работе с системой, поскольку отображает ее текущее состояние.

Палитры математических знаков служат для вывода шаблонов математических операторов (цифр, знаков арифметических операций, матриц, интегралов, производных и т.д.), функций системы и отдельных символов, например, греческих букв.

8.2. Редактирование с применением клавиатуры

Формы курсора ввода:

– крестообразный красный курсор (+) служит для указания места ввода новых блоков (текстовых, формульных или графических). Курсор имеет такой вид только на пустом месте экрана и может перемещаться клавишами управления курсором или устанавливаться щелчком мыши;

– курсор в виде красной вертикальной черты (курсор ввода) служит для указания на отдельные элементы блоков и обычно используется для

ввода данных и заполнения шаблонов. В текстовых блоках позволяет указать места вставки или удаления отдельных объектов;

– курсор в виде синих уголков разного размера (└ или ┘), выделяющих отдельные части выражения или выражение целиком. Вид курсора зависит от направления ввода.

Клавиши для выделения:

↑ – превращает курсор в выделяющий уголок и расширяет его;

↓ – сужает выделяющий уголок;

→ – перемещает курсор и меняет вид уголка;

← – перемещает курсор и меняет вид уголка;

Shift ↑ – выводит курсор из выражения вверх в свободное поле, делая его крестообразным;

Shift ↓ – выводит курсор из выражения вниз в свободное поле;

Shift → – выводит курсор из выражения вправо в свободное поле

Shift ← – выводит курсор из выражения влево в свободное поле

Пробел – заключает в рамку операнд, действуя как несколько нажатий клавиши ↑, [и выводит курсор из выделенного выражения (двойное нажатие)];

Ins – меняет направление ввода и расположение курсора. Например, если курсор имеет вид ┘, то ввод будет идти вправо.

Управляющие клавиши:

Tab – в тексте перемещает курсор в начало следующего слова, в уравнении управляет выделением частей блока (в частности, выделяя выражения в скобках);

Shift+Tab – в тексте перемещает курсор в начало предыдущего слова, в уравнении управляет выделением частей блока (в частности, выделяя выражения в скобках);

PgUp – перемещает курсор и вызывает прокрутку на пять строк вверх;

PgDn – перемещает курсор и вызывает прокрутку на пять строк вниз;

Ctrl+PgUp – вызывает прокрутку на одно окно вверх;

Ctrl+PgDn – вызывает прокрутку на одно окно вниз;

Home – устанавливает курсор в начало предыдущего блока;

Ctrl+Home – устанавливает курсор в начало документа;

Ctrl+End – устанавливает курсор в конец документа.

Клавиши для создания объектов:

- @ – создание шаблона двумерной графики;
- Ctrl+@ – создание шаблона трехмерной графики;
- Ctrl+G – замена латинской буквы на греческую;
- Ctrl+V – задание шаблона вектора или матрицы;
- Ctrl+P – ввод греческой буквы π ;
- Ctrl+Z – ввод математического символа бесконечности.

8.2.1. Правила ввода текстовых комментариев с помощью текстового редактора

1. Ввести символ " (одна двойная кавычка). В появившемся прямоугольнике можно начинать вводить текст.

2. С помощью раскрывающегося списка выбора шрифта на панели форматирования установить нужный шрифт.

3. Набрать текст.

4. Для завершения ввода текста отвести курсор мыши в сторону от текстового блока и щелкнуть левой кнопкой мыши или нажать Ctrl+Shift+Enter.

Блок имеет маркеры изменения размера в виде маленьких черных прямоугольников, перемещая которые, блок можно увеличивать или уменьшать в том или ином направлении. Выделенные рамкой текстовые блоки можно переносить на другое место, поместив курсор мыши на рамку – он при этом превращается в изображение черной ладошки.

Для коррекции текста надо подвести курсор мыши к месту коррекции и щелкнуть левой кнопкой мыши. Появится рамка текстового блока, а на месте щелчка мышью – курсор ввода.

8.3. Алфавит входного языка системы Mathcad

1) *Алфавит системы Mathcad содержит:*

- малые и большие латинские буквы и греческие буквы;
- арабские цифры от 0 до 9;
- системные переменные;
- математические операторы;
- имена встроенных функций;
- спецзнаки;
- малые и большие буквы кириллицы (при работе с русифицированными документами);
- все, что находится в палитрах математических знаков.

2) Константы:

- целочисленные (0; 1; 23; - 45 и т.д.);
- вещественные числа с мантиссой и порядком ($12.3 \cdot 10^{-5}$ – десятичная константа с мантиссой 12,3 и порядком - 5);
- восьмеричные числа (идентифицируются латинской буквой «O» – от слова «octal» – восьмеричное);
- шестнадцатеричные числа (имеют в конце отличительный признак в виде буквы *h* или *H*, от слова «hexagonal» – шестнадцатеричное; если число начинается с буквы, то перед ней надо ввести ноль);
- комплексные числа $Z = \text{Re}Z + i \cdot \text{Im}Z$, где $\text{Re}Z$ – действительная часть комплексного числа Z , $\text{Im}Z$ – его мнимая часть, а символ i обозначает мнимую единицу, т.е. $\sqrt{-1}$;
- системные константы, хранящие определенные параметры системы;
- строковые константы – любые цепочки символов, заключенные в кавычки. Арифметические выражения в строковых константах рассматриваются как текст и не вычисляются;
- единицы измерения физических величин.

3) Переменные

Тип переменной определяется ее значением. Переменные могут быть числовыми, строковыми, символьными и т.д., поэтому тип переменной предварительно не задается. Каждой переменной должно быть присвоено значение (т.е. переменная должна быть определена). Попытка использования неопределенной переменной ведет к выводу сообщения об ошибке.

Константы, переменные и иные объекты имеют имена (идентификаторы). Имена могут иметь практически любую длину, и в них могут входить любые латинские и греческие буквы, а также цифры и некоторые спецсимволы (например, `_`), но начинаться имя может только с буквы. Нельзя использовать в именах буквы русского языка. Малые и большие буквы различаются. Имена не могут совпадать с именами встроенных или определенных пользователем функций.

4) Операторы

Это элементы языка, предназначенные для создания математических выражений совместно с данными, именуемыми операндами. Это знаки арифметических операций, вычисления сумм, произведений, производной, интеграла и т.д.

5) Встроенные функции

Это функции определенные в самой системе и готовые к немедленному использованию (например, $\sin(x)$, $\text{atan}(x)$, $\ln(x)$ и др.).

б) *Функции пользователя*

Это те функции, которые создаются самим пользователем. Благодаря этим функциям обеспечивается расширение входного языка Mathcad и его адаптация к специфическим задачам пользователя.

8.3.1. Ввод математических выражений и работа с формульным редактором

Операции вывода и присваивания

Для вычисления любого выражения достаточно установить после него оператор вывода (знак $=$). Оператор $=$ можно использовать только для первого присваивания.

Для присвоения переменным новых значений используется оператор $:=$, который вводится своим первым символом – двоеточием ($:$).

Существует также «жирный» знак равенства, который используется в логических операциях сравнения.

Использование оператора $:=$ обеспечивает локальное присваивание. С помощью знака \equiv можно выполнить глобальное присваивание, т.е. независимо от того, в каком месте документа расположен оператор глобального присваивания, переменная получает это значение.

Оператор умножения вводится звездочкой ($*$), но представляется точкой в середине строки.

Оператор деления вводится как косая черта ($/$), но заменяется горизонтальной чертой.

Оператор возведения в степень вводится знаком \wedge , но число в степени представляется в обычном виде.

Для ввода десятичных чисел в качестве разделителя целой и дробной части используется точка.

По умолчанию десятичные числа имеют представление с тремя знаками после разделительной точки.

Mathcad идентифицирует наиболее распространенные константы, например, e – основание натурального логарифма.

8.4. Простые вычисления арифметических выражений и их редактирование

Рассмотрим пример на вычисление отношения суммы $2 + 3$ к корню квадратному из числа 5.

Вначале введем подряд символы $2 + 3$. Получим

$2 + 3$

Введем знак деления. Для этого нажатием клавиши Пробел выделим все выражение $2 + 3$, а затем нажмем клавишу со знаком $/$. Получим

$$\frac{2 + 3}{}$$

Введем знак квадратного корня. Для этого нажмем клавишу со знаком $\sqrt{\quad}$. Получим

$$\frac{2 + 3}{\sqrt{\quad}}$$

Введем подкоренное выражение. Для этого нажмем клавишу с цифрой 5. Получим

$$\frac{2 + 3}{\sqrt{5}}$$

Выведем результат вычисления. С помощью пробела выделим все выражение,

$$\frac{2 + 3}{\sqrt{5}}$$

а затем в конце выражения поставим оператор вывода $=$. Получим

$$\frac{2 + 3}{\sqrt{5}} = 2.236$$

Рассмотрим случай, когда нужно изменить подкоренное выражение. Пусть подкоренное выражение будет число 5 в степени 1,25.

Сначала аккуратно поместим курсор мыши после числа 5 и щелкнем левой кнопкой. Получим

$$\frac{2 + 3}{\sqrt{5}} = 2.236$$

Поставим знак возведения в степень. Для этого на клавиатуре нажмем клавишу со значком \wedge . Получим

$$\frac{2 + 3}{\sqrt{5}^{\quad}} = \dots$$

Введем показатель степени числа 5, т.е. число 1,25. Получим

$$\frac{2 + 3}{\sqrt{5^{1.25}}} = \dots$$

Отведем курсор мыши от формульного блока на свободное поле документа и щелкнем левой кнопкой.

$$\frac{2 + 3}{\sqrt{5^{1.25}}} = 1.829$$

Получим

Задача 1. Простые вычисления в Mathcad.

Функция 1 $f1(x) := |\sin(x)| \cdot \exp(x)$.

Функция 2 $f2(x) := 10 - x^{1.9}$.

Интервал аргумента $a := 0.5$, $b := 4$.

Число точек и шаг $N := 40$ $h := \frac{b - a}{N}$.

Цикл аргумента $n := 0 \dots N$.

Задача 2.

Расчет функции 1

$$y_n := f1(a + nh)$$

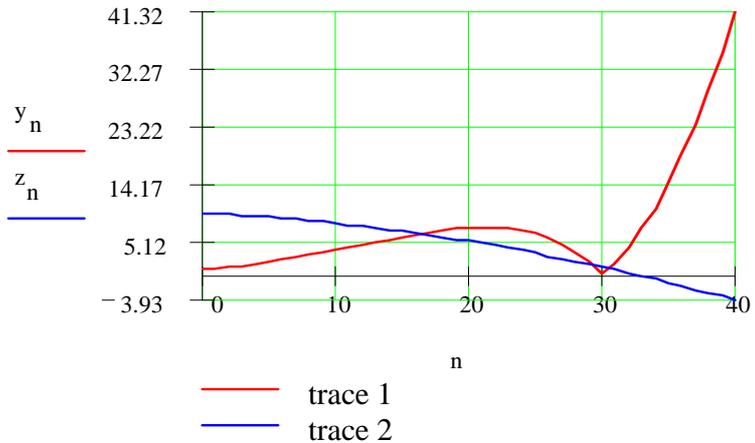
	0
0	0.79
1	0.997
2	1.227
3	1.481
4	1.758
5	2.058
6	2.382
7	2.728
8	3.094
9	3.479
10	3.88
11	4.291

Расчет функции 2

$$z_n := f2(a + nh)$$

	0	1	2	3	4	5	6
0	9.732	9.636	9.526	9.403	9.266	9.115	8.952

Задача 3.
Графики



Ранжированные переменные имеют множественные значения. Если требуется задать ряд чисел с шагом, то ранжированная переменная записывается следующим образом:

$$X := X_1, X_2 .. X_{end},$$

где X_1, X_2, X_{end} – начальное, второе, конечное значение переменной X , шаг вычисляется системой как $X_2 - X_1$. Если значение X_2 не указано, то шаг принимается равным 1.

Пример 1.

$$i := 1..5 \quad x := 2, 1.5..0 \quad z := -0.5, -0.2..0.5$$

i =
1
2
3
4
5

x =
2
1.5
1
0.5
0

z =
-0.5
-0.25
0
0.25
0.5

ORIGIN := 1

$$f_i := i^2 \quad f = \begin{pmatrix} 1 \\ 4 \\ 9 \\ 16 \\ 25 \end{pmatrix} \quad \begin{matrix} f_2 = 4 \\ f_3 = 9 \end{matrix}$$

8.5. Работа с графикой

1. Построение двумерного графика одной функции:

- ввести функцию, набрав выражение $\sin(x)^3$;
- на панели математических знаков щелкнуть на кнопке ;
- на экране появится палитра графиков;
 - в палитре графиков щелкнуть на кнопке ;
 - на экране появится шаблон графика;
 - на оси X введите имя аргумента – x ;
 - на оси Y введите $\sin(x)^3$;
 - для завершения построения графика щелкнуть левой кнопкой мыши вне пределов графика (рис. 8.1).

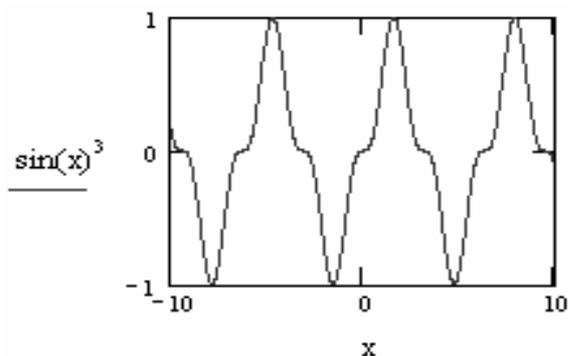


Рис. 8.1

2. Изменение размеров и перемещение графика:

- поместить курсор мыши в область графика и щелкнуть левой кнопкой (активизировать график) – вокруг графика появится черная рамка;
- подвести курсор мыши к маркеру изменения размера в правом нижнем углу рамки и, нажав левую кнопку, растянуть график по диагонали;
- навести курсор мыши на любую сторону рамки (кроме маркеров изменения размера), при этом курсор должен превратиться в черную ладошку;
- нажав левую кнопку мыши, передвинуть весь блок графика в желаемую область экрана. В результате этих действий получим увеличенный и перемещенный в другую область экрана график с обрамляющей его рамкой (см. рис. 8.1).

3. Построение графиков нескольких функций

Для примера рассмотрим построение графиков функций $\sin(x)^2$ и $\cos(x)$:

- щелкнуть левой кнопкой мыши точно в конце выражения $\sin(x)^3$;
- нажатием клавиши *Пробел* добиться, чтобы все выражение $\sin(x)^3$ было охвачено синим уголком, и поставить запятую;
- ввести выражение $\sin(x)^2$;
- нажатием клавиши *Пробел* добиться, чтобы все выражение $\sin(x)^2$ было охвачено синим уголком, и поставить запятую;
- ввести выражение $\cos(x)$;
- для завершения построения графика щелкнуть левой кнопкой мыши вне пределов графика (рис. 8.2).

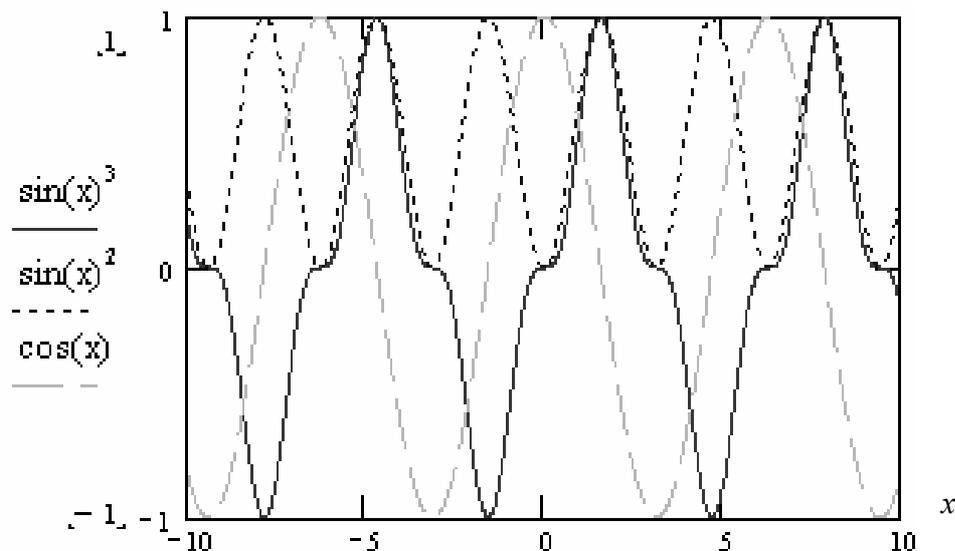


Рис. 8.2

4. Простейшие приемы форматирования графиков:

- графики можно перемещать по полю окна документа и изменять их размеры;
- изменять различные параметры графиков можно при помощи окна форматирования, которое появляется, если дважды щелкнуть мышью на графике, либо щелкнуть один раз, если график выделен;
- форматирование можно также осуществлять с помощью контекстного меню, которое появляется при щелчке на графике правой кнопкой мыши.

5. Построение графиков поверхностей

Пусть необходимо построить график функции $z(x, y) = x^2 + y^2$, для чего необходимо:

- определить функцию $z(x, y)$;
- ввести шаблон трехмерного графика ;
- на единственное поле ввода под шаблоном ввести z ;
- для завершения построения графика щелкнуть левой кнопкой мыши вне пределов графика (рис. 8.3).

6. Вращение трехмерного графика:

- построенную поверхность поместить с помощью контекстного меню графика в призму (*box*) (рис. 8.4);
- поместить курсор мыши в область графика, нажать левую кнопку и, удерживая ее, двигать мышью в том или ином направлении. Фигура вместе с осями координат и призмой, в которой она находится, будет вращаться.

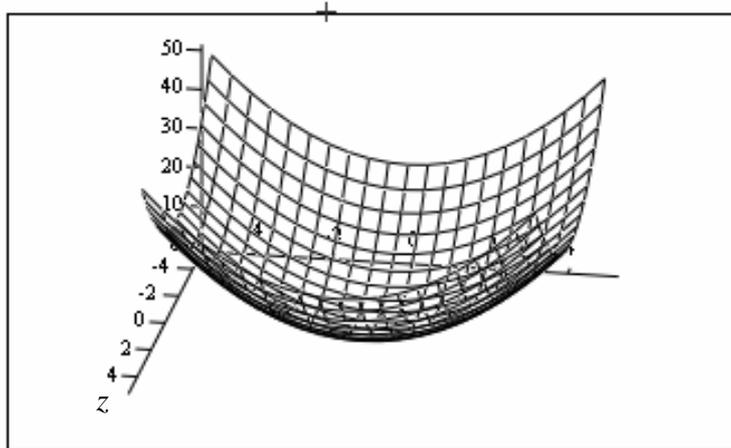


Рис. 8.3

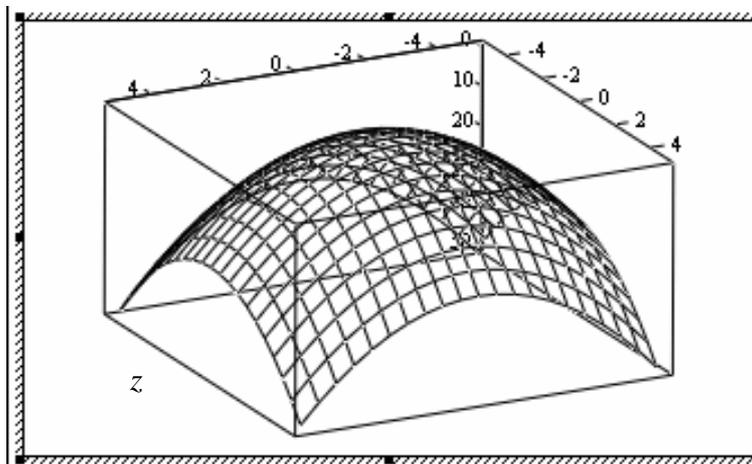


Рис. 8.4

Если при вращении фигуры удерживать нажатой клавишу `Ctrl`, можно удалять или приближать объект к наблюдателю. Если при вращении фигуры удерживать нажатой клавишу `Shift`, то после отпущения левой кнопки можно наблюдать анимированную картину вращения объекта в заданном предварительно направлении. Для остановки вращения надо щелкнуть левой кнопкой мыши.

7. Построение нескольких поверхностей на одном графике

Необходимо построить график двух трехмерных поверхностей. Пусть заданы две функции двух переменных $z_1(x,y)$ и $z_2(x,y)$:

$$z_1(x,y) := x^2 + y^2 \quad z_2(x,y) := -(x^2 + y^2).$$

- определить ряд необходимых функций, описывающих поверхности;
- ввести через запятую имена этих функций в поле ввода шаблона трехмерного графика;
- для завершения построения графика щелкнуть левой кнопкой мыши вне пределов графика (рис. 8.5).

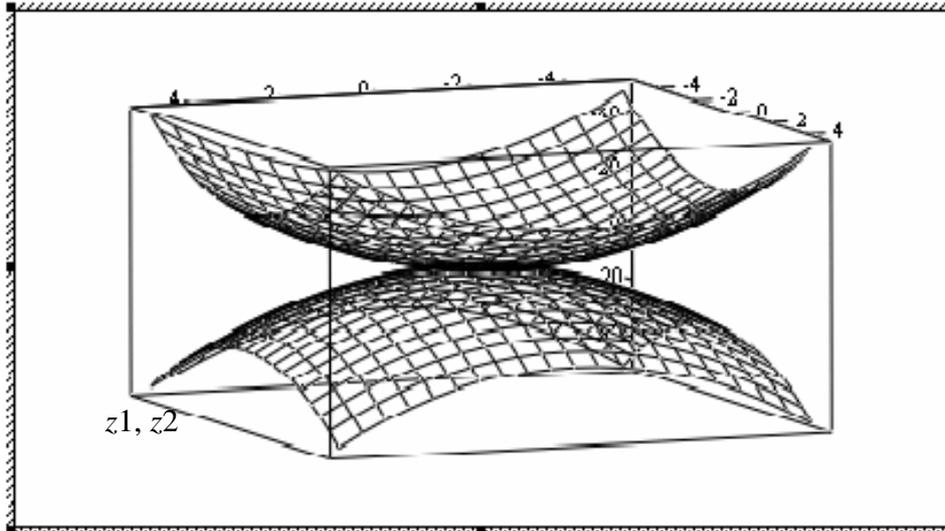


Рис. 8.5

Задача 1. Многомерные вычисления в Mathcad.

$$\text{Функция } y(x,t) := \text{if} \left(x = 0, 1, \frac{\sin(x)}{x} \right) \cdot \text{if} \left(t = 0, 1, \frac{\sin(t)}{t} \right)$$

Данные для циклов $N := 20$ $M := 20$ $h := \pi/5$.

Изменения аргумента $n := 0 \dots N$.

Изменения аргумента $m := 0 \dots M$.

Задача 2. Расчет функции

$$M_{n,m} := y \left[\left(n - \frac{N}{2} \right) \cdot h, \left(m - \frac{N}{2} \right) \cdot h \right]$$

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0.011	0.02	0.022	0.016	0	-0.024	-0.052
2	0	0.02	0.036	0.041	0.03	0	-0.044	-0.095
3	0	0.022	0.041	0.047	0.034	0	-0.051	-0.109
4	0	0.016	0.03	0.034	0.024	0	-0.036	-0.079
5	0	0	0	0	0	0	0	0
6	0	-0.024	-0.044	-0.051	-0.036	0	0.055	0.118
7	0	-0.052	-0.095	-0.109	-0.079	0	0.118	0.255
8	0	-0.079	-0.143	-0.164	-0.118	0	0.177	0.382
9	0	-0.097	-0.177	-0.202	-0.146	0	0.219	0.472
10	0	-0.104	-0.189	-0.216	-0.156	0	0.234	0.505
11	0	-0.097	-0.177	-0.202	-0.146	0	0.219	0.472
12	0	-0.079	-0.143	-0.164	-0.118	0	0.177	0.382
13	0	-0.052	-0.095	-0.109	-0.079	0	0.118	0.255

Задача 3.

Поверхностный график (рис. 8.6).

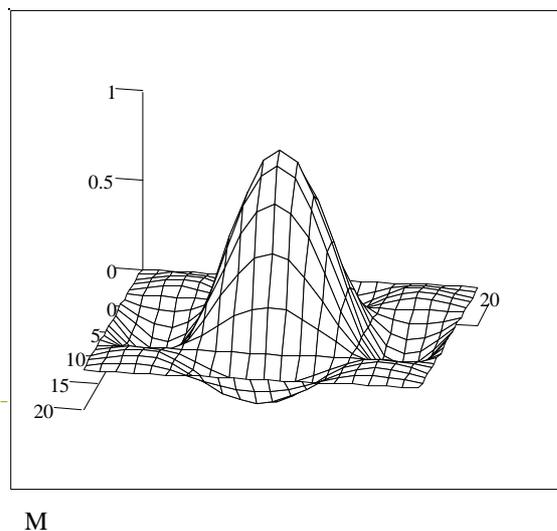


Рис. 8.6

Задача 4.

Контурный график (рис. 8.7).

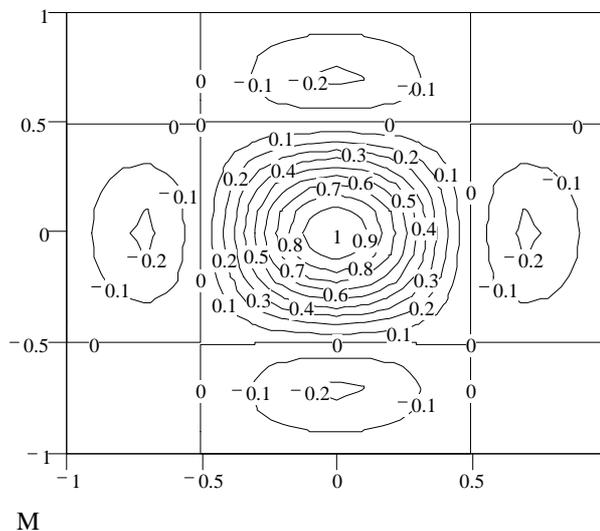


Рис. 8.7

8.6. Работа с векторами и матрицами

1. Создание матриц и векторов

Команда *Matrices* (матрицы) меню *Insert* (Вставка) обеспечивает вывод шаблона для вектора или матрицы. Шаблон содержит поля ввода элементов вектора или матрицы. Поля ввода можно активизировать щелчком мыши. С помощью клавиш перемещения курсора можно ввести все элементы вектора или матрицы.

Каждый элемент матрицы характеризуется индексированной переменной, а его положение в матрице определяется двумя индексами – номером строки и номером столбца. Для задания индексированной переменной прежде всего надо ввести имя переменной, а затем перейти к набору индексов нажатием клавиши `[]`. Сначала указывается индекс строки, а затем индекс столбца.

Вырожденная в одну строку или в один столбец матрица является вектором. Его элементы – индексированные переменные с одним индексом.

Нижняя граница индексов задается значением системной переменной `ORIGIN`. Обычно ее значение задают равным 0 или 1.

2. Векторные и матричные операторы

Примем следующие обозначения: V – вектор, M – матрица, Z – скалярная величина.

Оператор	Клавиши	Описание
$V1+V2$	$V1+V2$	Сложение двух векторов $V1$ и $V2$
$V1-V2$	$V1-V2$	Вычитание двух векторов $V1$ и $V2$
$-V$	$-V$	Смена знака у элементов вектора V
$-M$	$-M$	Смена знака у элементов матрицы M
$V \cdot Z$	$V \cdot Z$	Вычитание из всех элементов вектора V скаляра Z
$Z \cdot V, V \cdot Z$	$Z * V, V * Z$	Умножение вектора V на скаляр Z
$Z \cdot M, M \cdot Z$	$Z * M, M * Z$	Умножение матрицы M на скаляр Z
$V1 \cdot V2$	$V1 * V2$	Скалярное умножение двух векторов
$M \cdot V$	$M * V$	Умножение матрицы на вектор
$M1 \cdot M2$	$M1 * M2$	Умножение двух матриц
V/Z	V/Z	Деление всех элементов вектора V на скаляр Z
M/Z	M/Z	Деление матрицы M на скаляр Z
M^{-1}	M^{-1}	Обращение матрицы M
M^n	M^n	Возведение матрицы M в степень n
$ V $	$ V $	Вычисление модуля вектора V
$ M $	$ M $	Вычисление определителя матрицы M
V^T	$V \text{ Ctrl !}$	Транспонирование вектора V
M^T	$M \text{ Ctrl !}$	Транспонирование матрицы M
$V1 \times V2$	$V1 \text{ Ctrl } * V2$	Векторное умножение двух векторов
ΣV	$\text{Alt } \$ V$	Вычисление суммы элементов вектора V
\vec{V}	$V \text{ Ctrl } \text{---}$	Векторизация вектора V
\vec{M}	$M \text{ Ctrl } \text{---}$	Векторизация матрицы M
$M^{<n>}$	$M \text{ Ctrl } ^ n$	Выделение n -ного столбца матрицы
V_n	$V [n$	Выделение n -ного элемента вектора
$M_{m, n}$	$M [(m, n)$	Выделение элемента (m, n) матрицы

Все представленные в таблице операторы могут вызываться из палитры матричных операций.

8.7. Решение уравнений

8.7.1. Решение одного уравнения

Решаемое уравнение $x^3 - e^3 = 0$.

Начальное значение переменной $x := 3$.

$$z := \text{root}(x^3 - e^x, x).$$

Найденное решение $z = 1.857$.

8.7.2. Нахождение корней полинома

Полином $x^3 - 10x + 2 = 0$.

Вектор коэффициентов, начинающийся с константы. Сюда должны войти все коэффициенты, в том числе равные нулю.

$$v := \begin{pmatrix} 2 \\ -10 \\ 0 \\ 1 \end{pmatrix}.$$

$$\text{Найденное решение } \text{polyroots}(v) = \begin{pmatrix} -3.258 \\ 0.201 \\ 3.057 \end{pmatrix}.$$

8.7.3. Решение систем линейных уравнений

1) метод обратной матрицы

Решить систему уравнений

$$\begin{cases} 2 \cdot x_1 + 7 \cdot x_2 + 3 \cdot x_3 = 6 \\ 3 \cdot x_1 + 5 \cdot x_2 + 2 \cdot x_3 = 4 \\ 9 \cdot x_1 + 4 \cdot x_2 + 10 \cdot x_3 = 2 \end{cases}$$

Матрица коэффициентов системы линейных уравнений

$$A := \begin{pmatrix} 2 & 7 & 3 \\ 3 & 5 & 2 \\ 9 & 4 & 10 \end{pmatrix}.$$

$$\text{Вектор свободных членов } B := \begin{pmatrix} 6 \\ 4 \\ 2 \end{pmatrix}.$$

Решение системы $X := A^{-1} \cdot B$.

Результаты решения $X = \begin{pmatrix} -1.182 \\ 0.909 \\ 0 \end{pmatrix}$.

2) с помощью функций *Given* и *Find*

Задание начальных условий: $x1 := 1, x2 := 1, x3 := 1$.

Ввод функции *Given* $2 \cdot x1 + 7 \cdot x2 + 3 \cdot x3 = 6$

Ввод системы уравнений $3 \cdot x1 + 5 \cdot x2 + 2 \cdot x3 = 4$

$9 \cdot x1 + 4 \cdot x2 + 10 \cdot x3 = 2$

Получение результата с помощью функции *Find*

$$\text{Find}(x1, x2, x3) = \begin{pmatrix} -1.182 \\ 0.909 \\ 0 \end{pmatrix}.$$

8.7.4. Решение нелинейных уравнений

1) с помощью функций *Given* и *Find*

Задание начальных условий: $x := 1$.

Ввод функции *Given* Given

Ввод уравнения $x^3 + 0.2 \cdot x^2 - 1.2 = 0$

Получение результата с помощью функции $z := \text{Find}(x)$.

$z = 1$.

2) приближенное решение нелинейного уравнения с помощью функций *Given* и *Minerr*

Задание начальных условий: $x := 1$.

Ввод функции *Given* Given

Ввод уравнения $x^3 + 0.2 \cdot x^2 - 1.2 = 0$ $z := \text{Minerr}(x)$.

Получение результата с помощью функции *Minerr* $z = 1$.

8.7.5. Приближенное решение системы нелинейных уравнений

Начальные значения переменных: $x := 0$ $y := 0$.

Начало вычислительного блока Given

$$(x^2 + 1)^2 + (y^2 + 1)^2 = 5.5$$

Решаемая система уравнений $x + y = 0.95$

$z := \text{Minerr}(x, y)$

Найденное решение $z = \begin{pmatrix} -0.106 \\ 1.056 \end{pmatrix}$.

8.8. Выполнение символьных вычислений

Символьными называются такие вычисления, результаты которых представляются в аналитическом виде, т.е. в виде формул. В частном случае результат может быть числом. Команды, относящиеся к работе символьного процессора, содержатся в меню *Symbolics* (*Символика*). Чтобы выполнить символьную операцию, надо выделить то выражение, над которым она производится. Для ряда операций следует не только указать выражение, к которому относится операция, но и наметить переменную, относительно которой выполняется символьная операция. Для выполнения символьной операции нужно выбрать меню *Символика*→*Величина*→*Символически* (*Symbolics*→*Evaluate*→*Symbolically*).

Символьные операции:

1. Опция *Evaluate/Symbolically* – вычислить символически.
2. Опция *Evaluate/Floating Point* – вычислить с плавающей точкой.
3. Опция *Evaluate/Complex* – вычислить в комплексной форме.
4. Опция *Simplify* – упростить.
5. Опция *Expand* – развернуть (разложить по степеням).
6. Опция *Factor* – факторизовать (разложить на множители).
7. Опция *Collect* – группировать по подвыражению.
8. Опция *Polynomial Coefficients* – полиномиальные коэффициенты.
9. Опция *Variable/Solve* – решить для переменной.
10. Опция *Variable/Substitute* – подстановка подвыражения переменной.
11. Опция *Variable/Differentiate* – дифференцировать по переменной.
12. Опция *Variable/Integrate* – интегрировать по переменной.
13. Опция *Expand to Series* – разложить в ряд Тейлора.
14. Опция *Convert to Partial Fraction* – преобразовать в элементарные дроби.
15. Операции с матрицами (транспонирование, обращение, определитель).
16. Преобразования Фурье, Лапласа, z-.
17. Опция *Evaluation Style* – стиль эволюции.
18. Система *SmartMath* – чудо-математика.

Рассмотрим примеры вычисления интеграла, дифференциала и предела:

Исходное выражение	Результат операции
$\frac{d}{dx} \sin(x)$	$\cos(x)$
$\int_a^b x^2 dx$	$\frac{1}{3}b^3 - \frac{1}{3}a^3$
$\lim_{x \rightarrow 0} (1+x)^{1/x}$	$\exp(1)$

Развертка / свертка

Пример 1.

Функция $(a + x)^3$.

Ее развертка $a^3 + 3a^2x + 3ax^2 + x^3$.

Ее развертка в *SmartMath* (значение a использовано)

$$(a + x)^3 \text{ expand} \rightarrow 8 + 12x + 6x^2 + x^3.$$

Пример 2.

Функция $a^3 + 3a^2x + 3ax^2 + x^3$

Ее свертка $(x + a)^3$.

Разложение в ряд Тейлора

Функция $\frac{\sin(x)}{x}$.

Ее разложение в ряд по x $1 - \frac{1}{6}x^2 + \frac{1}{120}x^4 + O(x^5)$.

Отброшен остаточный член $F2(x) := 1 - \frac{1}{6}x^2 + \frac{1}{120}x^4$.

8.9. Нахождение экстремумов функций

Для нахождения экстремумов функций многих переменных существует две альтернативные возможности. Первая заключается в использовании блока *given* и функции *minerr*. Определим функцию двух переменных

$$G(x, y) := 25 - x^2 - y^2.$$

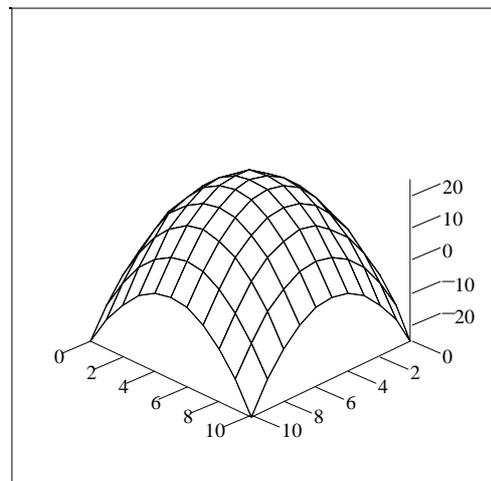
Зададимся целью найти ее экстремум в области $x = [-5, 5]$ $y = [-5, 5]$. Оценим по графику положение экстремума. Заносим в матрицу **M** значения функции в узловых точках:

$$i := 0 \dots 10, \quad j := 0 \dots 10,$$

$$M_{(i,j)} := G(i - 5, j - 5).$$

На заданном интервале функция не превосходит 25 (рис. 8.8). Зададим начальные приближения для поиска экстремума:

$$x := 1, \quad y := 1.$$



M

Рис. 8.8

Записываем блок уравнений или неравенств. Число уравнений и неравенств в блоке *given – Find* должно быть больше и равно числа иско- мых величин. Если уравнений и неравенств не хватает, то можно просто продублировать одно и то же уравнение или вписать какое-либо тождест- во, например, $2 = 2$.

$$\text{Given } G(x,y)=26 \quad x \geq -5 \quad x \leq 5 \quad y \geq -5 \quad y \leq 5$$

$$\text{Minerr}(x, y) = \begin{pmatrix} 4.584 \cdot 10^{-6} \\ 4.02 \cdot 10^{-5} \end{pmatrix}.$$

Функция *Minerr* ищет приближенной решение для системы уравне- ний и неравенств, записанных в блоке. В данном случае мы получили, что системе уравнений наилучшим образом соответствует точка $[0,0]$. (По- скольку по умолчанию точность вычислений составляет 0.001, мы округ- лили результат до 0).

Из графика видно, что значение 26 больше самого большого значе- ния функции в окрестностях точки $[1,1]$, т.е. точное решение найти нельзя и функция *Minerr* подбирает такое значение x , при котором функция бли- же всего к значению 26.

Вторая возможность – поиск нулей первой производной, т.е. стан- дартный математический подход. Для этого можно использовать блок *given – Find*. Функция *Find* ищет точное решение системы уравнений и неравенств, записанных после слова *given*

$$\text{Given } \frac{d}{dx}G(x,y) = 0 \quad \frac{d}{dy}G(x,y) = 0 \quad \text{Find}(x, y) = \begin{pmatrix} 1.059 \cdot 10^{-7} \\ 1.059 \cdot 10^{-7} \end{pmatrix}.$$

Результаты, полученные различными методами, совпадают, время счета мало в обоих случаях.

Дополнительная возможность поиска экстремумов с помощью функ- ций *Minimize* и *Maximize*, которые могут быть использованы как сами по себе, так и совместно с блоком *Given*. Аргументы функций: имя функции, экстремум которой ищем, и список ее аргументов.

Определяем функцию двух переменных $f(x,y) := 25 - x^2 - y^2$ и задаем начальные приближения $x := 1, y := 1$.

Задаем область поиска максимума внутри блока **Given**:

$$\text{Given } 0 \leq x \leq 5 \quad 0 \leq y \leq 5.$$

Находим максимум функции в заданной области:

$$P := \text{Maximize}(f, x, y) \quad P = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

В случае функции одной переменной задаем функцию $g(x) := x^4 - x^2$ и начальное приближение $x := 1$.

Находим максимум $\text{Maximize}(g, x) = 0$.

Для нахождения максимального или минимального значения функции на некотором интервале, то необходимо определить этот интервал в блоке **Given**:

$$\text{Given } 0 < x \leq 2 \quad \text{Maximize}(g, x) = 2$$

На приведенном графике (рис. 8.9) видно, что первый из найденных максимумов соответствовал случаю, когда производная обращается в ноль; второй максимум лежит на границе интервала $x := -2, -1.9..2$.

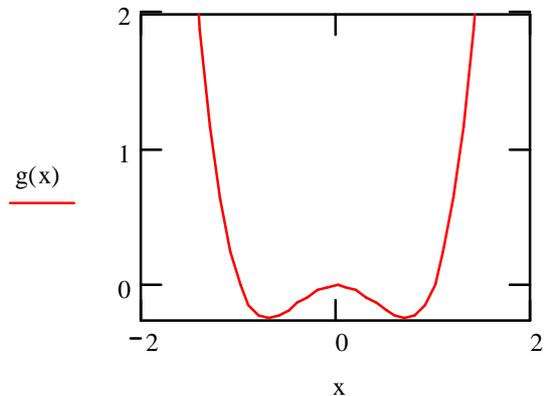


Рис. 8.9

8.10. Аппроксимация функций

Аппроксимацией (приближением) функции $f(x)$ называется нахождение такой функции $g(x)$ (*аппроксимирующей функции*), которая была бы близка заданной. Критерии близости функций $f(x)$ и $g(x)$ могут быть различные. В том случае, когда приближение строится на дискретном наборе точек, аппроксимацию называют *точечной* или *дискретной*.

В том случае, когда аппроксимация проводится на непрерывном множестве точек (отрезке), она называется *непрерывной* или *интегральной*. Примером такой аппроксимации может служить разложение функции в ряд Тейлора, т.е. замена некоторой функции степенным многочленом.

Наиболее часто встречающимся видом точечной аппроксимации является *интерполяция* (в широком смысле).

Пусть задан дискретный набор точек x_i ($i = 0, 1, \dots, n$), называемых узлами интерполяции, причем среди этих точек нет совпадающих, а также значения функции y_i в этих точках. Требуется построить функцию $g(x)$, проходящую через все заданные узлы. Таким образом, критерием близости функции является $g(x_i) = y_i$.

В качестве функции $g(x)$ обычно выбирается полином, который называют *интерполяционным полиномом*.

В том случае, когда полином един для всей области интерполяции, говорят, что интерполяция *глобальная*.

В тех случаях, когда между различными узлами полиномы различны, говорят о *кусочной* или *локальной интерполяции*.

Найдя интерполяционный полином, можно вычислить значения функции $f(x)$ между узлами (провести *интерполяцию в узком смысле слова*), а также определить значение функции $f(x)$ даже за пределами заданного интервала (провести *экстраполяцию*). Следует иметь в виду, что точность экстраполяции обычно очень невелика.

8.10.1. Локальная интерполяция. Линейная интерполяция

При линейной интерполяции Mathcad соединяет существующие точки данных прямыми линиями. Это выполняют функцией *linterp*, описанной ниже.

Функция *linterp*(v_x, v_y, x) – использует векторы данных VX и VY , чтобы возвратить линейно интерполируемое значение y , соответствующее третьему аргументу x . Аргументы VX и VY должны быть векторами одинаковой длины. Вектор VX должен содержать вещественные значения, расположенные в порядке возрастания.

Эта функция соединяет точки данных отрезками, создавая таким образом ломаную. Интерполируемое значение для конкретного x есть ордината y соответствующей точки ломаной. Для значений x , расположенных перед первой точкой в VX , Mathcad продолжает ломаную прямой линией, проходящей через первые две точки данных. Для значений x , расположенных за последней точкой в VX , Mathcad продолжает ломаную прямой линией, проходящей через последние две точки данных.

Для получения наилучших результатов значение x должно находиться между самым большим и самым маленьким значениями в векторе VX .

Пример 1. Функция задана таблично. Найти значения этой функции при указанных значениях аргумента x (рис. 8.10): $x = 0,512$; $x = 0,535$.

$$VX := \begin{pmatrix} 0.51 \\ 0.52 \\ 0.53 \\ 0.54 \\ 0.55 \\ 0.56 \\ 0.57 \end{pmatrix} \quad VY := \begin{pmatrix} 1.6651 \\ 1.6820 \\ 1.6989 \\ 1.7160 \\ 1.7333 \\ 1.7507 \\ 1.7683 \end{pmatrix}$$

$$\text{linterp}(VX, VY, 0, 512) = 1,668 \quad \text{linterp}(VX, VY, 0, 535) = 1,707$$

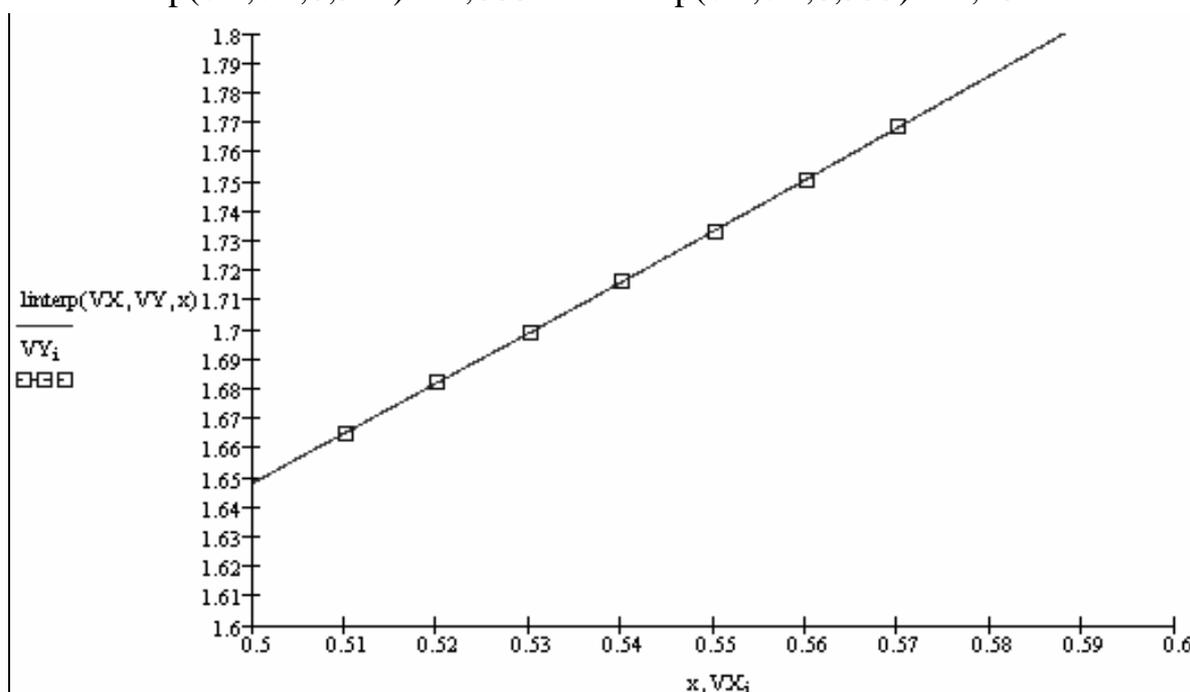


Рис. 8.10

8.10.2. Кубическая сплайн-интерполяция

Кубическая сплайн-интерполяция позволяет провести кривую через набор точек таким образом, что первые и вторые производные кривой непрерывны в каждой точке. Эта кривая образуется путем создания ряда кубических полиномов, проходящих через наборы из трех смежных точек. Кубические полиномы потом состыковываются друг с другом, чтобы образовать одну кривую.

Для того чтобы провести кубический сплайн через набор точек, нужно:

- создать векторы VX и VY , содержащие координаты x и y , через которые нужно провести кубический сплайн. Элементы VX должны быть расположены в порядке возрастания;

- вычислить вектор $VS := cspline(VX, VY)$; вектор VS содержит вторые производные интерполяционной кривой в рассматриваемых точках;
- чтобы найти интерполируемое значение в произвольной точке, скажем $X0$, вычислить $interp(VS, VX, VY, X0)$, где VS, VX, VY – векторы, описанные ранее.

Можно сделать то же самое, вычисляя $interp(cspline(VX, VY), VX, VY, X0)$.

Пример 1. Функция задана таблично. Найти значения этой функции при указанных значениях аргумента x (рис. 8.11): $x = 0,512$; $x = 0,535$.

$$\begin{array}{l}
 \left(\begin{array}{c} 0.51 \\ 0.52 \\ 0.53 \\ 0.54 \\ 0.55 \\ 0.56 \\ 0.57 \end{array} \right) \\
 VX :=
 \end{array}
 \quad
 \begin{array}{l}
 \left(\begin{array}{c} 1.6651 \\ 1.6820 \\ 1.6989 \\ 1.7160 \\ 1.7333 \\ 1.7507 \\ 1.7683 \end{array} \right) \\
 VY :=
 \end{array}
 \quad
 VS := cspline(VX, VY)$$

+

$$\begin{array}{l}
 interp(VS, VX, VY, 0,512) = 1,668 \\
 interp(VS, VX, VY, 0,535) = 1,707
 \end{array}$$

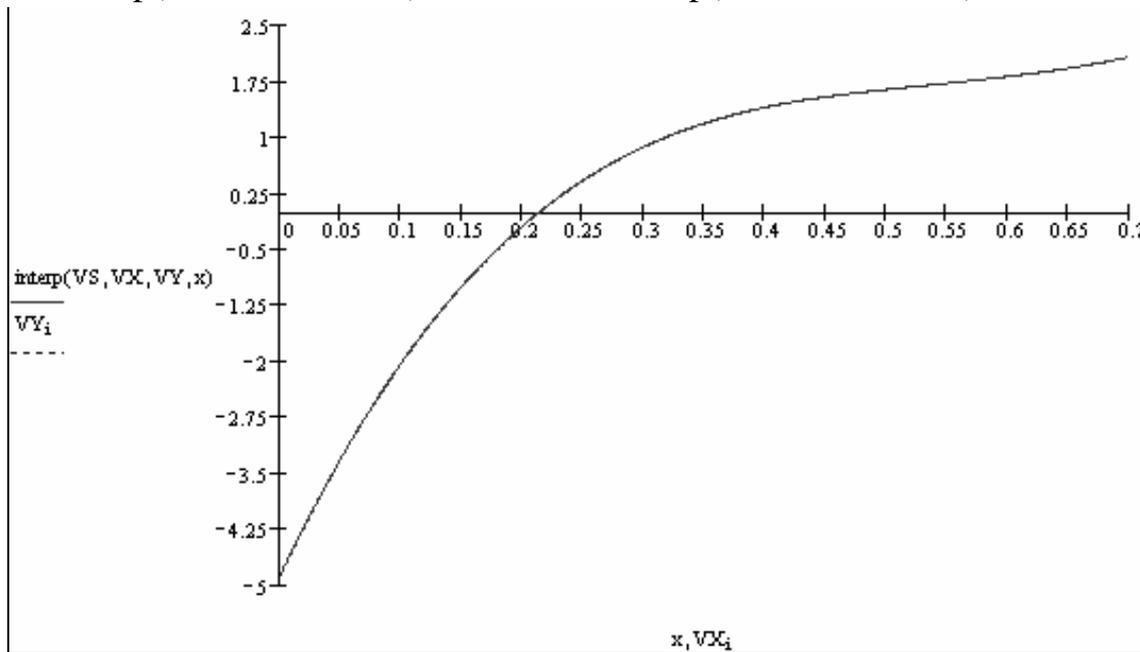


Рис. 8.11

В дополнение к $cspline$ в MathCAD имеются две другие кубические сплайн-функции: $pspline(VX, VY)$, $lspline(VX, VY)$. Все эти сплайн-функции возвращают вектор коэффициентов вторых производных, который называется VS . Он используется в функции $interp$. Аргументы VX и VY должны

быть вещественнозначными векторами одинаковой длины. Значения в VX должны быть вещественны и расположены в порядке возрастания.

Функция $lspline$ генерирует кривую сплайна, которая приближается к параболе в граничных точках.

Функция $pspline$ генерирует кривую сплайна, которая приближается к параболе в граничных точках.

Функция генерирует кривую сплайна, которая может быть кубическим полиномом в граничных точках.

Функция $interp(VS, VX, VY, X0)$ – возвращает интерполируемое значение y , соответствующее аргументу x . Вектор VS вычисляется на основе векторов данных VX, VY одной из функций $lspline, pspline$ или $cspline$.

Интерполируемое значение для конкретного x есть ордината y соответствующей точки сплайна.

8.10.3. Глобальная интерполяция

При глобальной интерполяции ищется единый полином для всего интервала. Если среди узлов $\{x_i, y_i\}$ нет совпадающих, то такой полином будет единственным, и его степень не будет превышать n . Запишем систему уравнений для определения коэффициентов полинома:

$$\begin{aligned} c_0 + c_1x_0 + c_2x_0^2 + \dots + c_{n-1}x_0^{n-1} &= y_0, \\ c_0 + c_1x_1 + c_2x_1^2 + \dots + c_{n-1}x_1^{n-1} &= y_1, \\ &\vdots \\ c_0 + c_1x_{n-1} + c_2x_{n-1}^2 + \dots + c_{n-1}x_{n-1}^{n-1} &= y_{n-1}. \end{aligned}$$

Определим матрицу коэффициентов системы уравнений

$$i := 0..n-1 \quad j := 0..n-1 \quad a_{(j,i)} := (x_j)^i.$$

Решим систему уравнений матричным методом $c := a^{-1} y$.

Определим интерполяционный полином $L(z) := \sum_i c_i z^i$.

Представим результаты на графике (рис. 8.12).

Вычислим значения интерполяционного полинома в заданных точках и сравним их с точными значениями:

X_k	$L(X_k)$	$\sin(X_k)$
-0.5	-0.594	-0.479
1.111	0.901	0.896
2.333	0.72	0.723
4.574	-1.036	-0.99

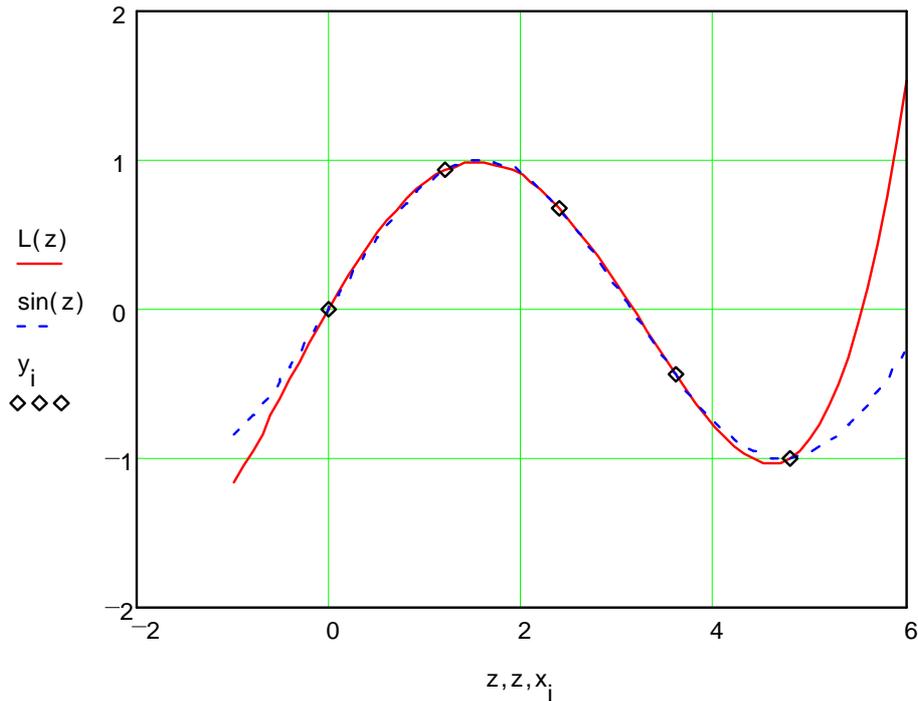


Рис. 8.12

Коэффициенты интерполяционного полинома следующие: $c = \begin{bmatrix} 0 \\ 1.134 \\ -0.177 \\ -0.127 \\ 0.022 \end{bmatrix}$.

Внимание! Из-за накопления вычислительной погрешности (ошибок округления) при большом числе узлов ($n > 10$) возможно резкое ухудшение результатов интерполяции. Кроме того, для целого ряда функций глобальная интерполяция полиномом вообще не дает удовлетворительного результата. Рассмотрим в качестве примера две таких функции. Для этих функций точность интерполяции с ростом числа узлов не увеличивается, а уменьшается.

Пример 1. Функция $f(x) := \frac{1}{1 + 25x^2}$.

Построим для нее интерполяционный полином на интервале $[-1; 1]$, используя 9 точек: $n := 9$ $i := 0..n-1$ $j := 0..n-1$.

$$x_j := -1 + \frac{j}{n-1} \cdot 2.$$

$$a_{(j,i)} := (x_j)^i \quad y_j := f(x_j) \quad c := a^{-1} \cdot y \quad L(z) := \sum_i c_i z^i.$$

Представим результаты на графике (рис. 8.13) $z := -1, -0.99..1$.

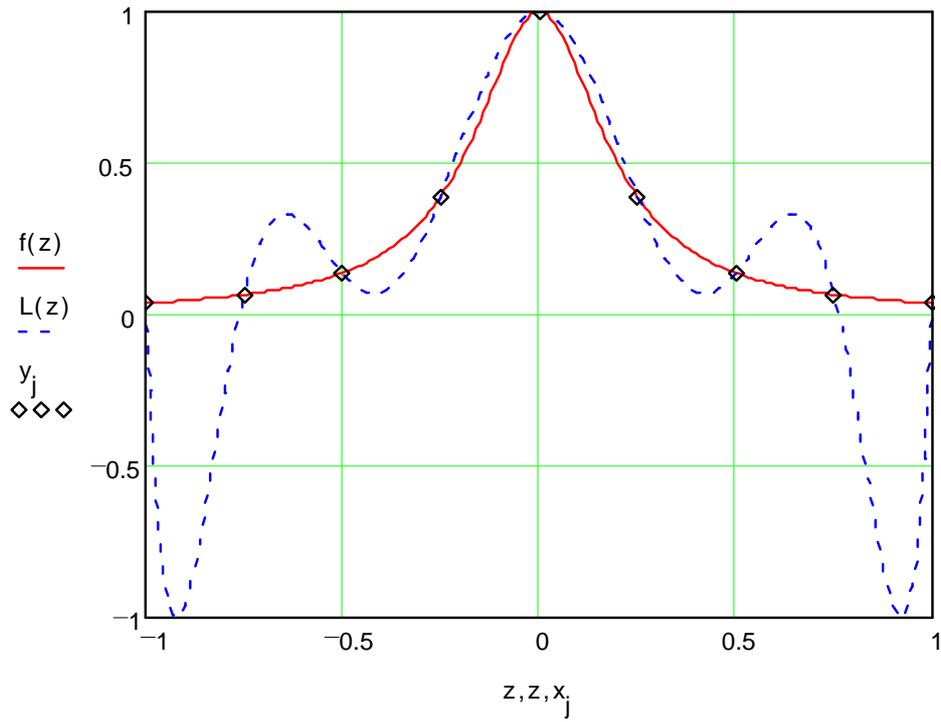


Рис. 8.13

Пример 2. Функция $g(x) := |x|$ (рис. 8.14).

Найдем интерполяционный полином, используя заданные точки.

$$y_j := g(x_j) \quad c := a^{-1} \cdot y \quad L(z) := \sum_i c_i z^i.$$

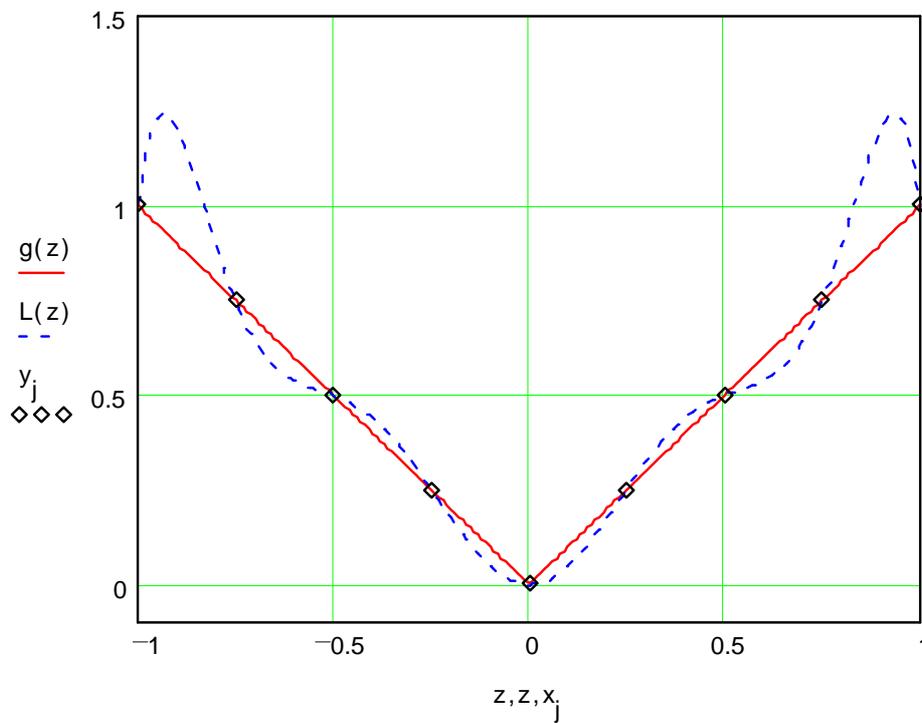


Рис. 8.14

Убедитесь самостоятельно, что при увеличении числа узлов интерполяции, результаты интерполирования вблизи концов интервала ухудшаются.

8.10.4. Метод наименьших квадратов

Наиболее распространенным методом аппроксимации экспериментальных данных является метод наименьших квадратов. Метод позволяет использовать аппроксимирующие функции произвольного вида и относится к группе глобальных методов. Простейшим вариантом метода наименьших квадратов является аппроксимация прямой линией (полиномом первой степени). Этот вариант метода наименьших квадратов носит также название линейной регрессии.

Критерием близости в методе наименьших квадратов является требование минимальности суммы квадратов отклонений от аппроксимирующей функции до экспериментальных точек:

$$\Phi = \sum_{i=1}^n (y_i - f(x_i))^2 \rightarrow \min.$$

8.10.5. Аппроксимация линейной функцией

Применим метод наименьших квадратов для аппроксимации экспериментальных данных.

Читаем данные из файлов *datax* и *datay*:

$$x := \text{READPRN}(\text{datax}) \quad y := \text{READPRN}(\text{datay}).$$

Имя файла следует заключать в кавычки и записывать его по правилам MS DOS, например, $\text{READPRN}(\text{"c:\mylib\datax.prn"})$.

Определяем количество прочитанных данных (число экспериментальных точек): $n := \text{last}(x) \quad i := 0 \dots n$.

Используем встроенные функции *slope* и *intercept* для определения коэффициентов линейной регрессии (аппроксимация данных прямой линией). Функция *slope* определяет угловой коэффициент прямой, а функция *intercept* – точку пересечения графика с вертикальной осью:

$$A := \text{intercept}(x, y), \quad B := \text{slope}(x, y).$$

Определяем аппроксимирующую функцию: $f_1(z) := A + Bz$.

Коэффициенты линейной регрессии – $A = -3.539, B = 1.81$.

Mathcad 2000 предлагает для этих же целей использовать функцию

$$\mathbf{line}: \quad \text{line}(x, y) = \begin{pmatrix} -3.539 \\ 1.81 \end{pmatrix}.$$

Вычислим стандартное отклонение (рис. 8.15).

$$S1 := \sqrt{\left[\frac{1}{(n-2)} \cdot \sum_i (f1(x_i) - y_i)^2 \right]} \quad S1 := 2.093 \quad z := 0, 0.1..10 \quad .$$

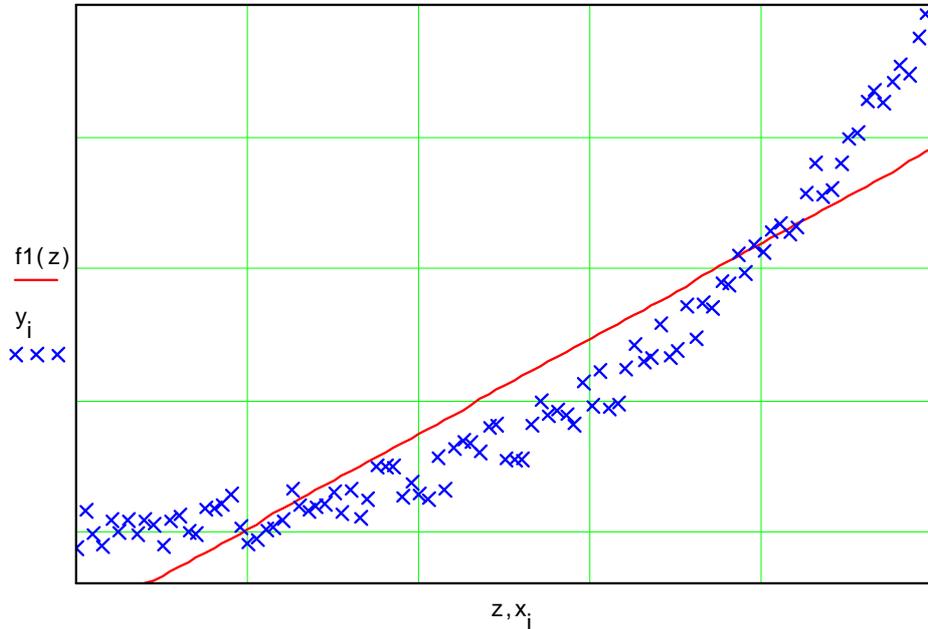


Рис. 8.15

8.10.6. Аппроксимация полиномами

Теперь попытаемся подобрать полиномы второй и третьей степени, в качестве аппроксимирующей функции. Для этих целей служат встроенные функции *regress* и *interp*. Очевидно, что если в качестве аппроксимирующей функции брать полином степени на единицу меньше числа точек, то задача сведется к задаче глобальной интерполяции и полученный полином будет точно проходить через все заданные узлы.

Вводим степени полиномов: $k2 := 2, \quad k3 := 3$.

Функция *regress* является вспомогательной, она подготавливает данные, необходимые для работы функции *interp*. Вектор *vs* содержит, в том числе, и коэффициенты полинома: $vs2 := regress(x,y,k2), \quad vs3 := regress(x,y,k3)$.

Функция *interp* возвращает значение полинома в точке *z*. Определив новые функции *f2*, *f3*, получаем возможность находить значение полинома в любой заданной точке:

$$\begin{aligned} f2(z) &:= interp(vs2,x,y,z) & f3(z) &:= interp(vs3,x,y,z) \\ coeffs2 &:= submatrix(vs2,3,length(vs2) - 1,0,0) \\ coeffs3 &:= submatrix(vs3,3,length(vs3) - 1,0,0). \end{aligned}$$

Коэффициенты:

$$\begin{aligned} (\text{coeffs2})^T &= (0.701 \quad -0.76 \quad 0.257), \\ (\text{coeffs3})^T &= (-0.122 \quad 0.253 \quad 2.377 \cdot 10^{-3} \quad 0.017), \end{aligned}$$

$$S2 := \sqrt{\left[\frac{1}{(n-2)} \cdot \sum_i (f2(x_i) - y_i)^2 \right]} \quad S2 := 0.67,$$

$$S3 := \sqrt{\left[\frac{1}{(n-2)} \cdot \sum_i (f3(x_i) - y_i)^2 \right]} \quad S3 := 0.58.$$

Стандартные отклонения почти не отличаются друг от друга, коэффициент при четвертой степени z невелик, поэтому дальнейшее увеличение степени полинома нецелесообразно и достаточно ограничиться только второй степенью.

К сожалению функция *regress* имеется далеко не во всех версиях Mathcad. Однако, провести полиномиальную регрессию можно и без использования этой функции. Для этого нужно определить коэффициенты нормальной системы и решить полученную систему уравнений, например, матричным методом.

Теперь попытаемся аппроксимировать экспериментальные данные полиномами степени m и $m1$, не прибегая к помощи встроенной функции *regress*.

$$\begin{aligned} m &:= 2 \quad t := 0..m \quad j := 0..m, \\ m1 &:= 3 \quad t1 := 0..m1 \quad j1 := 0..m1. \end{aligned}$$

Вычисляем элементы матрицы коэффициентов нормальной системы

$$p_{t,j} := \sum_i (x_i)^{t+j}, \quad p1_{t1,j1} := \sum_i (x_i)^{t1+j1}$$

и столбец свободных членов

$$b_j := \sum_i y_i (x_i)^j, \quad b1_{j1} := \sum_i y_i (x_i)^{j1}$$

Находим коэффициенты полинома, решая систему матричным методом,

$$a := p^{-1} \cdot b, \quad a1 := p1^{-1} \cdot b1.$$

Определяем аппроксимирующие функции (рис. 8.16, 8.17):

$$f2(z) := \sum_t (a_t) \cdot z^t, \quad f3(z) := \sum_{t1} (a1_{t1}) \cdot z^{t1}.$$

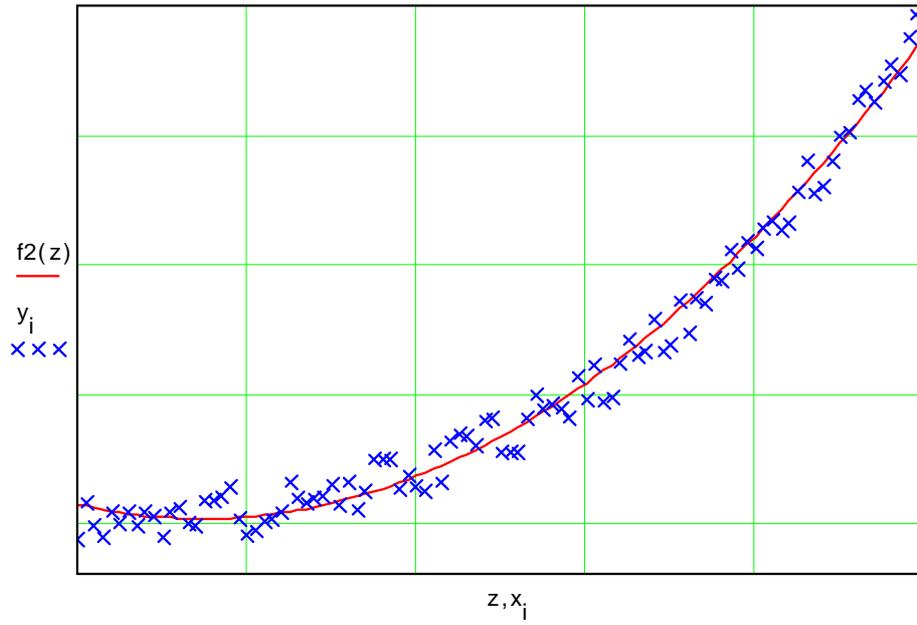


Рис. 8.16

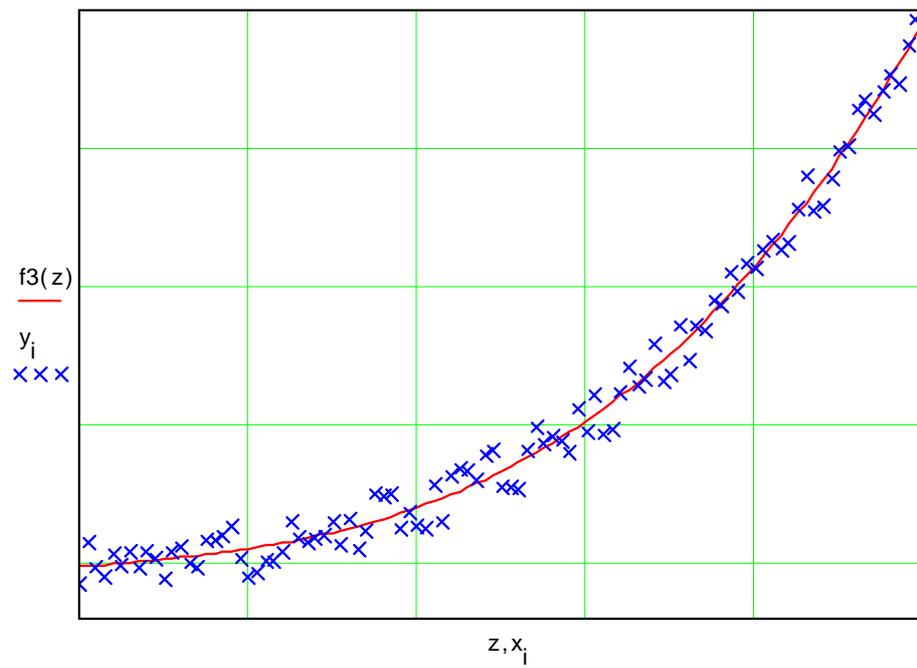


Рис. 8.17

Коэффициенты полиномов следующие:

$$a = \begin{pmatrix} 0.701 \\ -0.76 \\ 0.257 \end{pmatrix}, \quad a1 = \begin{bmatrix} -0.122 \\ 0.253 \\ 2.377 \cdot 10^{-3} \\ 0.017 \end{bmatrix}.$$

Вычислим стандартное отклонение:

$$S2 := \sqrt{\left[\frac{1}{(n-2)} \cdot \sum_i (f2(x_i) - y_i)^2 \right]} \quad S2 := 0.671,$$

$$S3 := \sqrt{\left[\frac{1}{(n-2)} \cdot \sum_i (f3(x_i) - y_i)^2 \right]} \quad S3 := 0.581.$$

8.10.7. Аппроксимация линейной комбинацией функций

Mathcad предоставляет пользователям встроенную функцию *linfit* для аппроксимации данных по методу наименьших квадратов линейной комбинацией произвольных функций.

Функция *linfit* имеет три аргумента:

- вектор x – x -координаты заданных точек,
- вектор y – y -координаты заданных точек,
- функция F – содержит набор функций, который будет использоваться для построения линейной комбинации.

Задаем функцию F (аппроксимирующая функция ищется в виде $a \frac{1}{x+1} + b \cdot x^2$):

$$F(x) := \begin{pmatrix} \frac{1}{x+1} \\ x^2 \end{pmatrix}, \quad S := \text{linfit}(x, y, F), \quad S = \begin{pmatrix} -0.802 \\ 0.176 \end{pmatrix}.$$

Определяем аппроксимирующую функцию (рис. 8.18):
 $f4(x) := F(x) \cdot S$.

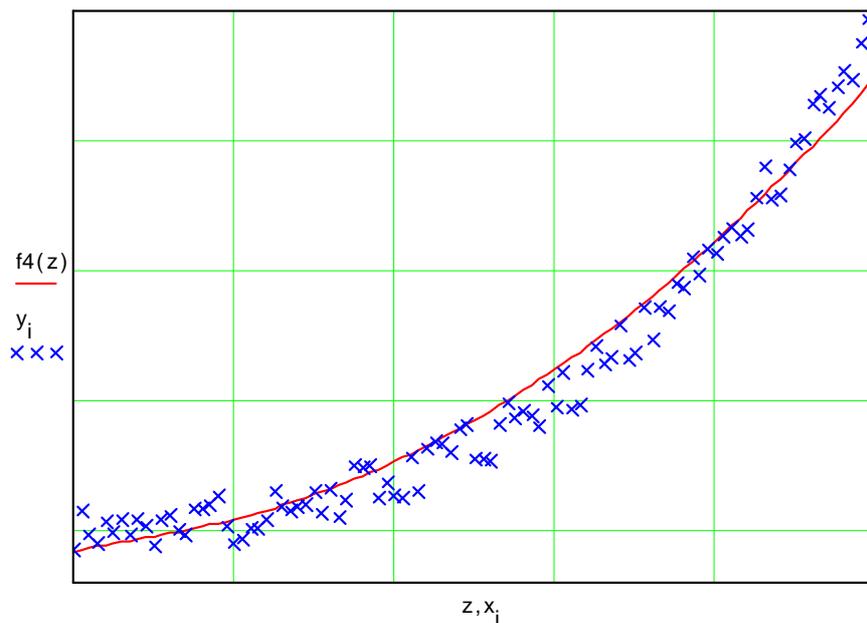


Рис. 8.18

Вычисляем дисперсию:

$$S4 := \sqrt{\left[\frac{1}{(n-2)} \sum_i (f4(x_i) - y_i)^2 \right]}, \quad S4 = 0.975.$$

8.10.8. Аппроксимация функцией произвольного вида

Теперь построим аппроксимирующую функцию дробно-рационального типа $f(x) = \frac{ax^2}{b+x}$. Для этого воспользуемся функцией *genfit*. Функция

имеет следующие параметры:

- x, y – векторы, содержащие координаты заданных точек;
- F – функция, задающая искомую функциональную n -параметрическую зависимость и частные производные этой зависимости по параметрам;
- v – вектор, задающий начальные приближения для поиска параметров.

$$F(z, u) := \begin{bmatrix} \frac{u_0 \cdot z^2}{u_1 + z} \\ \frac{z^2}{u_1 + z} \\ \frac{u_0 \cdot z^2}{(u_1 + z)^2} \end{bmatrix}, \quad v := \begin{pmatrix} -1 \\ -15 \end{pmatrix}, \quad S := \text{genfit}(x, y, v, F). \quad S = \begin{pmatrix} -2.146 \\ -20.85 \end{pmatrix}$$

Поскольку нулевой элемент функции F содержит искомую функцию, определяем функцию следующим образом (рис. 8.19): $f5(z) := F(z, S)_0$.

Вычисляем среднее квадратичное отклонение:

$$S5 := \sqrt{\left[\frac{1}{(n-2)} \sum_i (f5(x_i) - y_i)^2 \right]}, \quad S5 = 0.581.$$

Функция *genfit* имеется не во всех реализациях Mathcad. Возможно, однако, решить задачу, проведя линеаризацию.

Заданная функциональная зависимость может быть линеаризована введением переменных $z = \frac{1}{y}$ и $t = \frac{1}{x}$. Тогда $z = \frac{1}{a} + b \frac{t}{a}$.

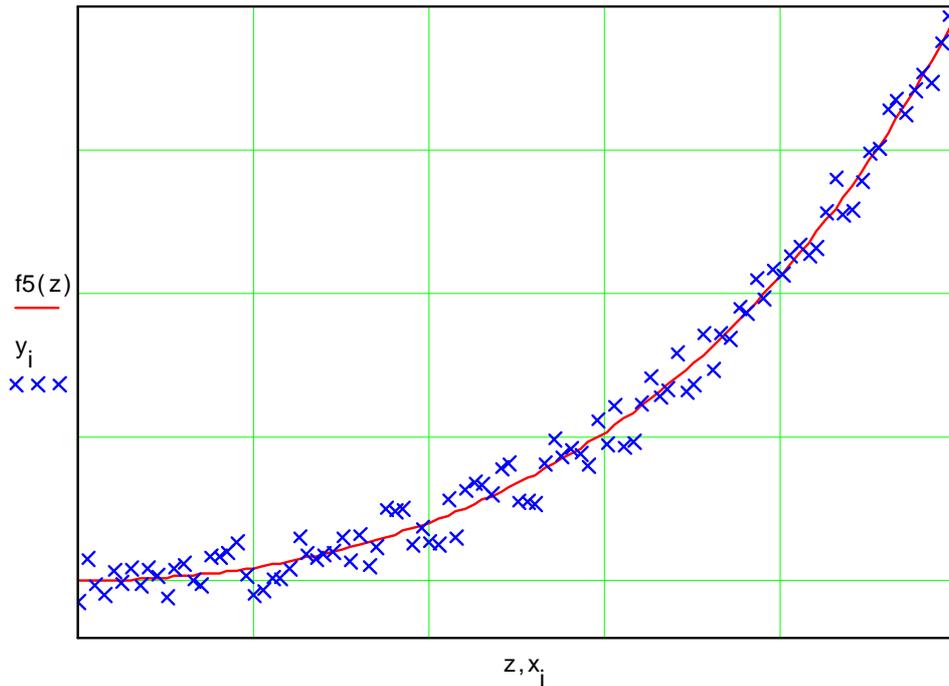


Рис. 8.19

Определим матрицы коэффициентов нормальной системы [8]:

$$e := \begin{bmatrix} \sum_i [(x_i)^3 \cdot y_i] \\ \sum_i [x_i \cdot (y_i)^2] \end{bmatrix} \quad o := \begin{bmatrix} \sum_i (x_i)^4 - \left[\sum_i (x_i)^2 \cdot y_i \right] \\ \sum_i [(x_i)^2 \cdot y_i] - \left[\sum_i (y_i)^2 \right] \end{bmatrix}$$

Находим коэффициенты функции, решая систему матричным методом,

$$d := o^{-1} e \quad d = \begin{pmatrix} -1.218 \\ -15.517 \end{pmatrix}.$$

Определяем функцию (рис. 8.20): $f5(z) := \frac{z^2 d_0}{d_1 + z}$.

Вычислим стандартное отклонение:

$$S5 := \sqrt{\left[\frac{1}{(n-2)} \sum_i (f5(x_i) - y_i)^2 \right]}, \quad S5 = 0.827.$$

Внимание! Мы получили другие коэффициенты! Вспомните, задача на нахождение минимума нелинейной функции, особенно нескольких переменных, может иметь несколько решений. Стандартное отклонение больше, чем в случае аппроксимации полиномами, поэтому следует остановить свой выбор на аппроксимации полиномом. Представим результаты аппроксимации на графике (см. рис. 8.20).

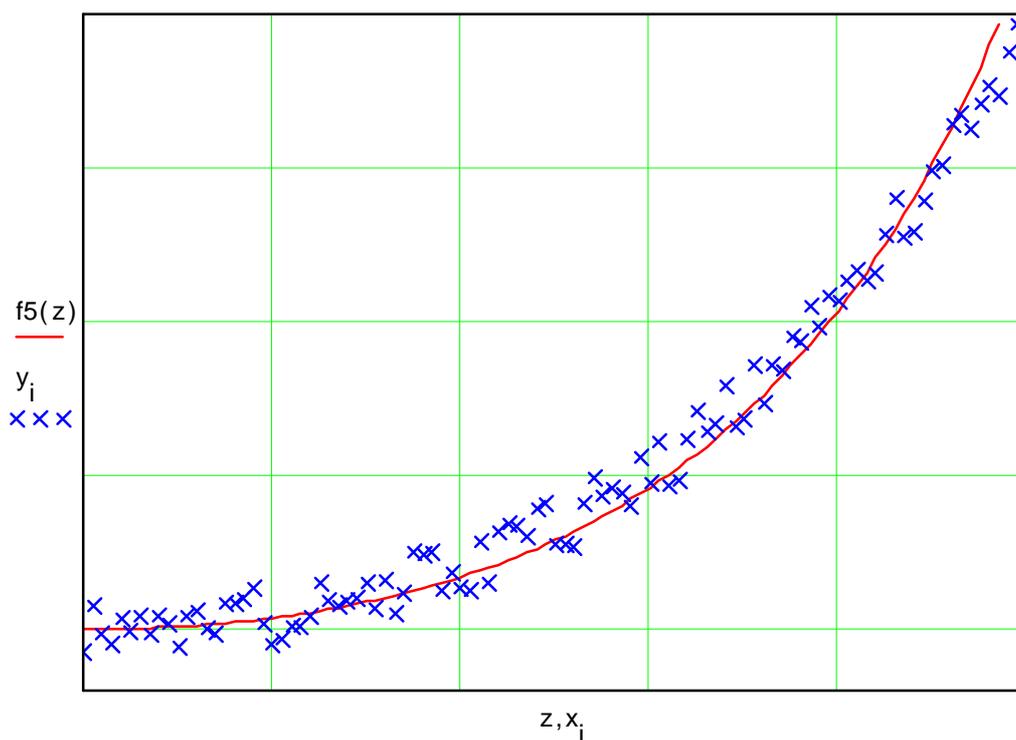


Рис. 8.20

В тех случаях, когда функциональная зависимость оказывается достаточно сложной, может оказаться, что самый простой способ нахождения коэффициентов – минимизация функционала Φ .

8.11. Вычисление определенных интегралов

Для вычисления определенного интеграла необходимо выбрать знак интеграла из палитры или набрать его нажатием клавиши &. После этого следует вписать пределы интегрирования, подынтегральную функцию и переменную интегрирования. Mathcad успешно справляется с большинством интегралов, в том числе с несобственными. Точность вычислений регулируется встроенной переменной TOL. По умолчанию ее значение установлено $TOL := 10^{-3}$.

Приведем несколько примеров успешного вычисления несобственных интегралов, интеграла от быстро осциллирующей функции и интеграла от ступенчатой функции:

$$\int_{-\infty}^{-1} \frac{1}{x^2} dx = 1,$$

$$\int_0^{\infty} x \cdot e^{-x^2} dx = 0.5,$$

$$\int_{10}^{20} \sin(x^3) dx = 1.822 \cdot 10^{-3},$$

$$\int_{-1}^1 \Phi(x) dx = 1.$$

Здесь
$$\Phi(x) = \begin{cases} 0, & \text{если } x < 0 \\ 1, & \text{если } x \geq 0 \end{cases}.$$

Зависимость результата от заданной точности вычислений:

$$\int_{-\infty}^{\infty} \frac{1}{1+x^2} dx = 3.14159265369356$$

$$\text{TOL} := 10^{-6}, \quad \int_{-\infty}^{\infty} \frac{1}{1+x^2} dx = 3.14159265358979$$

Для этого примера результат может быть получен также в символьном виде. Для этого вместо знака равенства необходимо нажать знак символического равенства `Ctrl+`

$$\int_{-\infty}^{\infty} \frac{1}{1+x^2} dx \rightarrow \pi.$$

В то же время в некоторых случаях несобственные интегралы вычисляются неправильно:

$$\int_{-1}^1 \frac{1}{x^2} dx = 1.376 \cdot 10^3.$$

Хотя очевидно, что
$$\int_{-1}^1 \frac{1}{x^2} dx \rightarrow \infty.$$

8.12. Решение дифференциальных уравнений

8.12.1. Обыкновенные дифференциальные уравнения

Пусть необходимо найти решение уравнения $y' = f(x, y)$ с начальным условием $y(x_0) = y_0$. Такая задача называется *задачей Коши*. Разложим искомую функцию $y(x)$ в ряд вблизи точки x_0 и ограничимся первыми двумя членами разложения $y(x) = y(x_0) + y'(x)(x - x_0) + \dots$. Учтя уравнение и обозначив $x - x_0 = h$, получаем $y(x) = y(x_0) + f(x_0, y_0)\Delta x$. Эту формулу можно применять многократно, находя значения функции во все новых и новых точках:

$$y_{i+1} = y_i + f(x_i, y_i)h.$$

Такой метод решения обыкновенных дифференциальных уравнений называется методом Эйлера. Геометрически метод Эйлера означает, что на каждом шаге мы аппроксимируем решение (интегральную кривую) отрезком касательной, проведенной к графику решения в начале интервала. Точность метода невелика и имеет порядок h . Говорят, что метод Эйлера – метод первого порядка, т.е. его точность растет линейно с уменьшением

шага h . Существуют различные модификации метода Эйлера, позволяющие увеличить его точность. Все они основаны на том, что производную, вычисленную в начале интервала, заменяют на среднее значение производной на данном интервале.

Оценку значения производной можно улучшить, увеличивая число вспомогательных шагов. На практике наиболее распространенным методом решения обыкновенных дифференциальных уравнений **является метод Рунге – Кутты** четвертого порядка. Для оценки значения производной в этом методе используется четыре вспомогательных шага. Формулы метода Рунге – Кутты следующие:

$$\begin{aligned}
 k_1^i &= hf(x_i, y_i), \\
 k_2^i &= hf\left(x_i + \frac{h}{2}, y_i + \frac{k_1^i}{2}\right), \\
 k_3^i &= hf\left(x_i + \frac{h}{2}, y_i + \frac{k_2^i}{2}\right), \\
 k_4^i &= hf(x_i + h, y_i + k_3^i), \\
 \Delta y_i &= \frac{1}{2}(k_1^i + 2k_2^i + 2k_3^i + k_4^i), \\
 y_{i+1} &= y_i + \Delta y_i.
 \end{aligned}$$

Перечисленные методы можно применять и для решения систем дифференциальных уравнений. Поскольку многие дифференциальные уравнения высших порядков могут быть сведены заменой переменных к системе дифференциальных уравнений первого порядка, рассмотренные методы могут быть использованы и для решения дифференциальных уравнений выше первого порядка.

Еще один тип задач, часто встречающихся на практике, – краевые задачи. Пусть имеется дифференциальное уравнение второго порядка $y'' = f(x, y, y')$. Решение уравнения требуется найти на интервале $[0, 1]$, причем известно, что $y(0) = y_0$, $y(1) = y_1$. Понятно, что произвольный интервал $[a, b]$ заменой переменных $t = \frac{x-a}{b-a}$ может быть сведен к единично-

му. Для решения краевой задачи обычно применяют **метод стрельбы**. Пусть $y'(0) = k$, где k – некоторый параметр. Для некоторого пробного значения k может быть решена задача Коши, например, методом Рунге – Кутты. Полученное решение будет зависеть от значения параметра $y = y(x; k)$.

8.12.2. Метод Эйлера для дифференциальных уравнений первого порядка

Решим задачу Коши для дифференциального уравнения первого порядка $y' = f(x, y)$ методом Эйлера.

Пусть правая часть уравнения равна $f(x, y) \equiv x \cdot y$.

Зададим границы изменения x : $x_{\min} \equiv 0$ $x_{\max} \equiv 1$.

Зададим число точек и величину шага: $n \equiv 10$ $h = \frac{(x_{\max} - x_{\min})}{n}$.

Зададим начальные условия: $y_0 \equiv 1$ $x_0 \equiv x_{\min}$.

Вычислим x и y по формулам Эйлера $j \equiv 1..n$, $x_j \equiv x_{\min} + j \cdot h$,
 $y_j \equiv y_{j-1} + f(x_{j-1}, y_{j-1}) \cdot h$.

Представим результат графически (рис. 8.21) и сравним его с аналитическим решением $z := 0,0.1..1$, $y1(z) := \exp\left(\frac{z^2}{2}\right)$, $k \equiv 0..n$.

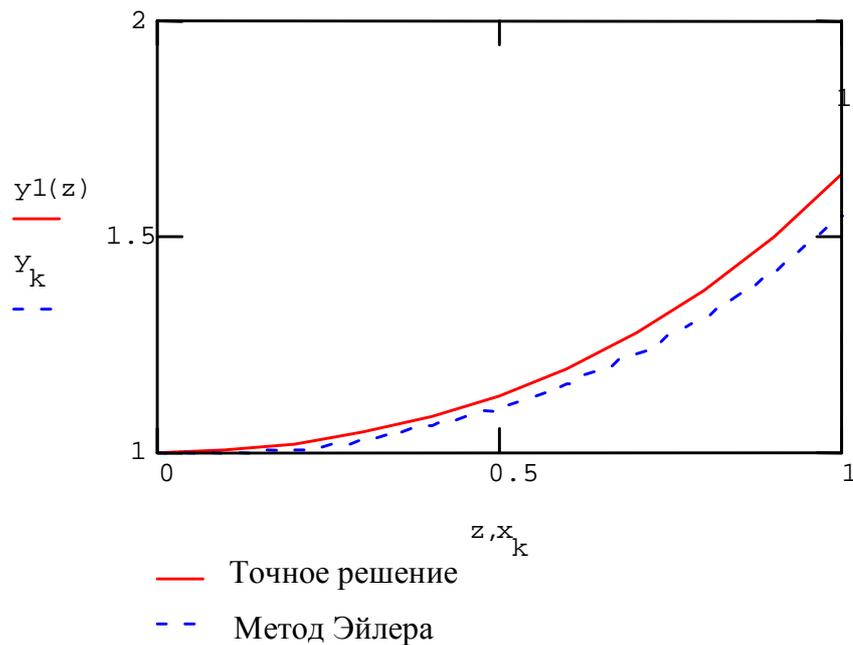


Рис. 8.21

Точное аналитическое решение и решение, полученное численно, отличаются в точке $x = 1$ на $y1(1) - y_n = 0.102$.

То есть относительная ошибка составляет $\frac{y1(1) - y_n}{y1(1)} = 6.163\%$.

Аргументы функции:

- y – вектор начальных значений (n элементов).
- $x1$ и $x2$ – границы интервала, на котором ищется решение дифференциального уравнения.
- $npoints$ – число точек внутри интервала ($x1, x2$), в которых ищется решение. Функция *rkfixed* возвращает матрицу, состоящую из $1 + npoints$ строк.
- D – вектор, состоящий из n элементов, который содержит первые производные искомой функции.

В качестве примера рассмотрим решение системы Вольтерра – Лотки. Эта система описывает динамику численности хищников и жертв на замкнутом ареале и является одной из базовых моделей экологии.

$$\begin{aligned}\frac{dN_1}{dt} &= N_1(\varepsilon_1 - \gamma_2 N_2), \\ \frac{dN_2}{dt} &= N_2(\varepsilon_2 - \gamma_1 N_1).\end{aligned}$$

Для решения систем дифференциальных уравнений используются функция *rkfixed*.

Внимание! В этом примере установлено значение *ORIGIN* = 1, т.е. нумерация элементов массива начинается с 1, а не с 0, как это принято в Mathcad по умолчанию.

Пусть в начальный момент времени число хищников $N_1 = 5$ и число жертв $N_2 = 10$.

Задаем вектор начальных значений $N := \begin{pmatrix} 5 \\ 10 \end{pmatrix}$, параметры системы

$\varepsilon := \begin{pmatrix} 0.1 \\ 0.3 \end{pmatrix}$, $y := \begin{pmatrix} 0.03 \\ 0.04 \end{pmatrix}$, интервал времени и количество точек, в которых

будет вычислено решение, $t_{\max} := 200$, $npoints := 400$ и вектор правых частей системы. (Поскольку исходная система не зависит явно от времени t , функция D также не зависит от времени явно хотя и содержит его в числе своих аргументов.)

$$D(t, N) := \begin{bmatrix} N_1 \cdot (\varepsilon_1 - \gamma_2 \cdot N_2) \\ -N_2 \cdot (\varepsilon_2 - \gamma_1 \cdot N_1) \end{bmatrix}.$$

Решаем систему с помощью встроенной функции

$$Z := \text{rkfixed}(N, 0, t_{\max}, \text{npoints}, D), \quad k := 1 \dots \text{npoints}.$$

Представим на графике результаты расчета – зависимость численности популяций от времени (рис. 8.22) и зависимость числа жертв от числа хищников (рис. 8.23).

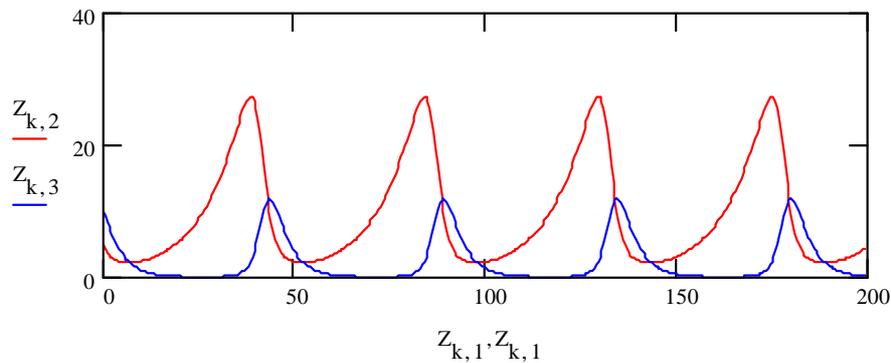


Рис. 8.22

Можно использовать обозначения $(z^{<i>>})_k$ или $Z_{k,i}$ – это одно и то же.

Поскольку дифференциальное уравнение порядка выше первого может быть преобразовано к системе дифференциальных уравнений первого порядка, функция *rkfixed* может быть использована и для решения дифференциальных уравнений.

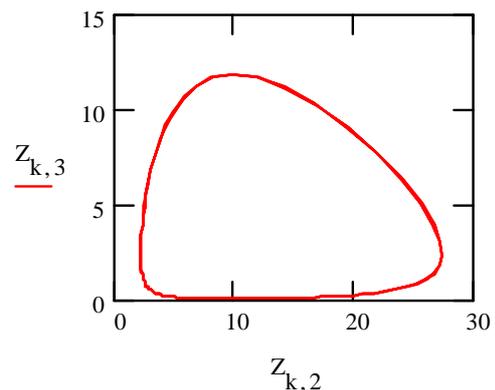


Рис. 8.23

8.12.5. Решение дифференциальных уравнений второго порядка

В качестве примера решим задачу о гармоническом осцилляторе, для которого известно аналитическое решение, и легко может быть оценена точность вычислений. Дифференциальное уравнение второго порядка

$$y'' + 2\beta y' + \omega^2 y = 0$$

преобразуем к системе из двух дифференциальных уравнений первого порядка

$$\begin{aligned} y' &= x, \\ x' &= -2\beta x - \omega^2 y. \end{aligned}$$

Пусть декремент затухания $\beta := 0.0$ и циклическая частота $\omega := 1$.

Зададим начальные условия $y := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

Начальной координате соответствует y_0 , а y_1 – начальной скорости. Зададим теперь матрицу D . С учетом того, что искомая величина соответствует нулевому элементу массива y , ее первая производная – первому, а вторая – второму, имеем

$$D(t, y) := \begin{pmatrix} y_1 \\ -2\beta y_1 - \omega^2 y_0 \end{pmatrix}, \quad Z := \text{rkfixed}(y, 0, 5\pi, 100, D).$$

Представим результаты расчета на графике (рис. 8.24) и сравним их с аналитическим решением $f(x) := y_1 \exp(-\beta \cdot x) \cdot \cos(\omega \cdot x)$, $k \equiv 0..100$ $x := 0, 0.1..5\pi$.

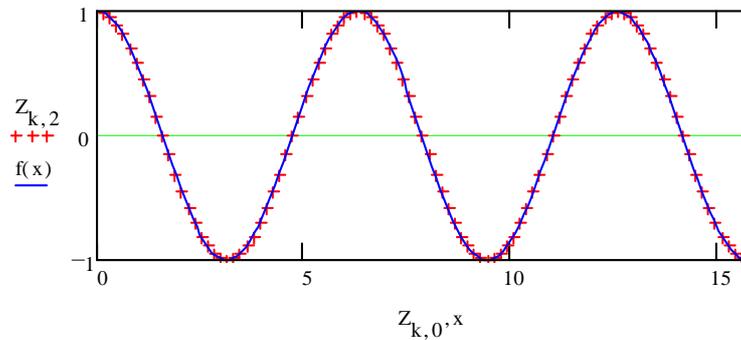


Рис. 8.24

Для контроля точности вычислений нарисуем фазовую траекторию (зависимость смещения от скорости) (рис. 8.25). Для гармонического осциллятора фазовая траектория должна иметь вид эллипса.

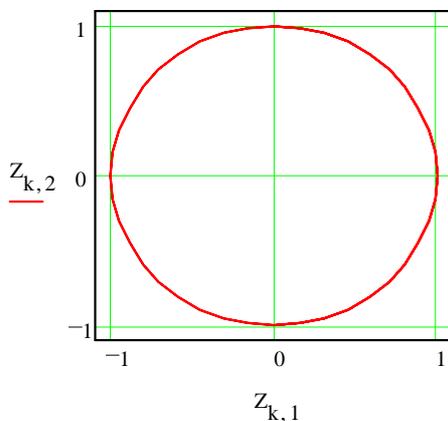


Рис. 8.25

Mathcad имеет еще две функции для решения задачи Коши. Это функции *Rkadapt* и *Bulstoer*. Эти функции имеют те же самые аргументы и возвращают решения в такой же форме, что и функция *rkfixed*. Первая из этих функций использует метод Рунге – Кутты с переменным шагом, что позволяет повысить точность вычислений и сократить их объем, если искомое решение имеет области, где ее значения меняются быстро, и области плавного

изменения. Функция *Rkadapt* будет варьировать величину шага в зависимости от скорости изменения решения.

Функция *Bulstoer* реализует иной численный метод – метод Булирша – Штера. Ее следует применять, если известно, что решение является гладкой функцией.

8.12.6. Решение краевой задачи

Пусть требуется найти решение дифференциального уравнения $y'' + 2\beta y + \omega_0^2 y = 0$ при условиях $y(0) = 1$ и $y(5\pi) = 0$.

При значениях параметров $\beta := 0.1$, $\omega_0 := 1$.

Для решения краевой задачи имеется встроенная функция *sbval*, реализующая метод стрельб и позволяющая свести краевую задачу к задаче Коши. Функция *sbval* имеет следующие параметры:

- v – вектор, содержащий начальные приближения для недостающих начальных условий;
- x_{min}, x_{max} – границы интервала, на котором ищется решение;
- $D(x, y)$ – вектор-функция, содержащая правые части системы дифференциальных уравнений первого порядка, эквивалентной исходному уравнению, размер вектора n совпадает со степенью старшей производной дифференциального уравнения;
- $load(xmin, v)$ – вектор-функция, элементы которой соответствуют n значениям функций на левой границе интервала. Часть этих значений известна, а для части заданы начальные приближения в векторе v . Их уточненные значения будут найдены в процессе вычисления
- $score(xmax, y)$ – вектор-функция, имеющая то же число элементов, что и v . Каждое значение является разностью между начальными значениями в конечной точке интервала и соответствующей оценки для решения. Этот вектор показывает, на сколько близко найденное решение к истинному.

Наша задача сводится к системе двух дифференциальных уравнений первого порядка:

$$\begin{aligned} y' &= z, \\ z' &= -2\beta z - \omega_0^2 y. \end{aligned}$$

Поэтому функция D имеет вид $D(t, y) := \begin{pmatrix} y_1 \\ -2\beta y_1 - \omega_0^2 y_0 \end{pmatrix}$.

Задаем граничные условия: $x_{\min} = 0$, $x_{\max} = 5\pi$.

Задача Коши для дифференциального уравнения второго порядка содержит два начальных условия. Нам известно только одно. Начальное приближение для недостающего значения задаем в векторе v , который в нашем случае состоит только из одного элемента. Несмотря на это, индекс 0 должен быть обязательно указан, чтобы подчеркнуть векторный характер этой величины: $v_0 := 0.1$.

На левой границе интервала нам известно значение y ($y(0) = 1$) и задано начальное приближение для y' ($y' = 0$). Это значение записано в v_0 . Задаем вектор-функцию *load*. Ее нулевой элемент – начальное значение для y_0 , первый – для y_1 .

$$\text{load}(x_{\min}, v) := \begin{pmatrix} 1 \\ v_0 \end{pmatrix}, \quad \text{score}(x_{\max}, y) := y_0 - 0,$$

$$S := \text{sbval}(v, x_{\min}, x_{\max}, D, \text{load}, \text{score}), \quad S = 12.511.$$

Теперь, когда нам стало известно недостающее начальное условие в задаче Коши, можно воспользоваться, например, функцией *rkfixed* (рис. 8.26):

$$n := 500 \quad i := 0..n \quad y := \begin{pmatrix} 1 \\ S_0 \end{pmatrix} \quad Z := \text{rkfixed}(y, x_{\min}, x_{\max}, 500, D).$$

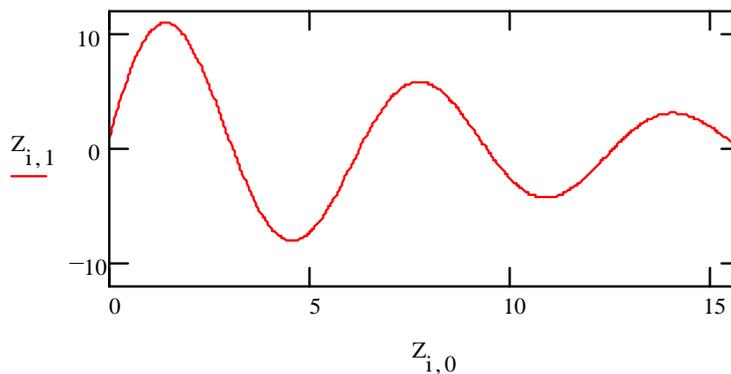


Рис. 8.26

8.12.7. Решение обыкновенных дифференциальных уравнений в Mathcad

Mathcad предлагает новый способ для решения обыкновенных дифференциальных уравнений, разрешенных относительно старшей производ-

ной. Для этих целей служит уже известный нам блок *given* совместно с функцией *odesolve*. Дифференциальное уравнение совместно с начальными или граничными условиями записывается в блоке *Given*. Производные можно обозначать как штрихами (Ctrl+F7), так и с помощью знака производной $\frac{d}{dx}$. Приведем пример использования функции для решения задачи Коши (рис. 8.27):

```

β := 0.1
Given
x''(t) + 2βx'(t) + x(t) = 0
x(0) = 1
x'(0) = 1
x := odesolve(t, 5π, 0.1)
t := 0, 0.1..5π

```

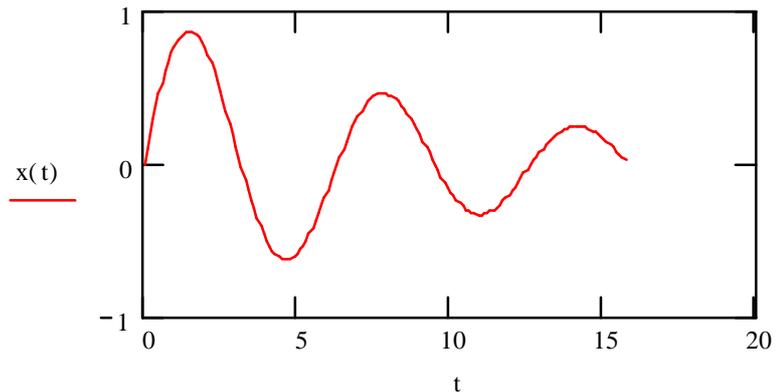


Рис. 8.27

У искомой функции явно указан аргумент, знак производной стоит перед скобкой. Функция *odesolve* имеет три аргумента. Первый аргумент – независимая переменная, вторая – граница интервала, на котором ищется решение, последний аргумент – шаг, с которым ищется решение. Последний аргумент может быть опущен.

Следующий пример демонстрирует решение краевой задачи. Показан другой способ записи производных, используется *odesolve* функция с двумя аргументами (рис. 8.28):

```

Given
 $\frac{d^2}{dt^2} x(t) + 2\beta \frac{d}{dt} x(t) + x(t) = 0$ 
x(0) = 1
x(5π) = 0.1
x := odesolve(t, 5π)
t := 0, 0.1..5π

```

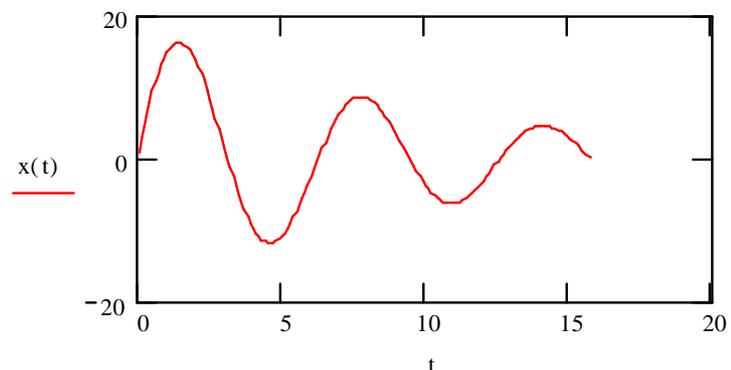


Рис. 8.28

8.13. Решение уравнений в частных производных

Одним из методов решения дифференциальных уравнений в частных производных является *метод сеток*. Идея метода заключается в следующем. Для простоты, ограничимся случаем только функции двух переменных, и будем полагать, что решение уравнения ищется на квадратной области единичного размера. Разобьем область сеткой. Шаг сетки по оси x и по оси y , вообще говоря, может быть разный. По определению частная производная равна

$$\frac{\partial u(x, y)}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{u(x + \Delta x, y) - u(x, y)}{\Delta x} \approx \frac{u(x + \Delta x, y) - u(x, y)}{\Delta x}.$$

Если рассматривать функцию только в узлах сетки, то частную производную можно записать в форме

$$\frac{\partial u(x, y)}{\partial x} \approx \frac{u_{i+1, j} - u_{i, j}}{h},$$

где узел (i, j) соответствует точке (x, y) . Полученное выражение называется *правой конечной разностью*. Название связано с тем, что для вычисления производной в точке используются значения функций в этой точке и точке, лежащей правее. Очевидно, что сходное выражение можно было бы получить, используя точку, лежащую слева

$$\frac{\partial u(x, y)}{\partial x} \approx \frac{u_{i, j} - u_{i-1, j}}{h}.$$

Такое выражение называется *левой конечной разностью*. Можно получить центральную конечную разность, найдя среднее этих выражений.

Теперь получим выражения для вторых производных

$$\frac{\partial^2 u(x, y)}{\partial x^2} \approx \frac{u_{i+1, j} - 2u_{i, j} + u_{i-1, j}}{h^2}.$$

В данном случае для нахождения производной использованы симметричные точки. Однако, очевидно, можно было бы использовать точки с несимметричным расположением.

8.13.1. Уравнения гиперболического типа

В качестве примера рассмотрим решение волнового уравнения (уравнения гиперболического типа).

$$\frac{\partial^2 U}{\partial x^2} = \frac{1}{v^2} \frac{\partial^2 U}{\partial t^2}.$$

Уравнение будем решать методом сеток. Запишем уравнение в конечных разностях

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} = \frac{1}{v^2} \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j}}{\tau^2}.$$

Полученное уравнение позволяет выразить значение функции u в момент времени $j+1$ через значения функции в предыдущие моменты времени

$$u_{i,j+1} = v^2 \left(\frac{\tau}{h} \right)^2 (u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) + 2u_{i,j} - u_{i,j-1}.$$

Такая разностная схема называется **явной**, т.к. искомая величина получается в явном виде. Она устойчива, если $\tau \leq h/v$.

Зададим начальные условия: смещение струны U в начальный и последующий моменты времени описывается синусоидальной функцией:

$$n := 20 \quad j := 0..n \quad i := 0..100$$

$$U_{i,0} := \sin\left(\pi \frac{i}{50}\right) \quad U_{i,1} := U_{i,0}.$$

Совпадение смещений при $j = 0$ и $j = 1$ соответствует нулевой начальной скорости.

Зададим граничные условия: на концах струны смещение равно 0 в любой момент времени: $U_{0,j} := 0$, $U_{100,j} := 0$.

Будем полагать коэффициент $a := 1$, $k := 0.02$, $i := 1..99$, $j := 1..n - 1$.

Записываем уравнение в конечных разностях, разрешенное относительно $U_{i,j+1}$

$$U_{i,j+1} := a^2 k (U_{i+1,j} - 2U_{i,j} + U_{i-1,j}) + 2U_{i,j} - U_{i,j-1}.$$

Представляем результат на графике (рис. 8.29)

8.13.2. Уравнения параболического типа

Еще один пример использования конечных разностей – уравнение диффузии

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2}.$$

Это уравнение параболического типа. Явная разностная схема для этого уравнения имеет вид

$$u_{i,j+1} = D \frac{\tau}{h^2} (u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) + u_{i,j}. \quad (8.1)$$

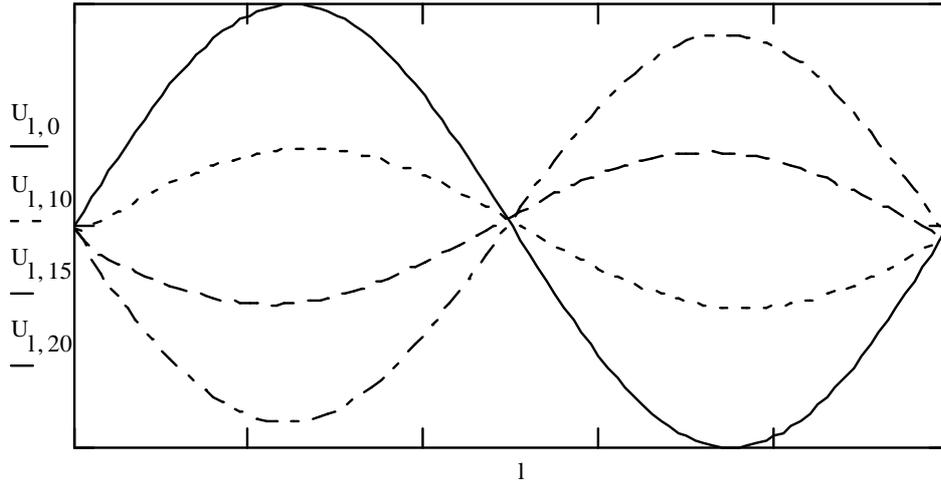


Рис. 8.29

Эта разностная схема устойчива, если $\tau \leq \frac{h^2}{2D}$. Для краткости в дальнейшем будем обозначать весь множитель, стоящий перед скобкой, как κ :

Задаем коэффициент $\kappa := 0.15$ и диапазон изменения пространственной и временной координат: $t := 0..29$, $x := 1..49$.

Задаем начальные и граничные условия: $f_{0,x} := 0$, $f_{0,0} := 0$, $f_{0,50} := 0$, $f_{0,25} := 1$.

Уравнение в конечных разностях имеет вид

$$f_{t+1,x} := f_{t,x} + \kappa(f_{t,x-1} - 2f_{t,x} + f_{t,x+1}).$$

Представляем результаты на графике (рис. 8.30) (для большей наглядности изображена только центральная часть) $x := 15..35$.

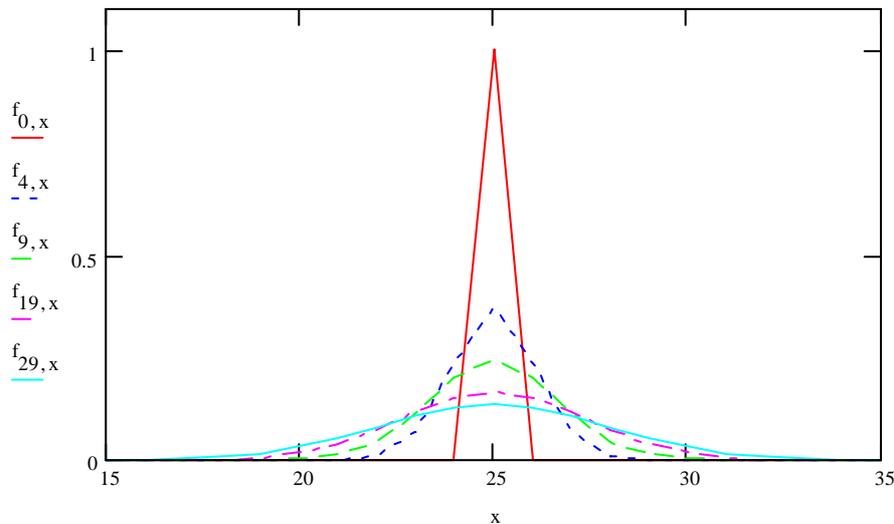


Рис. 8.30

Основное достоинство явных методов – их простота – зачастую сводится на нет достаточно жесткими ограничениями на величину шага. Явные схемы обычно устойчивы при столь малых шагах по времени, что они становятся непригодными для практических расчетов. Этому существенного недостатка позволяют избежать неявные схемы. Свое название они получили потому, что значения искомой функции на очередном временном шаге не могут быть явно выражены через значения функции на предыдущем шаге.

Рассмотрим применение неявной схемы на примере уравнения теплопроводности

$$\frac{\partial u}{\partial t} = c \frac{\partial^2 u}{\partial x^2} \quad (8.2)$$

Запишем неявную разностную схему для этого уравнения

$$\frac{u_{i,j+1} - u_{i,j}}{\tau} = \frac{c}{h^2} (u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}). \quad (8.3)$$

Здесь первый индекс соответствует пространственной, а второй – временной координате. В отличие от явной схемы, для вычисления в правой части уравнения используются значения функции на том же самом временном шаге. Вводя обозначение $\mu = \frac{c\tau}{h^2}$, уравнение (8.3) можно переписать в виде

$$(1 + 2\mu)u_{i,j+1} - \mu(u_{i+1,j+1} + u_{i-1,j+1}) = u_{i,j} \quad (8.4)$$

или в матричной форме

$$\begin{pmatrix} 1+2\mu & -\mu & & & \\ -\mu & 1+2\mu & -\mu & & \\ & & \dots & & \\ & & & 1+2\mu & -\mu \\ & & & -\mu & 1+2\mu \end{pmatrix} \begin{pmatrix} u_{1,j+1} \\ u_{2,j+1} \\ \vdots \\ u_{n-1,j+1} \\ u_{n,j+1} \end{pmatrix} = \begin{pmatrix} u_{1,j} + \mu\alpha \\ u_{2,j} \\ \vdots \\ u_{n-1,j} \\ u_{n,j} + \mu\beta \end{pmatrix} \quad (8.5)$$

где $u(0,t) = \alpha$, $u(t,1) = \beta$.

Задаем количество узлов сетки (в данном случае оно одинаково для обеих переменных) $n := 30$, $i := 0 \dots n$, $j := 0 \dots n$, $k := 0 \dots n - 1$, $m := 0 \dots n - 1$.

Задаем значения параметров $\alpha := 0$, $\beta := 1$, $\mu := 5$ и начальное распределение температуры в области $u_{i,0} := \sin\left(\pi \cdot \frac{i}{n}\right) + \frac{i}{n}$.

Формируем матрицы уравнения (8.5)

$$\begin{aligned} u_{0,j} &:= \alpha & u_{n,j} &:= \beta \\ A_{i,i} &:= 1 + 2\mu & A_{m,m-1} &:= -\mu & A_{m-1,m} &:= -\mu \\ \mu\alpha_i &:= 0 & \mu\alpha_0 &:= \mu\alpha & \mu\beta_i &:= 0 & \mu\beta_0 &:= \mu\beta \end{aligned}$$

Находим решение системы (рис. 8.31)

$$u^{(j+1)} := A^{-1} \cdot \left(u^{(j)} + \mu\alpha + \mu\beta \right).$$

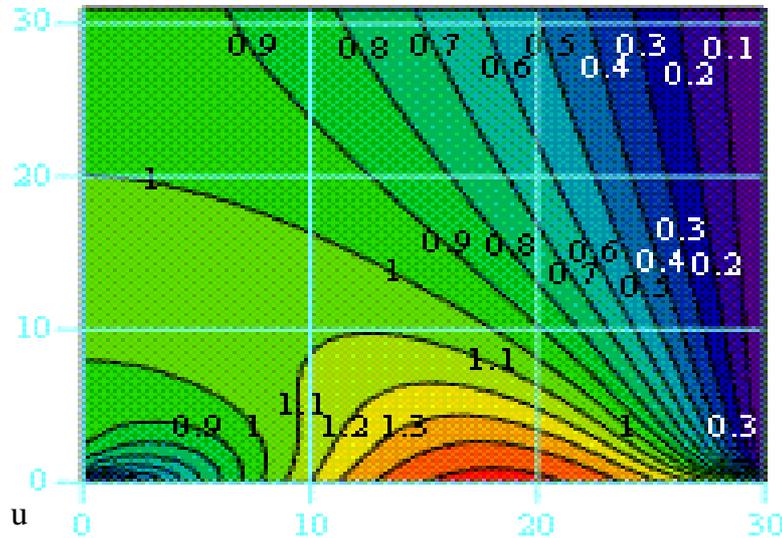


Рис. 8.31

8.13.3. Решение уравнений Лапласа и Пуассона

Для решения уравнений Пуассона $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = F(x, y)$ и Лапласа (частный случай, когда $F(x, y) = 0$) – уравнений эллиптического типа – предназначена функция *relax(a, b, c, d, e, f, u, rjac)*, реализующая метод релаксации. Фактически эту функцию можно использовать для решения эллиптического уравнения общего вида

$$A \frac{\partial^2 u}{\partial x^2} + 2B \frac{\partial^2 u}{\partial x \partial y} + C \frac{\partial^2 u}{\partial y^2} + a \frac{\partial u}{\partial x} + b \frac{\partial u}{\partial y} + cu = F(x, y),$$

$$D = AC - B^2 > 0,$$

которое может быть сведено к уравнению в конечных разностях

$$a_{i,j}u_{i+1,j} + b_{i,j}u_{i-1,j} + c_{i,j}u_{i,j+1} + d_{i,j}u_{i,j-1} + e_{i,j}u_{i,j} = f_{i,j}.$$

В частности, для уравнения Пуассона коэффициенты

$$a_{i,j} = b_{i,j} = c_{i,j} = d_{i,j} = 1, \quad e_{i,j} = -4.$$

Идея метода релаксации заключается в следующем. Если нет источников (уравнение Лапласа), то значение функции в данном узле на текущем шаге $k + 1$ определяется как среднее значение функции в ближайших узлах на предыдущем шаге k

$$u_{i,j}^{k+1} = \frac{1}{4}(u_{i-1,j}^k + u_{i+1,j}^k + u_{i,j-1}^k + u_{i,j+1}^k) \quad (8.6)$$

При наличии источников разностная схема имеет вид

$$u_{i,j}^{k+1} = \frac{1}{4}(u_{i-1,j}^k + u_{i+1,j}^k + u_{i,j-1}^k + u_{i,j+1}^k) - \frac{h^2}{4} f_{i,j} \quad (8.7)$$

Метод релаксации сходится достаточно медленно, т.к. фактически он использует разностную схему (8.1) с максимально возможным для двумерного случая шагом $\tau = \frac{h^2}{4}$.

В методе релаксации необходимо задать начальное приближение, т.е. значения функции во всех узлах области, а так же граничные условия.

Функция *relax* возвращает квадратную матрицу, в которой:

- 1) расположение элемента в матрице соответствует его положению внутри квадратной области,
- 2) это значение приближает решение в этой точке.

Эта функция использует метод релаксации для приближения к решению.

Функцию *relax* необходимо использовать, если известны значения искомой функции $u(x, y)$ на всех четырех сторонах квадратной области.

Аргументы:

a, b, c, d, e – квадратные матрицы одного и того же размера, содержащие коэффициенты дифференциального уравнения;

f – квадратная матрица, содержащая значения правой части уравнения в каждой точке внутри квадрата;

u – квадратная матрица, содержащая граничные значения функции на краях области, а также начальное приближение решения во внутренних точках области;

$rjac$ – параметр, управляющий сходимостью процесса релаксации. Он может быть в диапазоне от 0 до 1, но оптимальное значение зависит от деталей задачи.

Пример 1. $n := 2^4$ $i := 0..n$ $j := 0..n$.

Задаем правую часть уравнения Пуассона – два точечных источника:

$$M_{i,j} := 0, \quad M_{6,8} := 10, \quad M_{10,8} := -10.$$

Задаем значения параметров функции *relax*:

$$a_{i,j} := 1, \quad b := a, \quad c := a, \quad d := a, \quad f := M, \quad e := -4a.$$

Задаем граничные условия и начальное приближение – нули во всех внутренних точках области:

$$u_{i,j} := 0, \quad u_{i,n} := -1, \quad u_{0,j} := 1 - 2\frac{j}{n}, \quad u_{i,0} := 1, \quad u_{n,j} := 1 - 2\frac{j}{n}.$$

Находим решение $z := \text{relax}(a, b, c, d, e, f, u, 0.95)$ и представляем его графически (рис. 8.32, 8.33) в виде поверхности и линий уровней.

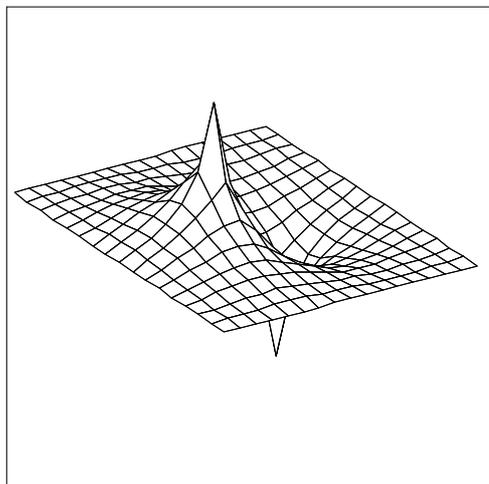


Рис. 8.32

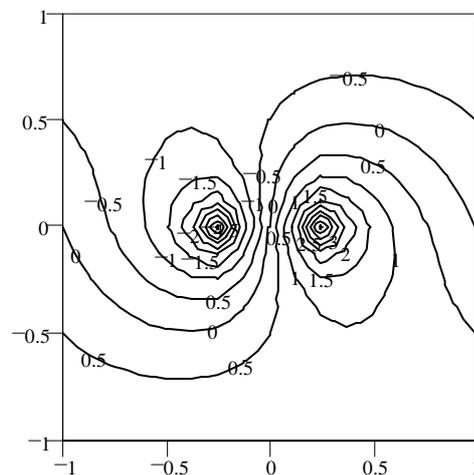


Рис. 8.33

Если граничные условия равны нулю на всех четырех сторонах квадрата, можно использовать функцию *multigrid*: $z := \text{multigrid}(M, 3)$.

9. ОСНОВЫ РАБОТЫ В AUTOCAD

9.1. Окно AutoCAD 2008

При первом запуске AutoCAD 2008 открывается окно Customer Involvement Program (Программа участия потребителя), которое является приглашением принять участие в программе улучшения продукта. Если вы согласитесь, программа будет сохранять и периодически отправлять компании Autodesk информацию о конфигурации системы и о том, как вы используете продукты Autodesk. Если вы не хотите принимать участие в программе, оставьте переключатель в положении *No, I don't want to participate at this time* (Нет, я не хочу принимать участие в данный момент) и снимите флажок *Remind me in 7 days* (Напомнить мне через 7 дней), чтобы это окно не появлялось на экране в течение недели. После закрытия этой панели перед вами появится рабочее окно AutoCAD 2008 (рис. 9.1). Рассмотрим основные элементы интерфейса программы.

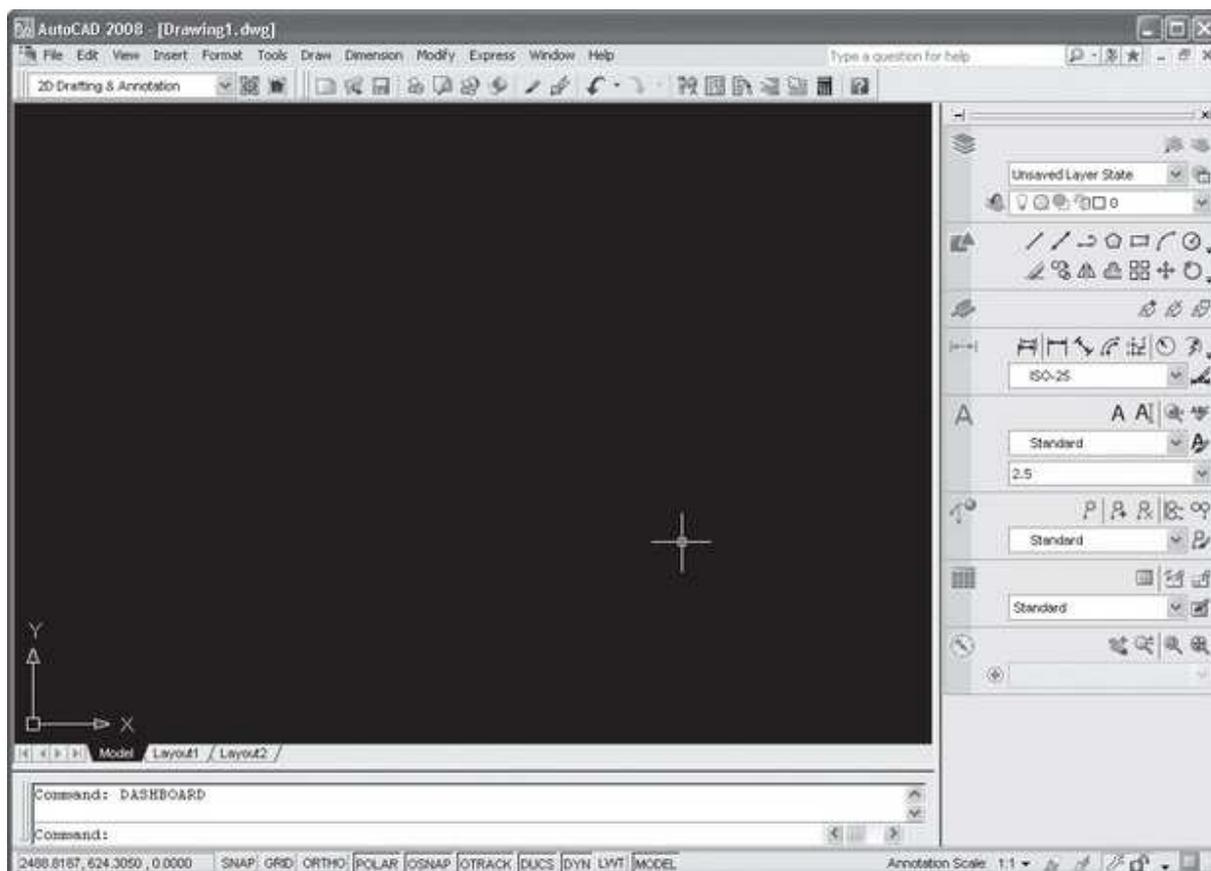


Рис. 9.1. Рабочее окно AutoCAD 2008

Основные элементы интерфейса

Центральная область рабочего окна (по умолчанию она черного цвета) называется графическим экраном. Это основная рабочая зона, в которой и выполняются все графические построения. Курсор, который перемещается в данной зоне, по виду напоминает мишень (рис. 9.2).

Размер перекрестий воображаемой мишени можно изменить в настройках или с помощью системной переменной. Чтобы курсор гарантированно попал в нужную точку графического объекта, существует определен-

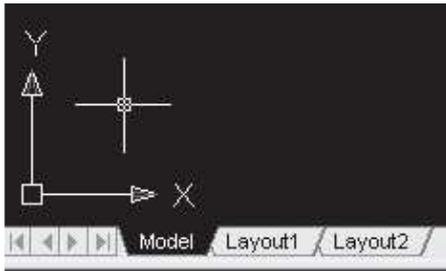


Рис. 9.2. Вид графического курсора

механизм. Так, при нахождении в области маркера точки курсор как бы приклеивается к ней, пока пользователь не сдвинет мышь в сторону. В это время характерная точка подсвечивается зеленым квадратиком, а в строке состояния можно увидеть координаты этой точки.

В нижней части графического экрана размещены вкладки *Model* (*Модель*), *Layout1* (*Лист1*) и *Layout2* (*Лист2*). По умолчанию при первом запуске AutoCAD активна вкладка *Model* (*Модель*) (рис. 9.3). При щелчке на вкладке *Model* (*Модель*) мы попадаем в пространство модели (двух- или трехмерное представление объекта), а при щелчке на вкладке *Layout* (*Лист*) переходим в пространство листа. Слева от вкладок расположены кнопки со стрелками, которые позволяют перемещаться по списку вкладок.



Рис. 9.3. Вкладки переключения между пространствами модели и листа

В левом нижнем углу графического экрана отображается пиктограмма осей координат (см. рис. 9.2). При необходимости этот значок можно настраивать. В AutoCAD, как правило, применяются декартовы координаты.

В правой и нижней частях графического экрана размещены полосы прокрутки. Пользуясь ими, можно передвигать рисунок по экрану (по умолчанию полосы прокрутки не отображаются, поэтому при первом запуске программы вы их не увидите).

Имеются также еще две полосы прокрутки, предназначенные для зоны командных строк. В верхней части окна AutoCAD расположена строка меню. Под ней находятся панели инструментов. С правой стороны окна размещена палитра Dashboard (Инструментальная панель), называемая также пультом управления или пультом инструментов. В нижней части экрана расположено окно командной строки (рис. 9.4).



Рис. 9.4. Окно командной строки

Под командной строкой находится панель с кнопками, которые отвечают за режимы рисования. Слева от этой панели расположен счетчик координат. В правой части строки состояния находится область, называемая лотком состояния. Здесь расположены значки, которые служат для индикации различных параметров программы и управления ими.

Настройка цвета фона рабочего окна

При работе с чертежами AutoCAD пользователю предоставляет возможность изменять цвет фона рабочего окна. По умолчанию рабочая область имеет черный цвет. Для изменения цвета необходимо выполнить следующие действия.

1. В меню *Tools (Сервис)* выберите пункт *Options (Параметры)*.
2. Откроется диалоговое окно, в котором нужно перейти на вкладку *Display (Экран)* (рис. 9.5).

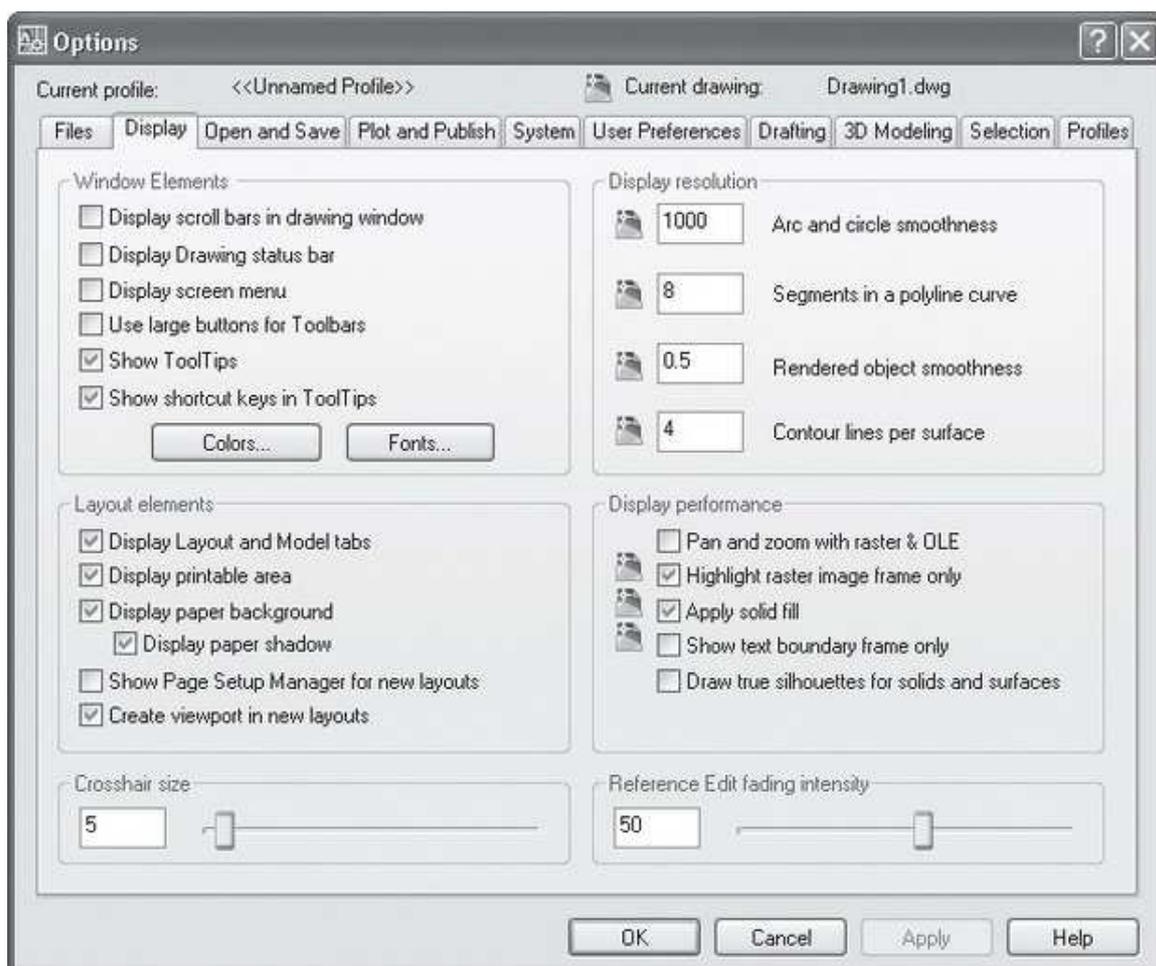


Рис. 9.5. Вкладка *Display (Экран)* диалогового окна *Options (Параметры)*

3. Вкладка *Display* (*Экран*) содержит шесть областей. В области *Window Elements* (*Элементы окна*) щелкните на кнопке *Colors* (*Цвета*). Появится диалоговое окно *Drawing Window Colors* (*Цвета окна чертежа*) (рис. 9.6).

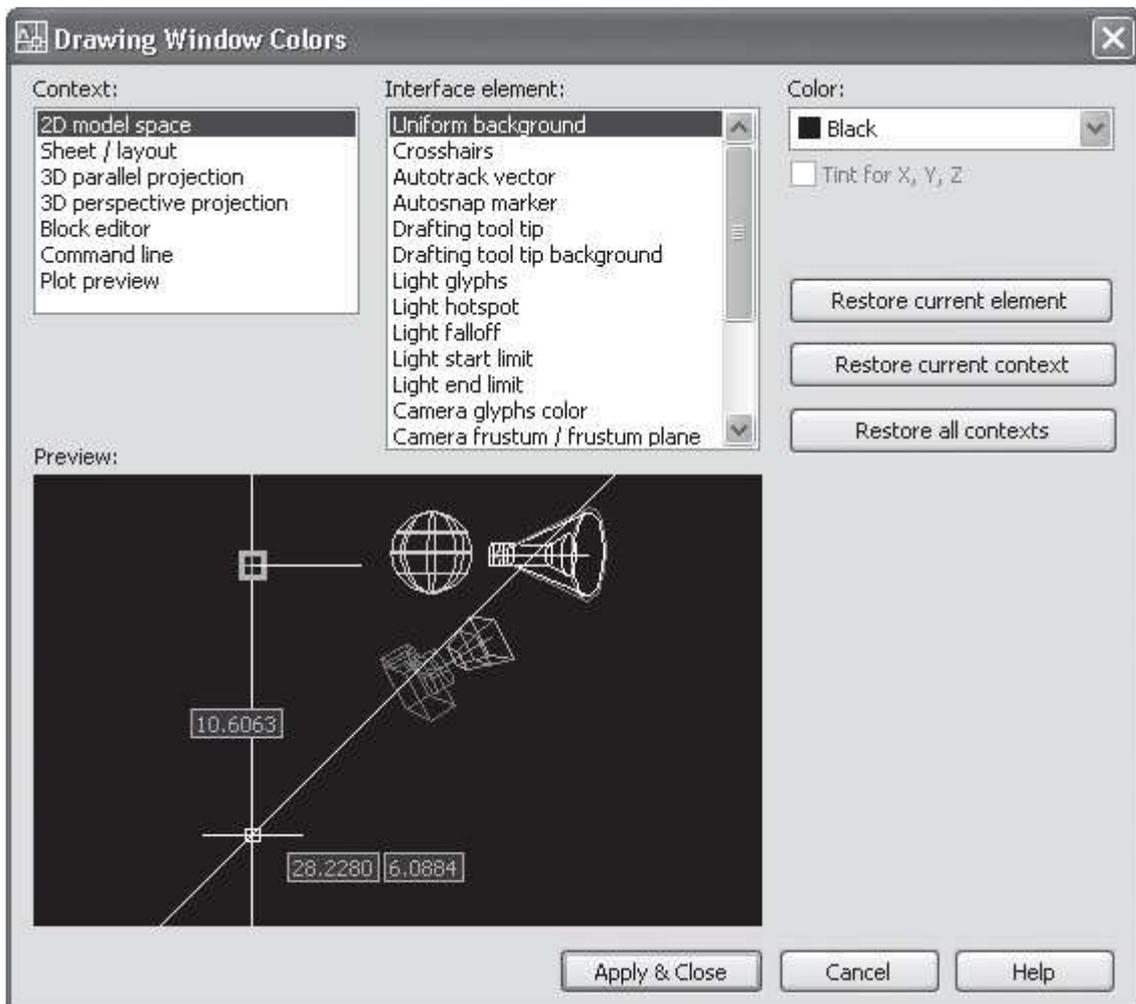


Рис. 9.6. Диалоговое окно настройки цветов программы

4. В списке *Context* (*Среда*) выберите пункт *2D model space* (*Пространство двумерной модели*), а в списке *Interface element* (*Элемент интерфейса*) – пункт *Uniform background* (*Общий фон*).

5. Из списка *Color* (*Цвет*) выберите нужный цвет и выйдите из диалогового окна *Drawing Window Colors* (*Цвета окна чертежа*), нажав кнопку *Apply & Close* (*Применить и закрыть*).

6. Теперь закройте диалоговое окно *Options* (*Параметры*), нажав кнопку ОК .

Рассмотрим основные элементы интерфейса программы, описанные выше, более подробно.

9.1.1. Строка меню

В верхней части окна AutoCAD расположена строка меню. Необходимо отметить, что в последних версиях программы в выпадающих меню напротив большинства пунктов появились соответствующие значки (рис. 9.7), что приятнее воспринимается визуально и определенно облегчает работу пользователей с недостаточно хорошим знанием английского языка.



Рис. 9.7. Пиктограммы в меню AutoCAD 2008

File (Файл)

Рассмотрим имеющиеся в программе пункты меню. Команды меню *File (Файл)* позволяют создать новый рисунок, новую подшивку, открыть ранее созданный чертеж, подшивку, установить настройки печати и распечатать чертеж. Здесь же можно получить доступ к таким полезным утилитам, как восстановление и очищение чертежей.

Edit (Правка)

Команды меню *Edit (Правка)* позволяют осуществлять стандартные для Windows операции работы с буфером обмена (вырезать, копировать, вставить), выполнять специальные вставки, а также производить поиск текста по чертежу.

View (Вид)

Команды меню *View (Вид)* позволяют управлять отображением объектов на экране. Это меню содержит такие пункты, как *Zoom (Масштабирование)*, *Pan (Панорамирование)*, *Render (Тонирование)* и т.д.

Отдельно необходимо отметить пункт *Clean Screen (Чистый экран)*, действие которого дублирует сочетание клавиш *Ctrl+0*. Эта функция пригодится для тех пользователей, чей монитор имеет небольшую диагональ. С помощью этого пункта пользователь может временно скрыть панели управления и заголовков окна AutoCAD, что позволит расширить рабочую зону графического экрана.

В версии AutoCAD 2008 в меню *View (Вид)* были добавлены новые команды работы с камерой и визуальными стилями.

Insert (Вставка)

Команды меню *Insert (Вставка)* позволяют вставлять в поле чертежа заранее созданные блоки и объекты, созданные в сторонних приложениях, а также импортировать различные файлы.

Format (Формат)

Команды меню *Format* (Формат) позволяют выполнять форматирование и предварительную настройку многих показателей (рис. 9.8).



Рис. 9.8. Меню *Format* (Формат)

Так, пункт *Layer* (Слой) дает возможность пользователю управлять слоями и их свойствами.

Новая функция *Layer States Manager* (Диспетчер состояний слоев) открывает диалоговое окно, в котором вы можете сохранять, редактировать (управляя такими параметрами, как цвет, вес линии и т.д.), переименовывать и удалять состояния слоев.

При выборе пункта *Layer tools* (Инструменты слоя) открывается дополнительное меню, которое содержит различные инструменты управления слоями.

Пункт *Color* (Цвет) устанавливает цвета для новых объектов. С помощью пункта *Linetype* (Тип линии) можно загрузить и установить типы линий, а пункт *Lineweight* (Вес линии) позволяет задать текущий вес линий.

С помощью пункта *Scale List* (Список масштабов) можно отредактировать наборы масштабов.

При выборе пунктов *Text Style* (Стиль текста) и *Dimension Style* (Размерный стиль) можно создавать, устанавливать и изменять текстовые и размерные стили соответственно.

Задавать и изменять стили таблиц можно с помощью пункта *Table Style* (Стиль таблицы).

Multileader Style (Стиль мультивыноски) позволяет устанавливать и изменять стили мультивыносок – новых объектов, появившихся в AutoCAD 2008.

В этом же меню можно выбрать соответствующий пункт для установки стилей печати – *Plot Style* (Стиль печати), задания режимов отображения точек – *Point Style* (Стиль точки), управления стилями мультиний – *Multiline Style* (Стиль мультиний).

При желании изменить название объекта можно воспользоваться пунктом *Rename* (Переименовать).

Tools (Сервис)

Команды меню *Tools (Сервис)* позволяют настраивать саму программу, вызывать на экран различные вспомогательные окна, а также загружать внешние приложения и управлять ими (рис. 9.9).

Это меню довольно сильно изменилось в данной версии программы. С помощью пункта *Workspaces (Рабочие пространства)* вы можете открыть меню управления рабочими пространствами.

Палитры программы сформированы в отдельное меню, открывающееся при выборе пункта *Palettes (Палитры)*. В новой версии это меню сильно увеличилось. В него были добавлены такие пункты, как *External References (Внешние ссылки)* – палитра управления внешними связями и *Markup Set Manager (Диспетчер пометок)* – палитра управления пометками, а также палитры для работы с трехмерными объектами и базами данных.

Рассмотрим некоторые команды меню *Palettes (Палитры)*.

Палитра *Dashboard (Инструментальная панель)* содержит контрольные панели с кнопками и элементами управления.

С помощью пункта *Properties (Свойства)* пользователь может управлять свойствами объектов.

Воспользовавшись пунктом *Tool Palettes (Инструментальные палитры)*, можно вызывать на экран и скрывать окно инструментальных палитр.

При выборе пункта *QuickCalc (Быстрый подсчет)* открывается палитра инженерного калькулятора (ее также можно вызвать сочетанием клавиш `Ctrl+8`).

Пункт *Sheet Set Manager (Диспетчер подшивок)* выводит на экран или скрывает диспетчер подшивок.

DesignCenter (Центр управления) управляет содержимым рисунков.



Рис. 9.9. Меню *Tools (Сервис)*

Воспользовавшись пунктом *Command Line* (Командная строка), можно убрать с экрана командную строку, расположенную в нижней части окна программы. Для быстрого вызова этой команды можно воспользоваться комбинацией клавиш **Ctrl+9**.

В меню *Tools* (Сервис) есть такие пункты, как *Spelling* (Орфография), позволяющий проверить орфографию в выбранном пользователем тексте; *Quick Select* (Быстрый выбор) – для быстрого создания групп объектов; *Draw Order* (Порядок следования), который предоставляет возможность переносить объекты на заданный план.

В этом же меню находится группа команд для работы с макросами, создаваемыми на языке программирования VBA.



Рис. 9.10. Меню вызова команд рисования

С помощью пункта *Drafting Settings* (Настройки рисования) можно устанавливать необходимые настройки режимов рисования, а пункт *Options* (Параметры) вызывает одноименное диалоговое окно для настройки параметров программы.

Draw (Рисование)

Команды меню *Draw* (Рисование) (рис. 9.10) позволяют создавать различные примитивные объекты: *Line* (Отрезок), *Ray* (Луч), *Construction Line* (Прямая), *Multiline* (Мультилиния), *Polyline* (Полилиния), *3D Polyline* (3D Полилиния), *Polygon* (Многоугольник), *Rectangle* (Прямоугольник), *Helix* (Спираль), *Arc* (Дуга), *Circle* (Круг), *Donut* (Кольцо), *Spline* (Слайн), *Ellipse* (Эллипс).

С помощью пункта *Modeling* (Моделирование) можно создавать различные тела: *Polysoid*, *Box* (Ящик), *Wedge* (Клин), *Cone* (Конус), *Sphere* (Шар), *Cylinder* (Цилиндр), *Torus* (Тор), *Pyramid* (Пирамида), а используя пункт *Wipeout* (Маскировка) – маскирующие объекты.

В этом же меню присутствуют пункты, позволяющие создавать блоки, – *Block* (Блок), таблицы – *Table* (Таблица), точки – *Point* (Точка), а также вводить текст – *Text* (Текст).

При выборе пункта *Hatch* (Штрих) можно использовать штриховку для заполнения ею выбранного объекта.

Команда *Boundary* (*Контур*) служит для создания области или полилинии из замкнутых объектов.

Dimension (Размер)

Команды меню *Dimension* (*Размер*) позволяют наносить на поле чертежа необходимые размеры, выноски, создавать и изменять размерные стили (рис. 9.11). Так, с помощью пунктов *Linear* (*Линейный*), *Aligned* (*Параллельный*), *Ordinate* (*Ординатный*) и *Angular* (*Угловой*) можно наносить линейные, параллельные, ординатные и угловые размеры соответственно.

Пункт *Arc Length* (*Длина дуги*) служит для простановки длины дуги, а *Jogged* (*Ломаный*) – для нанесения размера с изломом.

В AutoCAD 2008 появились такие новые пункты, как *Dimension Space* (*Расстояние между размерами*) и *Dimension Break* (*Разрыв размера*), позволяющие устанавливать расстояние между размерами и разрывать размерные линии при пересечении.

Пункт *Multileader* (*Мультивыноска*) позволяет добавлять мультивыноски, *Inspection* (*Контроль*) – вставлять связанную с размером информацию, помещенную в рамку, а *Jogged Linear* (*Искривленная линия*) – искривлять размерную линию.

С помощью параметра *Oblique* (*Наклонить*) можно наклонять выносные линии линейных размеров, а пункт *Dimension Style* (*Размерный стиль*) позволяет создавать новые и изменять существующие размерные стили.

Modify (Редактирование)

Команды меню *Modify* (*Редактирование*) (рис. 9.12) позволяют выполнять простейшие действия над объектами: масштабирование, обрезку, удлинение и многое другое.

Меню *Modify* (*Редактирование*) среди прочих содержит следующие пункты:

Erase (*Стереть*) – позволяет стирать объекты чертежа;

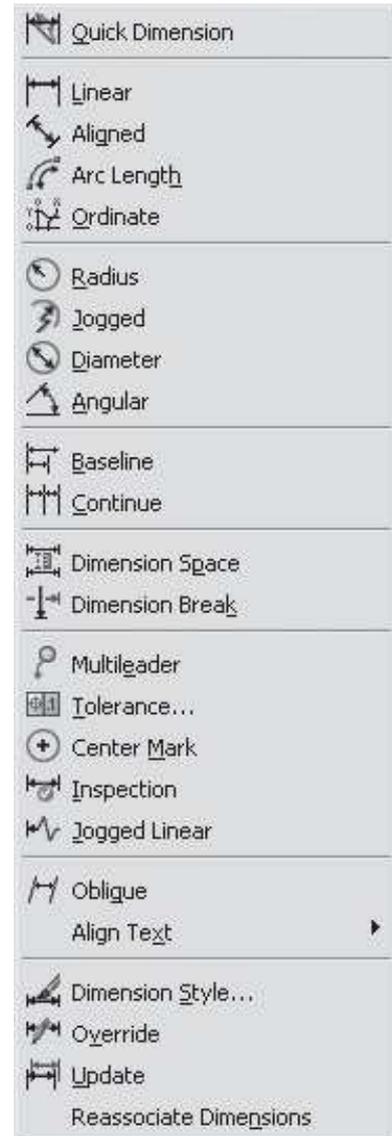


Рис. 9.11. Меню *Dimension* (*Размер*)

Copy (Копировать) – позволяет копировать объекты;
Mirror (Зеркало) – создает зеркальные отображения объектов;
Offset (Подобие) – с его помощью можно создавать подобные объекты;



Рис. 9. 12. Меню вызова команд редактирования

новый пункт – *Annotative Object Scale (Масштаб аннотационного объекта)*, открывающий меню управления масштабом объектов.

Window (Окно)

Меню *Window (Окно)* позволяет ориентироваться при одновременной работе с несколькими чертежами и поддерживать многодокументный режим работы.

Array (Массив) – позволяет упорядочивать много объектов в единую структуру;

Move (Переместить) – перемещает объекты;

Rotate (Повернуть) – позволяет повернуть объект вокруг базовой точки;

Scale (Масштабировать) – задает масштаб объектов;

Stretch (Растянуть) – позволяет растягивать объекты;

Lengthen (Увеличить) – увеличивает длину или величину центральных углов определенных объектов;

Trim (Обрезать) – обрезает объект по режущим кромкам;

Extend (Удлинить) – удлиняет объекты до кромки другого объекта;

Break (Разорвать) – позволяет произвести разрыв объекта между двумя точками;

Join (Объединить) – вызывает инструмент, позволяющий объединять объекты в форму единого, неразбиваемого объекта;

Chamfer (Фаска) – снимает фаски с объектов;

Fillet (Скругление) – округляет кромки объекта.

Change Space (Изменить пространство) – позволяет перемещать объекты из пространства модели в пространство листа и наоборот.

В рассматриваемой нами версии AutoCAD 2008 в меню *Modify (Редактирование)* добавлен

В рассматриваемом меню есть такой пункт, как *Lock Location (Закрепить положение)*, который блокирует позицию и размер панелей инструментов, а также окон.

Кроме того, с помощью данного меню можно закрывать окна и управлять их размещением на экране.

Help (Справка)

Меню *Help (Справка)* содержит разделы, помогающие пользоваться справочной системой, не прерывая работы с основным документом. Система справки хорошо сконструирована и содержит множество полезной информации.

Справа от строки меню расположено поле ввода, а также кнопки *InfoCenter (Информационный центр)*, *Communication Center (Коммуникационный центр)* и *Favorites (Избранное)*, которые служат для получения ответов на вопросы с помощью Интернета.

9.1.2. Панели инструментов

По умолчанию под строкой меню расположены две панели инструментов. Они содержат кнопки, каждая из которых выполняет определенные функции. Быстро узнать о предназначении каждой кнопки можно, подведя к ней указатель мыши и посмотрев на всплывающую подсказку, в которой, возможно, будет приведено сочетание клавиш для быстрого вызова команды.

Примечание. Некоторые кнопки на панелях инструментов дублируют пункты меню, с которыми вы познакомились выше. Пользователь имеет возможность выбрать, какой из этих способов доступа к командам для него удобнее.

Плавающие панели инструментов

Для удобства использования панели инструментов могут быть плавающими или закрепленными, что дает возможность размещать их в любом месте экрана на усмотрение пользователя. За эту возможность панели инструментов и называют плавающими.

Переместить панель инструментов можно, подведя указатель мыши к границе нужной панели инструментов и, удерживая левую кнопку нажатой, передвинув ее в необходимое место экрана, после чего отпустить кнопку. Вернуть панель инструментов на исходное место можно тем же способом.

Плавающим панелям инструментов можно придавать любую форму путем стягивания или растягивания их за углы.

Зачастую очень удобно располагать плавающую панель инструментов вплотную к краю экрана, но сделать это достаточно сложно. Дело в том, что при подтягивании панели инструментов к границе окна про-

граммы она автоматически становится закрепленной («прилипает» к краям экрана). Между тем решить проблему очень просто – достаточно при манипуляции указателем мыши во время перемещения панелей инструментов удерживать нажатой клавишу *Ctrl*.

Standard Annotation (Стандартная аннотационная)

Эта панель инструментов появилась в AutoCAD 2008 и содержит кнопки, предназначенные для вызова базовых команд: *QNew* (Создание), *Open* (Открытие), *Save* (Сохранение) файла, *Plot Preview* (Просмотр проекта перед печатью) и *Plot* (Вывод его на печать), *Undo* (Отмена) и *Redo* (Возврат) действий, а также вызова некоторых палитр программы (рис. 9.13).



Рис. 9.13. Панель инструментов *Standard Annotation* (Стандартная аннотационная)

Панель инструментов *Standard Annotation* (Стандартная аннотационная) похожа на панель *Standard* (Стандартная), из которой исключены кнопки вызова команд навигации по чертежу, а также копирования и вставки. Эта панель рассчитана на использование совместно с пультом управления.

Workspaces (Рабочие пространства)

Панель инструментов *Workspaces* (Рабочие пространства) содержит раскрывающийся список, в котором можно выбирать рабочие пространства, настраивать и сохранять их. Кнопка *Workspace Settings* (Настройки рабочего пространства) открывает диалоговое окно, служащее для изменения настроек выбора рабочих пространств. Кнопка *My Workspace* (Мое рабочее пространство) делает текущее рабочее пространство используемым по умолчанию.

Управление панелями инструментов

Пользователь может отображать и скрывать как эти, так и любые другие панели инструментов. Для этого нужно воспользоваться пунктом меню *View* → *Toolbars* (Вид → Панели инструментов). После выбора этого пункта откроется диалоговое окно, где следует выбрать панели инструментов, которые необходимо отобразить или скрыть.

Более простой способ удаления с экрана/отображения панелей инструментов – щелчок правой кнопкой мыши на одной из них. После выполнения этого действия, появится меню со списком всех доступных панелей.

Напротив названий панелей, которые в данный момент отображены, установлены флажки. Щелчок на названии такой панели скроет ее с экрана. Щелчок на названии панели, которая была скрыта, приведет к отображению ее на экране.

Добавление кнопок на панели инструментов

Для того чтобы изменить состав панелей инструментов и положение кнопок на них, выполните описанные ниже действия.

1. Выполните команду *Tools*→*Customize*→*Interface* (*Сервис*→*Настройка*→*Интерфейс*) или щелкните правой кнопкой мыши на любой панели инструментов и выберите пункт *Customize* (*Настройка*).

2. В области *Customization in All CUI Files* (*Настройка всех файлов*) открывшегося окна раскройте список *Toolbars* (*Панели инструментов*).

3. В области *Command List* (*Список команд*), содержащей все команды программы, выберите команду, которую вы хотите добавить на панель инструментов. Для того чтобы было проще найти необходимую команду, раскройте список и выберите ту категорию, к которой относится команда (рис. 9.14).

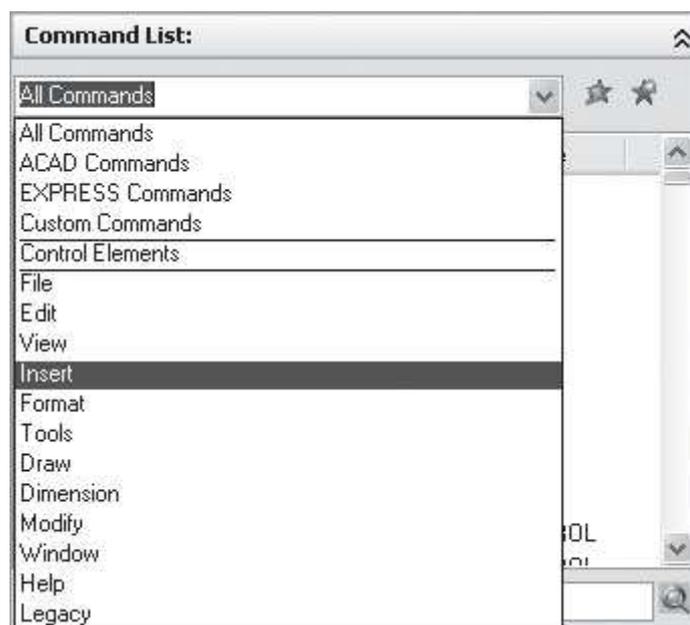


Рис. 9.14. Раскрытый список категорий

а) для добавления кнопки на панель инструментов щелкните на ней левой кнопкой мыши и, не отпуская ее, перетащите в список панелей инструментов, в область *Customizations in All CUI Files* (*Настройка всех файлов*). Когда указатель мыши достигнет панели инструментов, на которую необходимо добавить команду, наведите его на значок рядом с названием панели инструментов – список команд вы-

бранной панели раскроется. Подумайте, между какими кнопками вы хотите вставить вашу команду, переместите указатель мыши в выбранное место и отпустите кнопку мыши (рис. 9.15).

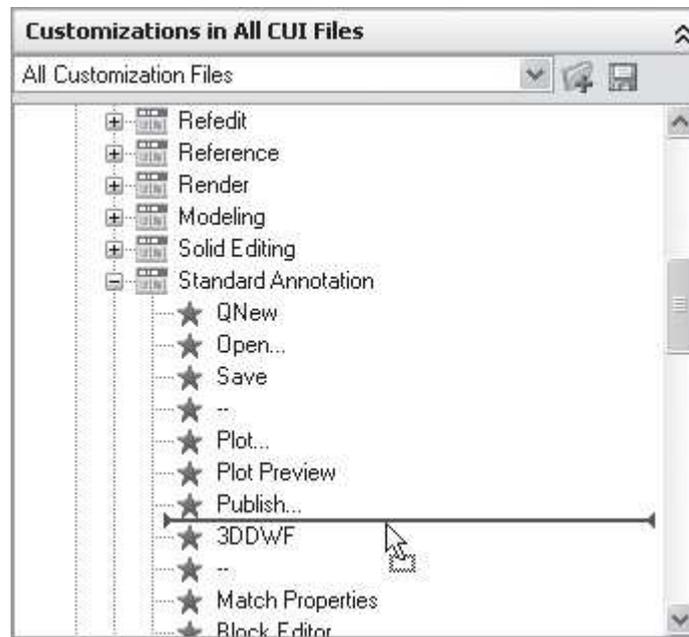


Рис. 9.15. Добавление новой кнопки на панель инструментов

б) чтобы проверить, была ли команда добавлена на панель инструментов, выделите название панели. В области *Toolbar Preview (Предварительный просмотр панели инструментов)* вы увидите, как выглядит панель инструментов после произведенных изменений. Найдите добавленную вами кнопку.

в) Для изменения положения кнопки на панели инструментов выберите кнопку в области *Customizations in All CUI Files (Настройка всех файлов)* и перетащите ее на другую позицию. Указатель в виде горизонтальной линии обозначит место, куда будет помещена кнопка.

г) Чтобы удалить кнопку, выберите ее в области *Customizations in All CUI Files (Настройка всех файлов)*, щелкните на ней правой кнопкой мыши и выберите команду *Remove (Удалить)*.

4. Нажмите кнопку *Apply (Применить)*, чтобы сохранить изменения на панелях инструментов.

Изменение внешнего вида кнопок на панелях инструментов

Для того чтобы изменять внешний вид кнопок на панелях инструментов, выделите нужную кнопку в области *Customizations in All CUI Files (Настройка всех файлов)* или *Command List (Список команд)* окна настроек пользовательского интерфейса и в области *Button Image (Значок кнопки)* выберите один из предложенных значков (рис. 9.16).

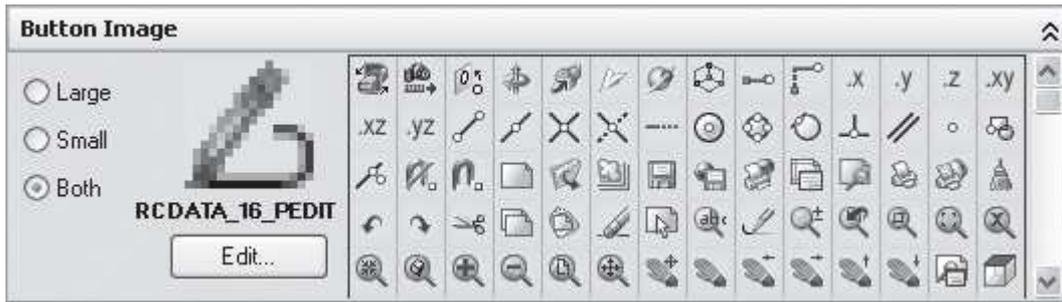


Рис. 9.16. Выбор значка для кнопки в области *Button Image* (Значок кнопки)

Нажмите кнопку *Edit* (Правка), чтобы нарисовать кнопку вручную. Используя возможности окна *Button Editor* (Редактор кнопок) (рис. 9.17), которое открывается при нажатии этой кнопки, нарисуйте значок.

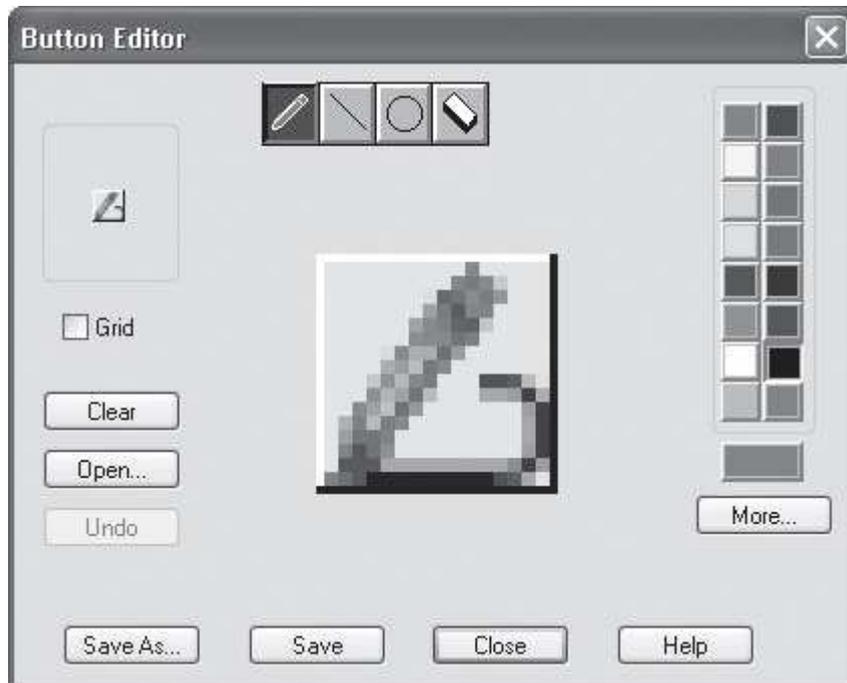


Рис. 9.17. Окно *Button Editor* (Редактор кнопок)

Вы можете использовать следующие инструменты, настройки и кнопки.

Карандаш – для рисования по клеткам. Цвет, которым будет рисовать инструмент, определяется щелчком на цветном квадратике в правой части окна.

Линия – для создания рисунка в виде линии.

Круг – для создания рисунка в виде круга.

Ластик – для удаления рисунка в выбранных клетках.

Grid (Сетка) – для отображения клеток на рисунке.

Clear (Очистить) – для удаления рисунка во всех клетках.

Open (Открыть) – для использования в качестве значка любого файла в формате BMP, RLE или DIB.

Undo (Отменить) – для отмены последнего действия.

Save (Сохранить) – для сохранения созданного значка.

Save As (Сохранить как) – для сохранения созданного значка под другим именем. При этом исходный значок, который редактировался, будет сохранен.

После закрытия окна *Button Editor (Редактор кнопок)* вы сможете выбрать созданный значок из списка и использовать его для оформления кнопки.

Создание пользовательской панели

Для создания собственной панели инструментов выполните описанные ниже действия.

1. Выполните команду *Tools→Customize→Interface (Сервис→Настройка→Интерфейс)* или щелкните правой кнопкой мыши на любой панели инструментов и выберите пункт *Customize (Настройка)*.

2. В области *Customizations in All CUI Files (Настройка всех файлов)* открывшегося окна выберите пункт *Toolbars (Панели инструментов)*. Щелкните на нем правой кнопкой мыши и выберите строку *New Toolbar (Новая панель инструментов)* (рис. 9.18).

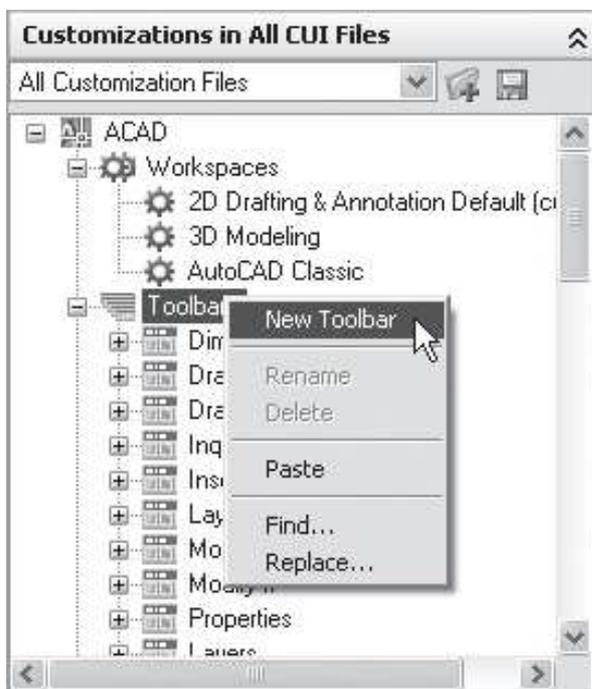


Рис. 9.18. Создание панели инструментов

3. Введите название панели инструментов и установите ее основные настройки в области *Properties (Свойства)*. Выберите тип панели: *Floating (Плавающая)*, *Top (Размещенная сверху)*, *Bottom (Размещенная снизу)*, *Left (Размещенная слева)* или *Right (Размещенная справа)*. Определитесь, будет ли панель отображаться по умолчанию. Введите описание панели в поле *Description (Описание)*.

4. В области *Command List (Список команд)* выберите команды, которые вы хотите поместить на панель, и добавьте их так, как было описано выше.

5. Нажмите кнопку ОК, чтобы добавить новую панель (рис. 9.19).

При необходимости скройте ненужные панели. Для этого щелкните правой кнопкой мыши на любом месте одной из панелей инструментов и в появившемся списке снимите флажок для панелей, которые отображаются.



Рис. 9.19. Созданная пользовательская панель

9.1.3. Пульт управления

Пульт инструментов (рис. 9.20) является мощным средством управления. Этот инструмент программы AutoCAD, представляет собой специальную палитру, на которой отображаются кнопки и элементы управления, связанные с задачей ориентированным рабочим пространством. Выполняя функции сразу нескольких панелей инструментов, он позволяет значительно увеличить площадь графической области.

Примечание. Введение пульта инструментов несколько изменило привычный вид окна программы AutoCAD – вместо графической области, окруженной панелями инструментов, мы видим пульт управления и две панели инструментов под строкой меню. Пользователи, ранее работавшие с программой AutoCAD и привыкшие к прежнему интерфейсу, легко могут вернуть ей классический вид, выбрав в раскрывающемся списке панели инструментов *Workspaces (Рабочие пространства)* пункт *AutoCAD Classic (Классический AutoCAD)*.

По умолчанию на пульте управления отображаются следующие панели.

Layers (Слои)

Панель *Layers (Слои)* пульта управления содержит те же кнопки, что и одноименная панель инструментов, включая новый инструмент *Layer States Manager (Диспетчер состояний слоев)*. Раскрывающийся список *Select a layer state (Выбрать состояние слоя)* позволяет управлять доступными состояниями слоя.

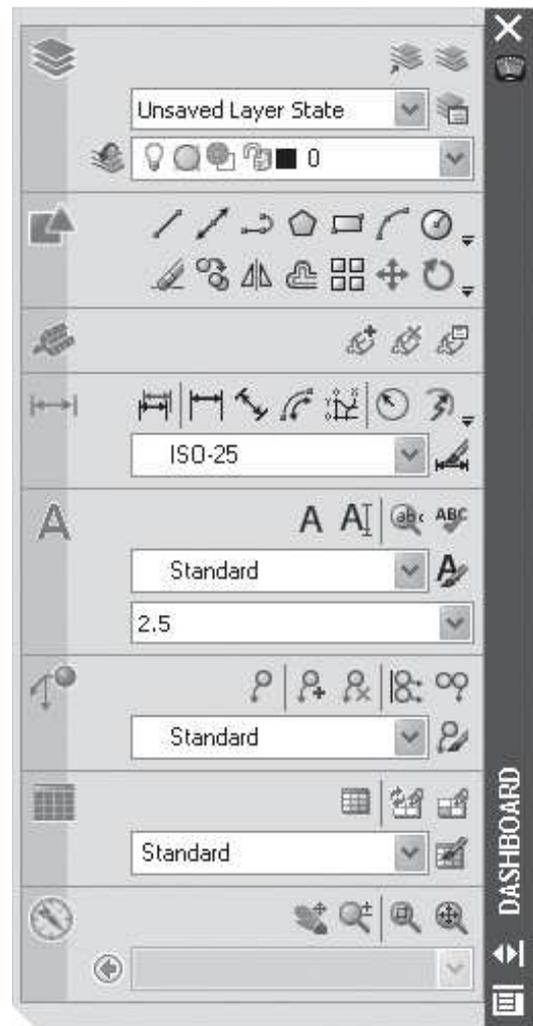


Рис. 9.20. Пульт управления

При щелчке на маленьких стрелках, появляющихся в левом нижнем углу при подведении к ним указателя мыши, открывается доступ к некоторым новым инструментам.

Кнопка *Layer Isolate* (*Изолирование слоя*) позволяет скрыть или заблокировать все слои, кроме тех, которые содержат выделенные объекты. Кнопка *Layer Unisolate* (*Отмена изолирования слоя*), напротив, позволяет восстанавливать слои, изолированные предыдущей командой. Кнопка *Toggle locked layer fading* (*Переключение прозрачности заблокированного слоя*) разрешает и запрещает прозрачность заблокированных слоев. Если прозрачность включена, ее уровень может быть установлен с помощью ползункового регулятора или поля ввода.

2D Draw (Двухмерное рисование)

Панель *2D Draw* (*Двухмерное рисование*) пульта управления содержит кнопки двух панелей инструментов: *Draw* (*Рисование*) и *Modify* (*Редактирование*), которые не изменились по сравнению с предыдущей версией программы.

Кнопки, служащие для создания объектов, позволяют строить различные примитивные объекты: *Line* (*Отрезок*), *Construction Line* (*Прямая*), *Polyline* (*Полилиния*), *Polygon* (*Многоугольник*), *Rectangle* (*Прямоугольник*), *Arc* (*Дуга*), *Circle* (*Круг*), *Spline* (*Сплайн*), *Ellipse* (*Эллипс*), *Ellipse Arc* (*Эллиптическая дуга*), а также, используя кнопку *Revision Cloud* (*Облако*), полилинии из дуговых сегментов в форме облака.

Здесь же присутствуют кнопки, с помощью которых можно вставлять в текущий рисунок блоки или другие рисунки – *Insert Block* (*Вставка блока*), создавать блоки – *Make Block* (*Создать блок*), таблицы – *Table* (*Таблица*), строить точки – *Point* (*Точка*), а также создавать многострочный текст – *Multiline Text* (*Многострочный текст*).

При выборе кнопки *Hatch* (*Штрих*) можно использовать штриховку для заполнения ею выбранного объекта. Применять заливку позволяет кнопка *Gradient* (*Градиент*). С помощью кнопки *Region* (*Область*) можно преобразовывать объект, ограничивающий замкнутое пространство, в объект-область.

Кнопки, предназначенные для редактирования объектов, позволяют перемещать их, стирать, растягивать и т.д. Нажав соответствующую кнопку, можно осуществить следующие операции: *Erase* (*Стереть*), *Copy* (*Копировать*), *Mirror* (*Зеркало*), *Offset* (*Подобие*), *Array* (*Массив*), *Move* (*Переместить*), *Rotate* (*Повернуть*), *Scale* (*Масштабировать*), *Stretch* (*Растянуть*), *Trim* (*Обрезать*), *Extend* (*Удлинить*), *Break* (*Разорвать*), *Chamfer* (*Фаска*), *Fillet* (*Скругление*) и пр.

Annotation Scaling (Масштабирование аннотационных объектов)

Аннотациями в программе AutoCAD называют текст, размеры, димензии, а также другие типы пояснительной информации, добавляемой к чертежу. Параметры аннотаций хранятся в файле чертежа. При создании аннотационных объектов можно заранее указать их масштаб, и они автоматически отобразятся на чертеже в нужном размере. Эта возможность появилась в AutoCAD 2008.

Кнопки панели *Annotation Scaling (Масштабирование аннотационных объектов)* позволяют добавлять и удалять текущий масштаб аннотационных объектов, а также добавлять/удалять существующие.

Dimensions (Размеры)

Кнопки данной панели дублируют пункты меню *Dimension (Размер)* и содержат большую часть его команд. Раскрывающийся список позволяет выбрать, на каком стиле будет основано построение размера: аннотационном или соответствующем стандарту ISO-25.

Text (Текст)

Команды панели *Text (Текст)* позволяют работать с текстом. С их помощью вы можете осуществлять создание, редактирование, поиск текста, проверку орфографии и другие операции с текстовыми объектами.

Multileaders (Мультивыноски)

Панель *Multileaders (Мультивыноски)* служит для создания различных выносок и содержит такие кнопки, как *Multileader (Мультивыноска)*, *Add Leader (Добавить выноску)*, *Remove Leader (Удалить выноску)*, *Align Multileaders (Выровнять мультивыноски)*. Кнопка *Collect Multileaders (Создать коллекцию мультивыносок)* позволяет объединить мультивыноски в группу.

При нажатии кнопки *Multileader Style Manager (Диспетчер стилей мультивыноски)* на экране появляется одноименное окно, позволяющее создавать и редактировать стили мультивыносок.

Кнопка, расположенная в левом верхнем углу панели открывает палитру, содержащую образцы различных выносок.

Такой объект, как мультивыноска, впервые появился в AutoCAD 2008, данная панель не имеет аналога среди меню или панелей инструментов.

Tables (Таблицы)

Эта панель содержит существовавшие ранее кнопки *Table (Таблица)* и *Table Style (Стиль таблицы)*, а также кнопки *Update Data Links (Обновить связанные данные)* и *Data Link Manager (Диспетчер связанных данных)*, позволяющие обновлять данные и создавать связи с файлами элек-

тронных таблиц программы Excel. Такая возможность также является новшеством программы AutoCAD 2008.

2D Navigate (Двухмерная навигация)

Данная панель служит для навигации в двухмерном пространстве и содержит такие кнопки, как *Pan Realtime* (Панорамирование в реальном времени), *Realtime Zoom* (Масштабирование в реальном времени), *Window* (Окно) и *Extents* (Границы).

Как и все элементы пользовательского интерфейса программы AutoCAD, пульт управления может быть настроен в соответствии с вашими нуждами. Чтобы указать, какие панели должны быть отображены, щелкните правой кнопкой мыши на свободной части палитры и в открывшемся меню выберите пункт *Control panels* (Панели управления). Установите флажки напротив названий нужных панелей и снимите напротив тех, которые вам не нужны.

Команды *Customize panel* (Настройка панели) и *Customize commands* (Настройка команд) открывают диалоговое окно *Customize User Interface* (Настройка пользовательского интерфейса), с помощью которого вы можете управлять как содержимым панелей, так и свойствами команд, соответствующих кнопкам панелей.

9.1.4. Командная строка и динамический ввод

Командная строка

Как уже отмечалось ранее, AutoCAD – это программа, которая существует около 20 лет. В связи с этим многие элементы, которые были актуальны в прошлом, сейчас уже частично или полностью утратили свою значимость, но сохранились в интерфейсе программы. В качестве примера можно привести командную строку, которая была востребована во времена, когда компьютер не имел мыши. Постепенно она утрачивает свою актуальность, но все же отказаться от нее полностью пока нельзя.

В нижней области экрана находится окно командной строки (см. рис. 9.4). В этой области отображаются вводимые пользователем команды для AutoCAD. Это все еще один из основных элементов интерфейса. В процессе работы в среде AutoCAD вы будете постоянно обращаться к командной строке. Командная строка представляет собой средство диалога пользователя и программы. Когда пользователь вызывает команду (из главного меню программы либо щелкая на соответствующих кнопках панелей инструментов), в командной строке автоматически отображается название вызываемой команды.

Практически у каждой команды есть определенные параметры или предназначенная для ввода дополнительная информация (например, координаты точек). В этом случае пользователь должен прочитать вопрос, который появляется в командной строке, и ответить на него. До этого момента нельзя начинать выполнять новую команду и выходить из программы, иначе данная команда будет прервана.

С небольшой областью командной строки работать не всегда удобно, для удобства пользования ее необходимо редактировать. Увеличить видимую область можно стандартным путем с помощью перетаскивания указателем мыши, но при этом увеличение будет происходить за счет уменьшения области графического экрана, что почти всегда неудобно для пользователя.

Для решения проблемы можно воспользоваться горячей клавишей F2. После ее нажатия открывается *AutoCAD Text Window* (Текстовое окно AutoCAD) (рис. 9.21), подобное окну командной строки. В нем также можно вводить команды и просматривать историю их ввода, при этом окно можно растягивать и сворачивать как обычное приложение Windows. Для закрытия *AutoCAD Text Window* (Текстовое окно AutoCAD) достаточно повторно нажать клавишу F2.

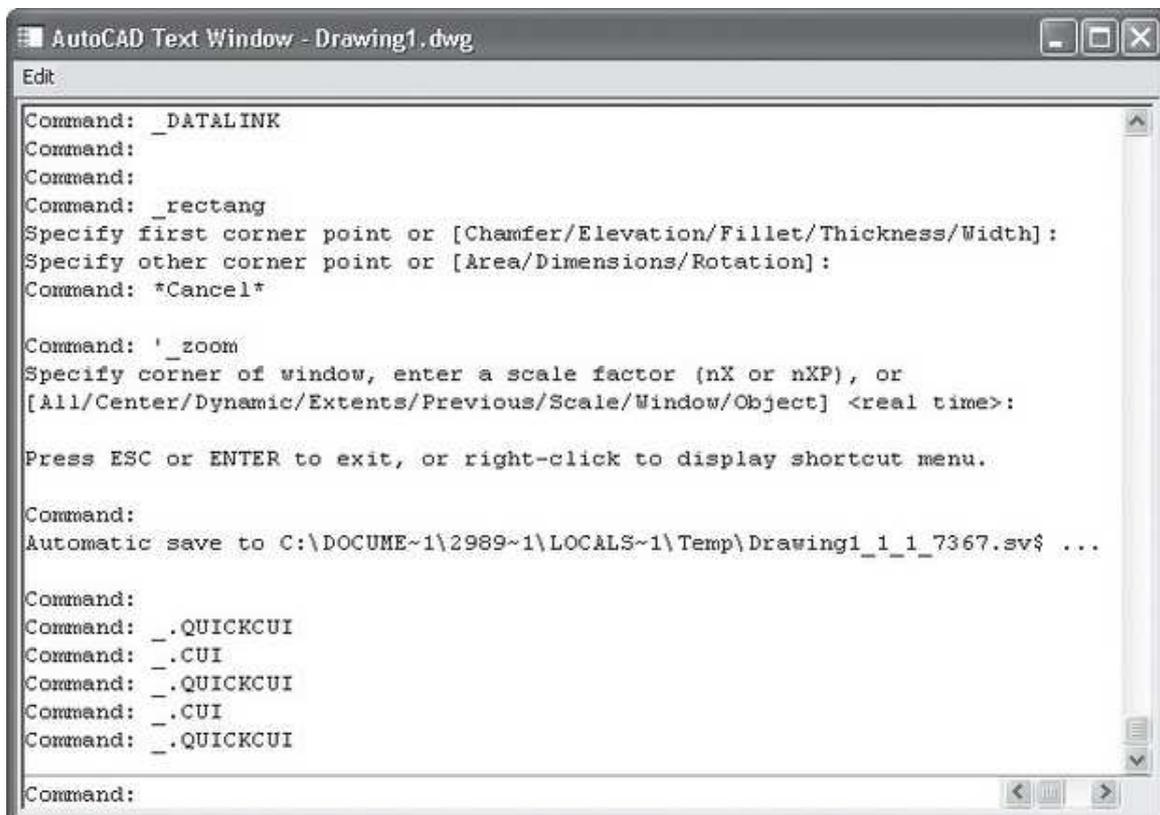


Рис. 9. 21. Вид *AutoCAD Text Window* (Текстовое окно AutoCAD)

Динамический ввод

В версии AutoCAD 2008 сообщения командной строки дублируются прямо возле указателя мыши. Также при построении объектов в области указателя отображаются динамические размеры строящегося объекта, которые можно здесь же и редактировать (рис. 9.22).

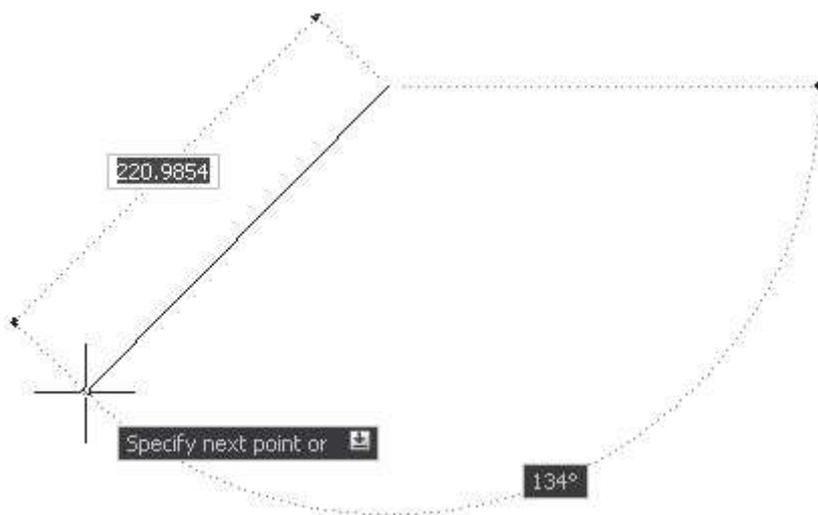


Рис. 9.22. Отображение динамических размеров

Для вызова списка параметров текущей команды можно просто нажать на клавиатуре клавишу _ (рис. 9.23).

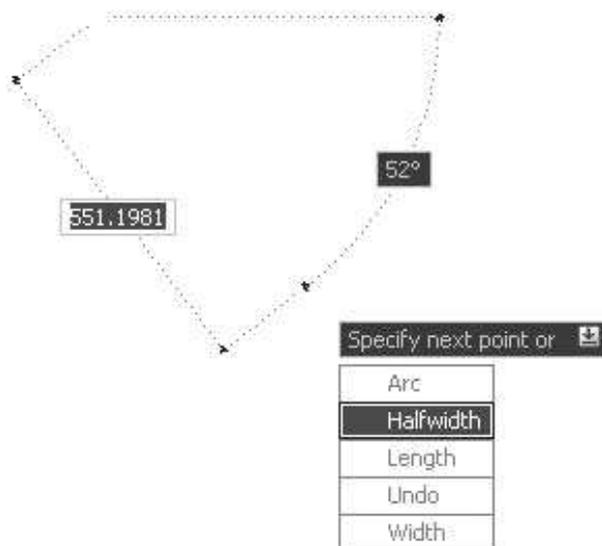


Рис. 9.23. Список параметров примитива *Polyline* (Полилиния)

Совет. Для отключения динамического ввода нужно нажать кнопку DYN (ДИН), расположенную внизу окна программы, в строке состояния.

Команда разработчиков Autodesk, давая понять пользователям, что с появлением возможности динамического ввода наличие на экране командной строки необязательно, добавила в меню Tools (Сервис) пункт Command Line (Командная строка), после выбора которого командную строку можно убрать с рабочего экрана. Для быстрого вызова этой команды можно воспользоваться сочетанием клавиш `Ctrl+9`.

9.1.5. Строка состояния

Внизу окна программы, под областью командной строки находится строка состояния. В этой строке размещены кнопки режимов рисования и счетчик координат.

Режимы рисования

Как было сказано выше, под командной строкой находится панель с кнопками, которые отвечают за режимы рисования: SNAP (ШАГ), GRID (СЕТКА), ORTHO (ОРТО), POLAR (ОТС-ПОЛЯР), OSNAP (ПРИВЯЗКА), OTRACK (ОТС-ОБЪЕКТ), DUCS (ДПСК), DYN (ДИН), LW T (ВЕС), MODEL (МОДЕЛЬ) (рис. 9.24).



Рис. 9.24. Кнопки режимов рисования

Режим считается включенным, если нажата соответствующая ему кнопка. Всего этих кнопок на панели по умолчанию десять, но любую из них можно либо убрать, либо добавить на панель. Для этого можно воспользоваться меню строки состояния, которое обозначено кнопкой в правой части строки состояния и появляется при щелчке на этой кнопке.

SNAP (ШАГ)

Кнопка SNAP (ШАГ) задает параметры шаговой привязки, т.е. управляет режимом привязки к точкам сетки с определенным шагом. Этой кнопке соответствует функциональная клавиша F9.

GRID (СЕТКА)

Кнопка GRID (СЕТКА) позволяет включать/выключать отображаемую сетку из точек. Этой кнопке соответствует функциональная клавиша F7.

ORTHO (ОРТО)

Кнопка ORTHO (ОРТО) включает/выключает специальный режим ортогональности (свойство объектов, когда касательные к ним в точке пересечения перпендикулярны друг другу). В процессе построения линии указатель мыши может перемещаться либо в вертикальном, либо в горизонтальном направлении. Этой кнопке соответствует функциональная клавиша F8.

POLAR (ОТС-ПОЛЯР)

Кнопка POLAR (ОТС-ПОЛЯР) включает/выключает режим полярного отслеживания. Этот режим представляет собой расширение режима ORTHO (ОРТО) и позволяет обеспечивать точность построений. Полярное отслеживание – это процесс отслеживания фиксированного направления от текущей точки привязки. При нахождении графического курсора в точке привязки через некоторое время появляется оранжевый крестик, отмечающий начало процесса полярного отслеживания. Отсчет полярного угла начинается против часовой стрелки от горизонтальной оси, направленной вправо. Кнопке POLAR (ОТС-ПОЛЯР) соответствует функциональная клавиша F10.

OSNAP (ПРИВЯЗКА)

Кнопка OSNAP (ПРИВЯЗКА) позволяет выбирать определенные точки в процессе редактирования рисунка. Этой кнопке соответствует функциональная клавиша F3.

OTRACK (ОТС-ОБЪЕКТ)

Кнопка OTRACK (ОТС-ОБЪЕКТ) позволяет включать/выключать режим объектного отслеживания, при котором можно использовать полярное отслеживание как средство обеспечения точности построений от промежуточной точки, указываемой с помощью объектной привязки. Отслеживание объектных привязок – это метод, являющийся комбинацией объектных привязок и полярных отслеживаний. Этой кнопке соответствует функциональная клавиша F11.

DUCS (ДПСК)

Кнопка DUCS (ДПСК) включает/отключает режим динамических координат при работе с трехмерными объектами. Этой кнопке соответствует функциональная клавиша F6.

DYN (ДИН)

Кнопка DYN (ДИН) обозначает динамический ввод. Она предназначена для включения режима дублирования информации, выводимой в командной строке, в области курсора. Этой кнопке соответствует функциональная клавиша F12.

В версиях AutoCAD 2008 и выше при создании объектов размеры для построения отображаются и редактируются непосредственно на чертеже. В активной области чертежа выводится также информация из командной строки и список параметров текущей команды.

Такая возможность позволяет сосредоточить внимание именно на создаваемом чертеже, не отвлекаясь на работу в командной строке. При

использовании динамического ввода также ведется история и поддерживается многократный повтор введенных пользователем команд и значений.

LWT (ВЕС)

Кнопка LWT (ВЕС) включает/выключает режим отображения линий в соответствии с их весом. При создании примитивов в AutoCAD можно задать не только толщину линий, с которой они будут отображаться на экране, но и толщину, с которой они будут выводиться на печать. Последнее свойство называется весом линии. Веса линий могут отображаться на экране или же быть видны только при печати. Для отображения их на экране нужно включить кнопку LWT (ВЕС).

MODEL (МОДЕЛЬ)

Кнопка MODEL (МОДЕЛЬ) позволяет переключаться между пространствами модели и листа.

Счетчик координат

Слева от рассмотренной выше панели режимов рисования находится счетчик координат (рис. 9.25).



871.2356, 707.4019, 0.0000

Рис. 9.25. Счетчик координат

Он предназначен для отображения текущих координат указателя мыши, находящегося в области графического экрана, и служит для ориентации в поле чертежа. Включить/выключить счетчик можно, щелкнув на нем кнопкой мыши.

В этом же месте строки состояния выводится и справочная информация о выполняемой команде. Так, при наведении указателя мыши на какую-либо кнопку панели инструментов в строке состояния появится описание этой кнопки.

Annotation Scale (Масштаб аннотационных объектов)

Меню *Annotation Scale (Масштаб аннотационных объектов)* позволяет установить текущее значение масштаба аннотаций. Любые аннотационные объекты, добавляемые на чертеж, будут отображаться в масштабе, соответствующем указанному значению.

Кнопка *Annotation Visibility (Отображение аннотации)* изменяет значение системной переменной ANNOALLVISIBLE. Когда ее значение равно 1, на экране отображаются все аннотационные объекты. Если переменной присвоить значение 0, будут видны лишь те объекты, которые поддерживаются текущим значением аннотационного масштаба.

Следующая кнопка изменяет значение системной переменной ANNOAUTOSCALE, которая управляет автоматическим масштабированием. Начальное значение переменной равно 4, автоматическое масштабирование при этом отключено. При нажатии кнопки переменная принимает значение 4. Все аннотационные объекты, поддерживающие текущее значение аннотационного масштаба, при его изменении будут также изменять свой масштаб.

Закрепление на экране палитр и панелей инструментов

Далее расположена кнопка *Toolbar/Window Positions Unlocked/Locked* (*Расположение панелей инструментов/окон не закреплено/Закреплено*). В зависимости от своего состояния эта кнопка имеет вид открытого или закрытого замка. При щелчке на ней можно заблокировать палитры и панели инструментов, тем самым оградив себя от случайного изменения их позиций на экране.

Очистка экрана

Последней в строке состояния расположена кнопка очистки экрана — *Clean Screen* (*Очистить экран*). Об одноименной команде меню *View* (*Вид*) уже говорилось выше. Вызов данной команды из строки состояния ничем не отличается от запуска ее с помощью меню или использования сочетания клавиш *Ctrl+0*.

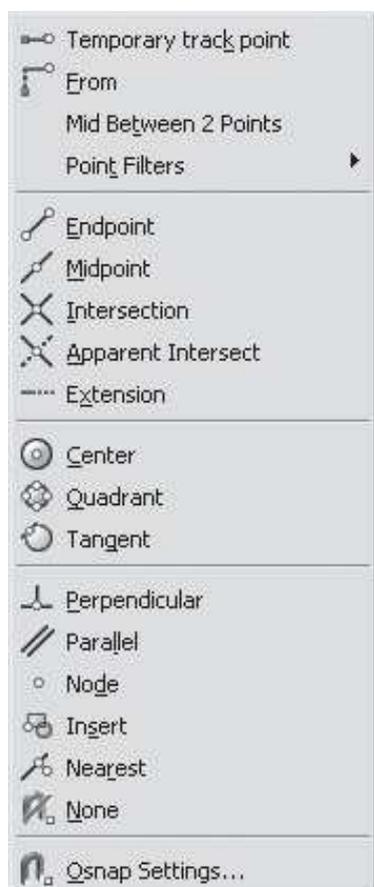


Рис. 9.26. Контекстное меню
Объектная привязка

9.1.6. Контекстные меню

При работе в пакете AutoCAD широко используются контекстные меню. Многие начинающие пользователи не до конца осознают все преимущества, которыми обладают эти меню. Контекстное меню — это меню, которое вызывается щелчком правой кнопкой мыши при нахождении указателя над каким-либо объектом интерфейса или при выполнении команды во время построения примитивов. Содержимое меню может различаться в зависимости от месторасположения в данный момент указателя мыши или от команды, выполняемой AutoCAD.

В качестве примера можно привести работу с привязками. В AutoCAD существует панель инструментов *Object Snap* (*Объектная привязка*), которая содержит кнопки со всевозможны-

ми вариантами привязок. Естественно, добавление очередной панели инструментов в окно программы закрывает определенную часть графического экрана, что весьма критично при небольшой диагонали монитора. При использовании контекстного меню необходимость постоянного присутствия на экране этой панели инструментов практически отпадает.

Так, если при построении какого-либо объекта возникает необходимость привязки, достаточно, удерживая клавишу Shift, щелкнуть правой кнопкой мыши на графической области экрана. При этом рядом с указателем мыши появится контекстное меню (рис. 9.26), содержащее все варианты привязки. Достаточно выбрать нужный и продолжить работу.

Здесь приведен только один вариант работы с контекстным меню. Вообще, таких вариантов существует множество. Контекстных меню в пакете AutoCAD огромное количество, и зачастую работа с ними намного удобнее, чем с панелями инструментов.

9.1.7. Работа с файлами

Все чертежи в системе AutoCAD представляют собой файлы, имеющие расширение DWG. В одном таком файле программа сохраняет всю содержащуюся на чертеже информацию: сам чертеж, всевозможные стили, блоки и массу других сведений. Чертежи, сохраняемые по умолчанию в AutoCAD 2008, могут быть свободно открыты лишь в текущей версии и в AutoCAD 2007 (более ранние версии использовать не рекомендуется).

Создание чертежа

Рассмотрим процесс создания нового документа в AutoCAD 2008. Для формирования чертежа необходимо выполнить команду File→New (Файл→Создать). После этого действия откроется диалоговое окно *Create New Drawing (Создание нового чертежа)* (рис. 9.27).

Внимание. Если диалоговое окно Create New Drawing (Создание нового чертежа) не появилось, значит, в настройках AutoCAD 2008 установлено создание чертежа по простейшему шаблону. Чтобы отменить эту настройку, необходимо присвоить системной переменной STARTUP значение 1.

В верхней части открывшегося окна размещены четыре кнопки. Первая из них, *Open a Drawing (Открытие чертежа)*, вызывает диалоговое окно открытия файла чертежа (рис. 9.28), в котором можно выбрать созданный ранее чертеж. Данная кнопка доступна только при старте системы. Если же вы выполняете команду File→New (Файл→Создать), то эта кнопка неактивна.

Следующая кнопка – *Start from Scratch (Простейший шаблон)* (см. рис. 9.27). С ее помощью можно создать простейший шаблон. При этом не-

обходимо указать всего один параметр – в каких единицах измерения будет производиться построение: *Imperial* (Британские) или *Metric* (Метрические).

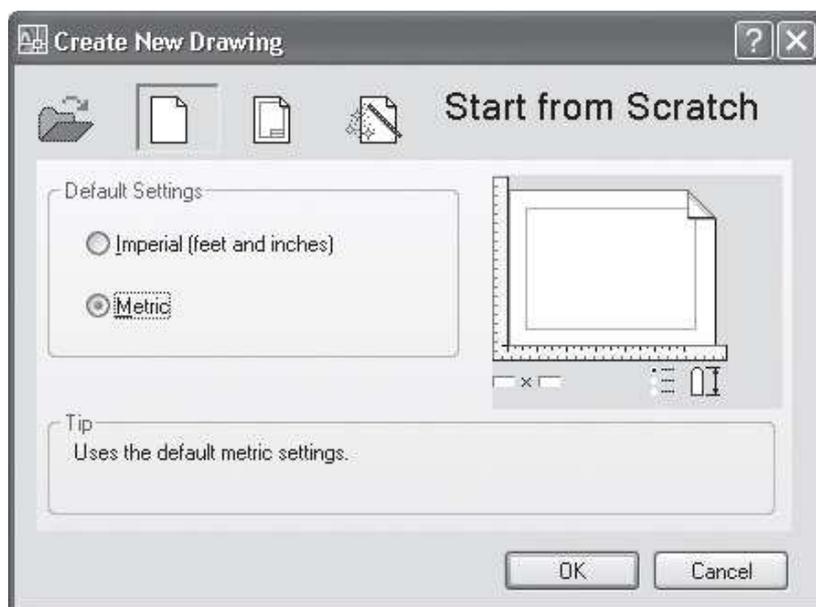


Рис. 9.27. Диалоговое окно *Create New Drawing* (Создание нового чертежа)

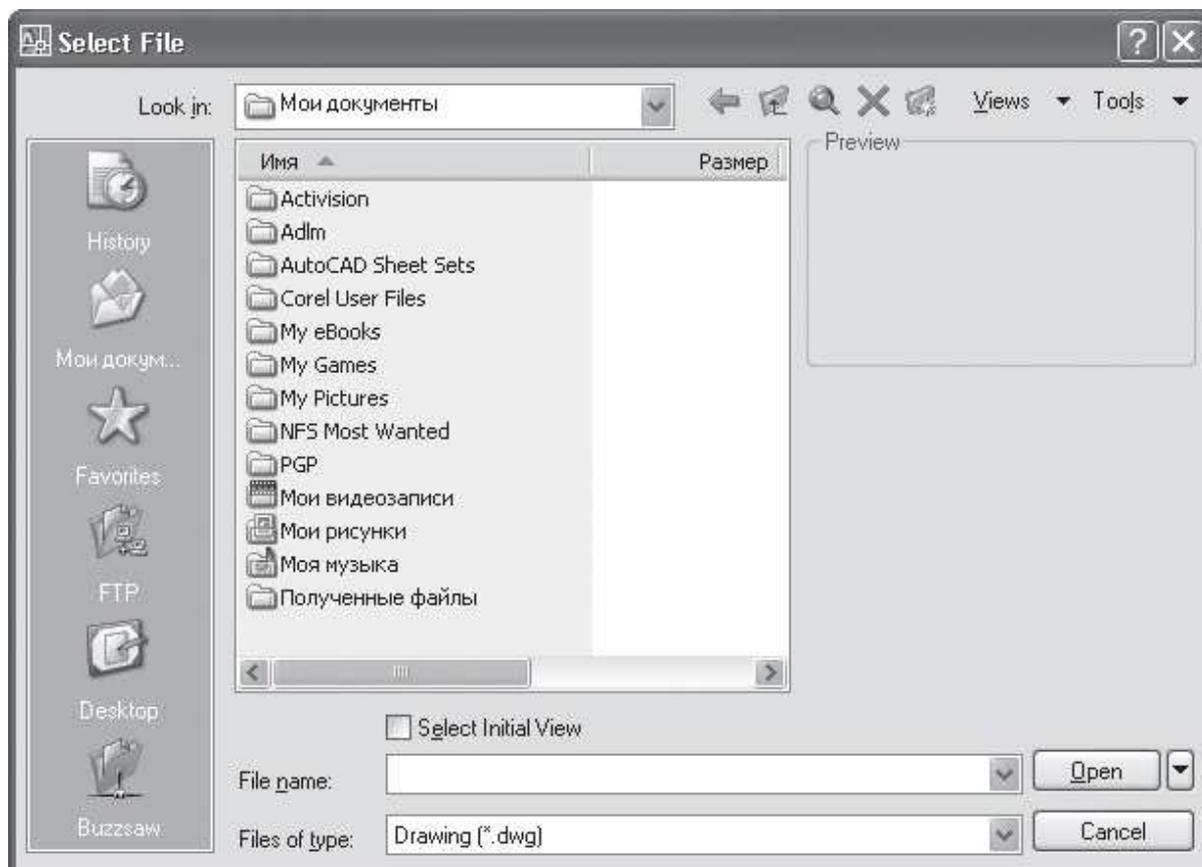


Рис. 9.28. Диалоговое окно открытия файла

Примечание. Шаблон – это чертеж, который содержит некоторые общие настройки и используется как основа для создания новых чертежей.

С помощью следующей кнопки *Use a Template (Использовать шаблон)* можно создать чертеж по заранее подготовленному шаблону (рис. 9.29). Шаблоны AutoCAD очень похожи на шаблоны Microsoft Word. При установке AutoCAD также устанавливается большое количество шаблонов.

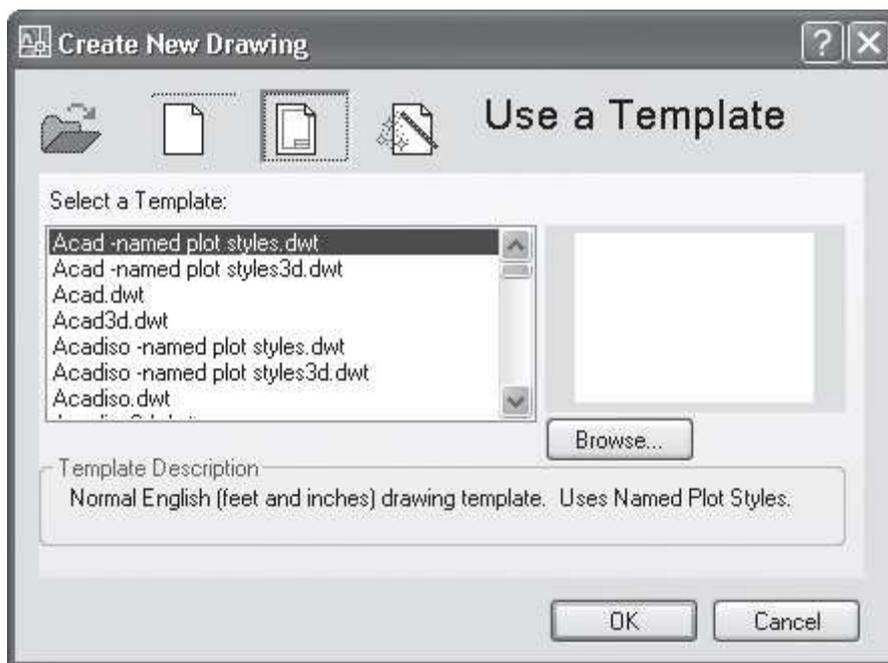


Рис. 9.29. Создание нового чертежа по шаблону

Примечание. Поставляемые вместе с AutoCAD шаблоны не соответствуют единой системе конструкторской документации (ЕСКД). Поэтому пользователю придется изготовить шаблоны с полной поддержкой российских стандартов самостоятельно.

И последняя в этом перечне кнопка *Use a Wizard (Использовать Мастер)* дает возможность быстро установить настройки будущего чертежа и приступить к работе. Пакет AutoCAD имеет два мастера, предназначенных для установки начальных параметров создаваемого рисунка: *Advanced Setup (Расширенная установка)* и *Quick Setup (Быстрая установка)* (рис. 9.30).

Рассмотрим подробнее работу с мастером *Advanced Setup (Расширенная установка)*. После его выбора на экране появляется окно, в котором вам предлагается указать единицы измерения для построения чертежа (рис. 9.31). Также в этом окне можно установить точность единиц измерения (количество знаков после точки).



Рис. 9.30. Создание чертежа с использованием мастера

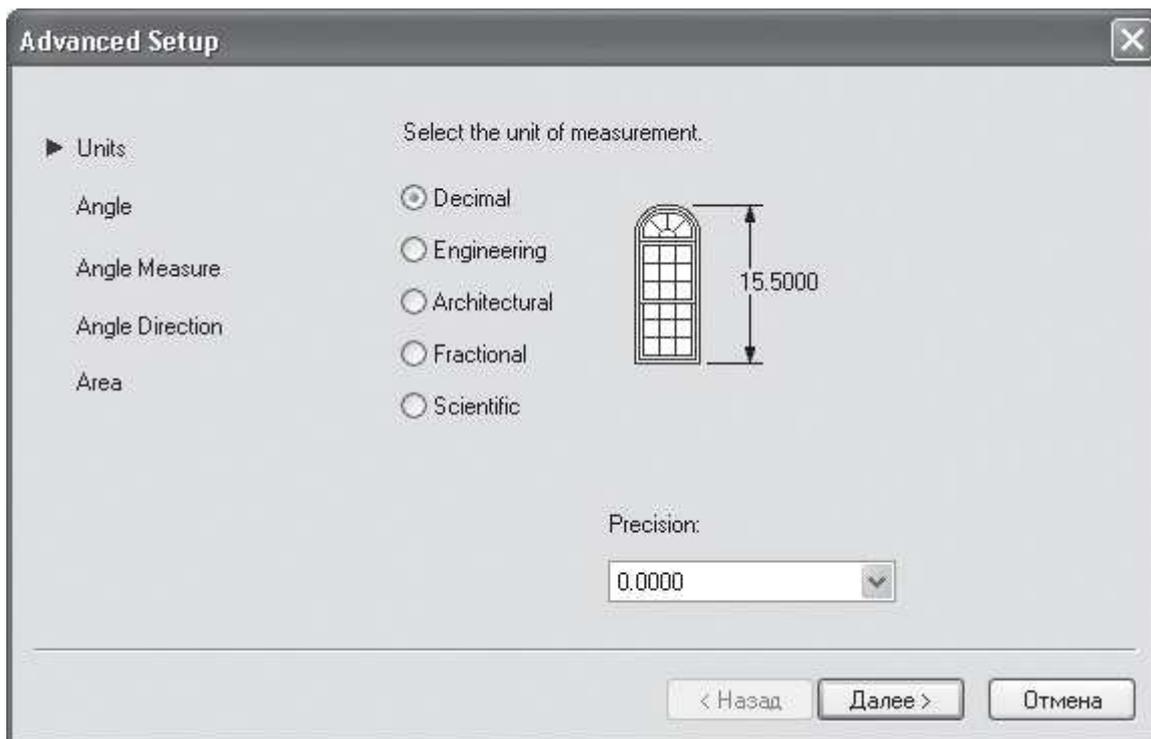


Рис. 9.31. Выбор единиц измерения

В следующем окне (рис. 9.32) выберите тип используемых угловых единиц: *Decimal Degrees* (Десятичные градусы), *Deg/Min/Sec* (Град/Мин/Сек), *Grads* (Грады), *Radians* (Радианы) или *Surveyor* (Топографические) – и их точность.

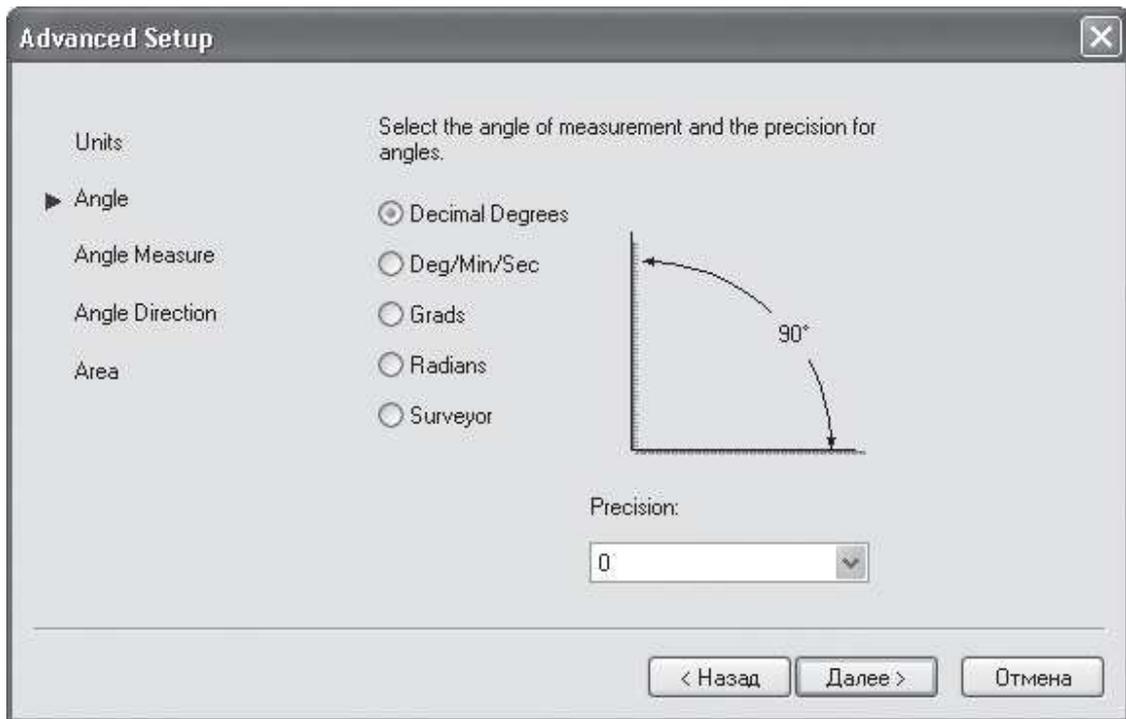


Рис. 9.32. Выбор угловых единиц

Далее выберите направление для нулевого угла.

В следующем окне задайте направление отсчета углов: *Counter-Clockwise* (Против часовой стрелки) или *Clockwise* (По часовой стрелке).

И последний шаг мастера – установка размеров области, выбранной для рисования (рис. 9.33).

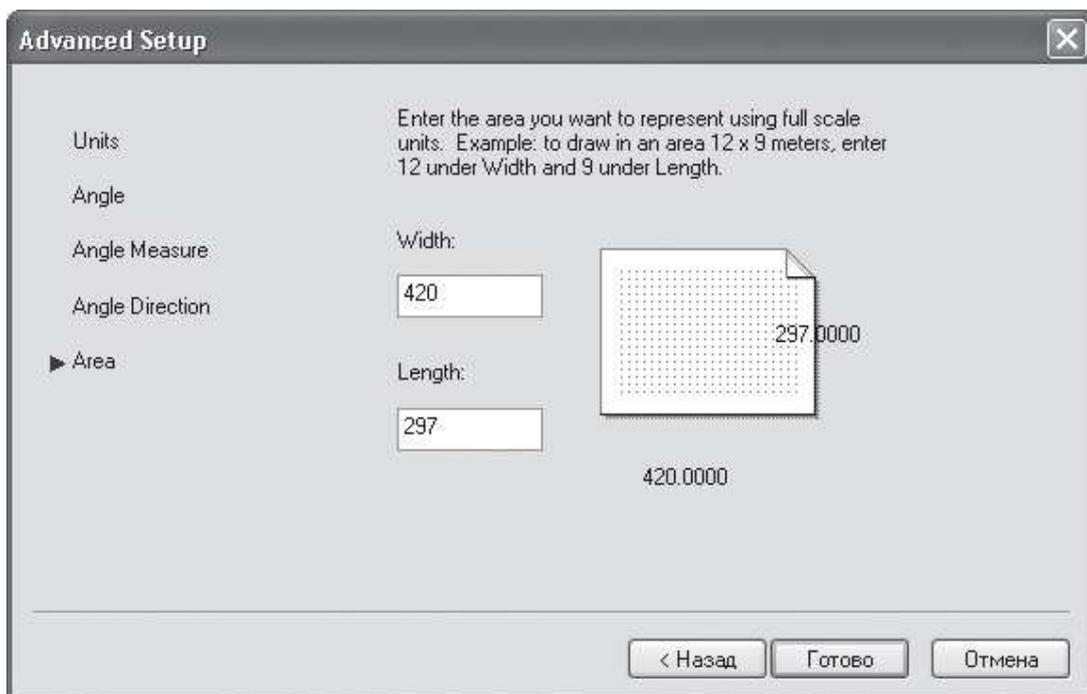


Рис. 9.33. Установка размеров листа

Сохранение чертежа

После создания чертежа следует выполнить первое сохранение рисунка. Для этого необходимо нажать кнопку *Save* (*Сохранить*), расположенную на панели инструментов *Standard Annotation* (*Стандартная аннотационная*). Откроется окно, в котором нужно выбрать папку для сохранения и ввести имя файла вашего чертежа. По умолчанию пакет AutoCAD присваивает чертежам имена *Drawing1*, *Drawing2* и т.д.

Совет. Не сохраняйте созданные файлы под именами, которые предлагает по умолчанию программа AutoCAD. Ориентироваться в большом количестве проектов со схожими именами будет проблематично.

Завершение работы

Для завершения работы программы необходимо выполнить команду *File*→*Exit* (*Файл*→*Выход*) или нажать кнопку закрытия программы, которая расположена в правом верхнем углу окна AutoCAD. Можно также воспользоваться сочетанием клавиш *Ctrl+Q*. Если с момента последнего сохранения в чертеже были произведены какие-либо изменения, при закрытии AutoCAD программа выдаст запрос о сохранении этих изменений.

9.1.8. Управление экраным отображением

Начиная работу с чертежами в AutoCAD, непременно овладеть навыками работы с инструментами управления экраным отображением. Часто в процессе работы необходимо изменить масштаб просмотра чертежа, переместиться к определенному его месту и т.д. Для управления экраным отображением в AutoCAD используется несколько основных команд.

Панорамирование

Команда *Pan* (*Панорамирование*) дает возможность перемещать видимую область чертежа. Она актуальна при работе над проектами большого размера, когда весь чертеж не может уместиться на экране.

Для использования команды *Pan* (*Панорамирование*) нажмите кнопку *Pan Realtime* (*Панорамирование в реальном времени*) на панели *2D Navigate* (*Двухмерная навигация*) пульта управления, переместите указатель в виде ладошки на рабочую область, нажмите кнопку мыши и, не отпуская ее, перетаскивайте его до тех пор, пока на экране не появится необходимая вам область.

Чтобы завершить выполнение команды, нажмите клавишу *Enter* или *Esc*.

Полосы прокрутки

Для перемещения по чертежу служат также горизонтальная и вертикальная полосы прокрутки. Позиция ползунка на каждой из них позволяет определить, в каком месте чертежа пользователь сейчас находится – в начале, в конце или посередине.

По умолчанию полосы прокрутки не отображаются в окне программы. Для того чтобы их вызвать, нужно выполнить команду *Tools*→*Options* (*Сервис*→*Параметры*), в открывшемся диалоговом окне перейти на вкладку *Display* (*Экран*) и в области *Window Elements* (*Элементы окна*) установить флажок *Display scroll bars in drawing window* (*Отображать полосы прокрутки в окне чертежа*).

Для навигации по чертежу с помощью полос прокрутки щелкните на одной из них кнопкой мыши и, удерживая ее, перетяните ползунок на вертикальной полосе вверх или вниз, а на горизонтальной – влево или вправо.

Для перемещения по чертежу можно использовать кнопки в виде стрелок, которые расположены по краям полос прокрутки.

Масштабирование

Еще один удобный способ управления экранным отображением – использование команды *Zoom* (*Масштабирование*). Она дает возможность увеличить определенный участок чертежа и рассмотреть его поближе или же, наоборот, отдалить изображение, чтобы весь чертеж помещался на экране.

Для выполнения команды масштабирования нажмите кнопку *Realtime* (*Масштабирование в реальном времени*) на панели *2D Navigate* (*Двухмерная навигация*) пульта управления. Затем щелкните на экране и, не отпуская кнопку мыши, перемещайте указатель вверх для увеличения масштаба изображения и вниз для его уменьшения.

Возможностей по масштабированию изображения в AutoCAD много, на панели пульта управления представлены лишь наиболее часто используемые. Полностью же команды находятся на панели инструментов *Zoom* (*Масштабирование*) (рис. 9.34).



Рис. 9.34. Панель инструментов *Zoom* (*Масштабирование*)

С ее помощью можно выполнить такие операции, как масштабирование выделенных объектов во весь экран, масштабирование всего изображения во весь экран, масштабирование по выбранной точке и т.д.

ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

1. Основные понятия информатики.
2. Виды и свойства информации.
3. Что собой представляют процессы восприятия, сбора, передачи, обработки и накопления информации?
4. Охарактеризуйте этапы развития информатики.
5. Классификация ЭВМ по назначению, уровню специализации, типоразмерам и совместимости.
6. Кодирование данных в ЭВМ двоичным кодом.
7. Системы счисления.
8. Единицы измерения информации.
9. Базовая конфигурация персонального компьютера.
10. Носители информации.
11. Периферийные устройства персонального компьютера.
12. Программное обеспечение средств вычислительной техники.

13. Назначение, классификация и основные функции операционных систем.
14. Общая характеристика операционной системы Windows.
15. Основные элементы графического интерфейса Windows: рабочий стол Windows, значки и ярлыки объектов, окна.
16. Файловая система и файловая структура Windows.
17. Что такое файл? Какие требования предъявляются к именам файлов? Из каких элементов состоит полное имя файла? Что собой представляют атрибуты файла?
18. Перечислите основные операции с файловой структурой. Какие средства существуют в Windows для работы с файловой структурой?
19. Главное меню Windows: его назначение и структура.
20. Обмен данными в Windows.
21. Стандартные приложения Windows.
22. Программы технического обслуживания.
23. Программы-архиваторы.
24. Антивирусные программы.

25. Понятие и основные свойства алгоритма.
26. Способы записи алгоритмов.
27. Разновидности структур алгоритмов.

28. Программирование. Языки программирования высокого уровня.
29. Назначение и возможности текстового процессора Microsoft Word.
30. Режимы просмотра документов в Microsoft Word.
31. Установка параметров страницы. Ввод и редактирование текста.
32. Форматирование текста. Использование стилей при создании документа.
33. Верстка многостраничного документа.
34. Буфер обмена Microsoft Office.
35. Списки в Microsoft Word.
36. Создание и редактирование таблиц.
37. Создание и редактирование художественных надписей (WordArt).
38. Рисунки в Microsoft Word. Печать документа.
39. Назначение компьютерных сетей.
40. Локальные и глобальные сети. Основные понятия.
41. Службы Internet: Telnet, электронная почта (E-Mail), служба телеконференций (Usenet).
42. Службы Internet: World Wide Web (WWW).
43. Работа с гипертекстовой информацией в сети Internet.
44. Службы Internet: служба имен доменов (DNS), служба передачи файлов (FTP), ICQ.
45. Поиск информации в Internet с помощью поисковых систем.
46. Растровая графика.
47. Векторная и фрактальная графика.
48. Программные средства для работы с графикой.
49. Средства подготовки презентаций.
50. Microsoft PowerPoint как средство создания презентаций, его основные возможности.
51. Воспроизведение презентаций: презентации на экране и в Internet.

52. Назначение и возможности табличного процессора Microsoft Excel.
53. Ввод и редактирование данных. Ввод последовательностей данных в ячейки. Настройка автозаполнения.
54. Организация вычислений: автоматическое суммирование строк и столбцов; составление формул; формулы с относительными и абсолютными адресами.
55. Диаграммы.
56. Сортировка данных.
57. Подведение итогов.

58. Подбор параметра.
59. Таблицы подстановки данных.
60. Приближенное решение нелинейных уравнений.
61. Методы решения систем линейных уравнений.
62. Приближенное вычисление определенных интегралов.
63. Приближенное решение дифференциальных уравнений.
64. Интерполирование функций.
65. Регрессия.

66. Пакет Mathcad: общая характеристика, пользовательский интерфейс.
67. Пакет Mathcad: базовые элементы входного языка.
68. Пакет Mathcad: основные виды вычислений, численные и символьные вычисления.
69. Пакет Mathcad: графический анализ данных.
70. Пакет Mathcad: работа с массивами данных (вектора, матрицы).
71. Пакет Mathcad: использование программных фрагментов, программирование основных типов алгоритмов.
72. Пакет Mathcad: ввод и вывод данных различных типов.
73. Пакет Mathcad: решение уравнений.
74. Пакет Mathcad: выполнение символьных вычислений.
75. Пакет Mathcad: нахождение экстремумов функций.
76. Пакет Mathcad: аппроксимация функций.
77. Пакет Mathcad: вычисление определенных интегралов.
78. Пакет Mathcad: решение дифференциальных уравнений.
79. Пакет Mathcad: решение уравнений в частных производных.

ЛИТЕРАТУРА

1. Информатика: Базовый курс / под ред. С.В. Симоновича. – СПб.: Питер, 2001.
2. Программирование на персональных ЭВМ: Практикум: учеб. пособие / Д.В. Офицеров и [и др.]. – Минск: Выш. шк., 1993. – 256 с.
3. Дьяконов, В.П. MathCad 2001: учеб. курс / В.П. Дьяконов. – СПб.: Питер, 2001.
4. Очков, В.Ф. MathCad 8 Pro для студентов и инженеров / В.Ф. Очков. – М., 1999.
5. Информатика и компьютерная графика: учеб.-метод. комплекс для студентов специальности 1-48 01 03. В 2 ч. Ч. 1 / сост. и общ. ред. А.В. Спиридонова. – Новополоцк: ПГУ, 2011. – 215 с.
6. Справочное пособие по приближенным методам решения задач высшей математики / Л.И. Бородич [и др.]. – Минск: Высш. шк., 1986.

Учебное издание

СПИРИДОНОВ Александр Владимирович

ИНФОРМАТИКА

Учебно-методический комплекс для студентов специальностей 1-70 04 02
«Теплогазоснабжение, вентиляция и охрана воздушного бассейна»,
1-70 04 03 «Водоснабжение, водоотведение и охрана водных ресурсов»

В двух частях

Часть 1

Редактор *О. П. Михайлова*
Дизайн обложки

Подписано в печать 26.12.2013. Формат 60×84 1/16. Бумага офсетная. Ризография.
Усл. печ. л. 16,24. Уч.-изд. л. 15,72. Тираж 30 экз. Заказ 1812.

Издатель и полиграфическое исполнение:
учреждение образования «Полоцкий государственный университет».

ЛИ № 02330/0548568 от 26.06.2009 ЛП № 02330/0494256 от 27.05.2009

Ул. Блохина, 29, 211440, г. Новополоцк.